*Thesis*

# Understanding, Formally Characterizing, and Robustly Handling Real-World Distribution Shift

Elan Rosenfeld

May 2024
CMU-ML-24-105

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

*Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.*

This research was sponsored by Air Force Research Laboratory award FA87502321015; Defense Advanced Research Projects Agency award HR00112020006; National Science Foundation awards CCF2238523, IIS1664720, IIS1909816, IIS1955532, OAC1934584; Office of Naval Research award N000142312368; an Amazon Research Award; and research grants from Optum Technology Inc., PricewaterhouseCoopers LLP, and Rakuten Inc.

**Keywords:** machine learning, robustness, adversarial robustness, distribution shift, out-of-distribution generalization, domain generalization, domain adaptation, representation learning, causal representation learning, invariance

*For Rafi*

## Abstract

Distribution shift remains a significant obstacle to successful and reliable deployment of machine learning (ML) systems. Long-term solutions to these vulnerabilities can only come with the understanding that benchmarks fundamentally cannot capture all possible variation which may occur; equally important, however, is careful experimentation with AI systems to understand their failures under shift in practice.

This thesis describes my work towards building a foundation for trustworthy and reliable machine learning. The surveyed work falls roughly into three major categories: (i) designing formal, practical characterizations of the structure of real-world distribution shift; (ii) leveraging this structure to develop provably correct and efficient learning algorithms which handle such shifts robustly; and (iii) experimenting with modern ML systems to to understand the practical implications of real-world heavy tails and distribution shift, both average- and worst-case.

Part I describes work on scalably certifying the robustness of deep neural networks to adversarial attacks. The proposed approach can be used to certify robustness to attacks on test samples, training data, or more generally any input which influences the model's eventual prediction. In Part II, we focus on latent variable models of shifts, drawing on concepts from causality and other structured encodings of real-world variation. We demonstrate how these models enable formal analysis of methods that use multiple distributions for robust deep learning, particularly through the new lens of *environment/intervention complexity*—a core statistical measure for domain generalization and causal representation learning which quantifies error and/or structured identifiability conditions as a function of the number and diversity of available training distributions. Finally, in Part III we broadly explore ways to better understand and leverage the variation in natural data, and we show how the resulting insights can facilitate the design of new methods with more robust and reliable real-world behavior.

# Acknowledgements

First and foremost I would like to thank my advisors, Pradeep Ravikumar and Andrej Risteski. Pradeep took me on as an advisee even when I was very uncertain about what I wanted to work on. Early on he provided me with an expert balance of directed guidance and personal autonomy, allowing me to explore topics at my own pace until I found a field which truly excited me. I am also grateful for his astute insights into the importance of prioritizing impactful research and for enabling and encouraging me to take such risks. Likewise, this thesis would not have been possible without Andrej's continuous guidance and support. I have learned so much from our time working together, and I aspire to one day emulate his tenacity, optimism, and wisdom. I have enormous gratitude for his immense patience and understanding over the years—which was surely tested many times, though he never showed it.

Thanks to Zico Kolter, Zachary Lipton, and Sivaraman Balakrishnan, who generously offered their valuable time and advice whenever I needed it. I am especially indebted to Zico, who graciously took on the role of an unofficial co-advisor for my first two years.

I am deeply grateful for the close relationships I've formed over the years, each of which played an instrumental role in my personal growth. From the very start I was fortunate to be assigned to the same office as Jeremy Cohen, who quickly became one of my closest friends. We did our first class project together, which then turned into our first paper. Working alongside Jeremy inspired me to work harder and think outside the box, and I cherished our frequent (occasionally heated) discussions. Perhaps the only person with whom I spent more time during the past six years is Brenna Rosen. I feel so lucky to have met Brenna shortly after moving back to Pittsburgh. I can't imagine having made it this far without her loving support and encouragement, for which I will be forever grateful.

There are many other friends with whom I am glad to have shared this journey, including (but certainly not limited to) my officemate and late-night confidant Adarsh Prasad, my labmates Bingbin Liu, Tanya Marwah, and Yuchen Li, as well as Ojash Neopane, Samuel Sokota, Ian Char, Shantanu Gupta, Saurabh Garg, Jeremiah Milbauer, Nicholas Roberts, Roger Iyengar, Vijay Viswanathan, Wojtek Nawrocki, Rattana Pukdee, Roie Levin, Kelvin Liu-Huang, and Stefani Karp. Thanks also to Laura Winter, Amy Protos, Dorothy Holland-Minkley, and especially Diane Stidle and Sharon Cavlovich for their tireless support, and without whom the entire department would surely cease to function.

I have had many advisors and educators who provided the guidance and inspiration which set me on this path. I am especially grateful to Manuel Blum, my first academic advisor. Manuel introduced me to the joys of research, and working with him is what convinced me to pursue a PhD. I would also like to thank David Nassar for believing in me from the beginning, and for providing the first classroom environment which I can remember genuinely looking forward to.

Finally, none of this would have been possible without my family. I want to thank my father Roni Rosenfeld, my mother Ilana Diamond, and my sisters Amitte Rosenfeld and Maia Rosenfeld for their unwavering love and support, now and forever.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Prediction algorithms are evaluated by—and valued for—their performance on unseen test data. In classical machine learning (ML), it is common to assume that such data are drawn independently from one another, and from an identical distribution to that which gave rise to the dataset on which the learning algorithm was trained (this is known as the *IID assumption*). In the real world, however, such a condition is almost never satisfied. The IID assumption serves as a valuable abstraction for studying how to efficiently and reliably learn from data. But statisticians have long understood that this assumption is an oversimplification, and that the underlying distribution of real-world data is continuously undergoing *shift*: examples include shift over time, shift across heterogenous subpopulations, shift induced as a response to past actions, etc. Because of this discrepancy between reality and the idealized assumption of IID data, algorithms which provide strong in-distribution generalization guarantees, such as empirical risk minimization [Vapnik, 1999], fail unexpectedly in the real world, often with high confidence and no prior warning. In particular, while modern deep neural networks achieve superhuman performance on many tasks, there is growing evidence that their incredible generalization ability is primarily limited to out-of-sample settings where the test data remains very similar to what they were trained on. In particular, these models seem to rely on statistically informative representations of the data which—for reasons not yet fully understood—go well beyond simple memorization of the training data, yet which will often not allow them to generalize to new domains or tasks. This is the case even for seemingly trivial distribution shifts which pose little-to-no trouble to humans [Beery et al., 2018, Geirhos et al., 2018]. As a result, modern state-of-the-art generative and discriminative deep networks are brittle in deployment and are prone to errors under surprisingly minor distribution shift [Su et al., 2019, Recht et al., 2019].

In considering how to address this weakness, it is tempting to imagine that the methods which have enabled the aforementioned practical successes in deep learning will eventually solve this problem as well. The primary driving force behind the incredible pace of the past decade of ML research has been the “benchmark approach”: progression on one or more tasks via consistent, incremental improvements on a collection of representative benchmark datasets. Though the success of this strategy is undeniable, it has become clear that it is insufficient for achieving a future of truly robust and reliable ML. Artificial intelligence (AI) is rapidly being deployed in myriad new domains—and it will only become more widespread—but it cannot yet be widely relied upon, while the looming costs of unexpected failures continue to grow. Meanwhile, examples of the shifts which induce such failures in the real world abound: they arise, for example, as simple changes in scenery and/or weather encountered by a self-driving car, or when users adapt their behavior to an ML system to increase the likelihood of their preferred outcome [Hardt et al., 2016]. Even worse, AI is increasingly being used in safety-critical settings, which presents serious security vulnerabilities in the face of a determined adversary [Sharif et al., 2016]. This brittleness remains a significant obstacle to further trustworthy deployment of ML systems.

Long-term solutions to these vulnerabilities can only come with the understanding that benchmarks fundamentally cannot capture all possible variation which may occur. But it is also clear that robustness to *all* distribution shifts is infeasible. Instead, we must first design precise, realistic mathematical definitions of real-world distribution shift: by formally specifying the “threat model” of shift to which we wish to be robust, we will be able to make reliable progress towards formal robustness guarantees. At the same time, there is frequent mismatch between ML theory and practice (especially in deep learning), and thus a mathematical definition of shift alone is not sufficient. It is also necessary that we carefully experiment with AI systems to understand their failure modes in practice—only through such experimentation can we understand and reconcile the discrepancies between real-world data and our mathematical understanding of it. In turn, this will enable the development of new, more reliable and interpretable ML methods with practical downstream benefits to performance.

This thesis describes progress towards a foundation for trustworthy and reliable machine learning, achieved via a combination of these two core approaches. More precisely, the surveyed work falls roughly into three major categories: (i) designing formal, practical characterizations of the structure of real-world distribution shift, both benign and adversarial; (ii) leveraging this structure to develop provably correct and efficient learning algorithms which handle such shifts robustly; and (iii) experimenting with modern ML systems to to understand the practical implications of distribution shift, both average- and worst-case, so that future analysis might better capture the kinds of difficulties we expect AI to encounter moving forward.

## 1.1 Thesis Overview

### Part I

The first part of this thesis describes work on scalably certifying the robustness of deep neural networks to adversarial attacks. Chapter 2 shows how to turn any classifier that classifies well under Gaussian noise into a new classifier that is certifiably robust to adversarial perturbations under the $\ell_2$ norm. We prove a tight robustness guarantee in $\ell_2$ norm for smoothing with Gaussian noise, obtaining an ImageNet classifier with (e.g.) a certified top-1 accuracy of 49% under adversarial perturbations with $\ell_2$ norm less than 0.5 (=127/255). In Chapter 3, we demonstrate how the proposed approach can be used to certify robustness to attacks more generally, such as adversarial modification to the training data, or more generally any input which influences the model's eventual prediction.

### Part II

The second part focuses on latent variable models of shifts, taking inspiration from causality and other proposed structured encodings of real-world variation. We demonstrate the importance of these models and how they enable new approaches formal analysis of methods that use multiple distributions for robust deep learning. In particular, we study these algorithm's behavior through the new lens of *environment/intervention complexity*—a core statistical measure for domain generalization and Causal Representation Learning which quantifies error and/or latent feature identifiability as a function of the number of environments seen. Chapter 4 presents the first analysis of classification under various objectives proposed for these tasks under a fairly natural and general model. We furthermore present the very first results in the non-linear regime: demonstrating that these methods can fail catastrophically unless the test data are sufficiently similar to the training distribution. This is followed by an improved analysis along with even stronger lower bounds in Chapter 5. Chapter 6 considers the setting of online domain generalization, formally quantifying for the first time a computational complexity gap between domain “interpolation” and “extrapolation”.

### Part III

The last part of this thesis broadly explores ways to better understand and leverage the variation in natural data. First, in Chapter 7 we show that pretrained features are sufficient for substantially more robust predictors than previously believed. Chapter 8 describes how this finding enables the use of unlabeled test data to provably adapt neural networks to shifts just-in-time or to give (almost) provable non-vacuous bounds on their test error. Next, Chapter 9 develops a robust optimization approach to strategic classification, enabling doubly robust prediction which gracefully handles both strategic response *and* the inevitable uncertainty in users' cost functions. Lastly, Chapter 10 presents findings on the significant influence of outliers on neural network optimization—this result gives new insight into how the heavy tails of natural data affect network behavior, and it suggests a more coherent picture of the origin of a variety of phenomena in neural network optimization.

# Part I

# Certifying Robustness to Adversarial Attacks

Chapter 2

# Certified Adversarial Robustness via Randomized Smoothing

This chapter is based on Cohen et al. [2019]:
Cohen, J. M., Rosenfeld, E., & Kolter, J. Z.
Certified Adversarial Robustness via Randomized Smoothing.
In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

## 2.1 Introduction

Modern image classifiers achieve high accuracy on i.i.d. test sets but are not robust to small, adversarially-chosen perturbations of their inputs [Szegedy et al., 2013b, Biggio et al., 2013]. Given an image $x$ correctly classified by, say, a neural network, an adversary can usually engineer an adversarial perturbation $\delta$ so small that $x + \delta$ looks just like $x$ to the human eye, yet the network classifies $x + \delta$ as a different, incorrect class. Many works have proposed heuristic methods for training classifiers intended to be robust to adversarial perturbations. However, most of these heuristics have been subsequently shown to fail against suitably powerful adversaries [Carlini and Wagner, 2017, Athalye et al., 2018, Uesato et al., 2018]. In response, a line of work on *certifiable robustness* studies classifiers whose prediction at any point $x$ is verifiably constant within some set around $x$ [e.g. Wong and Kolter, 2018, Raghunathan et al., 2018a]. In most of these works, the robust classifier takes the form of a neural network. Unfortunately, all existing approaches for certifying the robustness of neural networks have trouble scaling to networks that are large and expressive enough to solve problems like ImageNet.

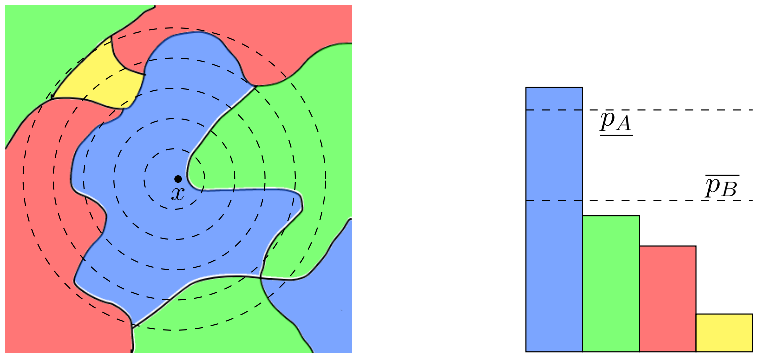


Figure 2.1: Evaluating the smoothed classifier at an input $x$. **Left**: the decision regions of the base classifier $f$ are drawn in different colors. The dotted lines are the level sets of the distribution $\mathcal{N}(x, \sigma^2 I)$. **Right**: the distribution $f(\mathcal{N}(x, \sigma^2 I))$. As discussed below, $\underline{p_A}$ is a lower bound on the probability of the top class and $\overline{p_B}$ is an upper bound on the probability of each other class. Here, $g(x)$ is "blue."

.

One workaround is to look for robust classifiers that are not neural networks. Recently, two papers [Lecuyer et al., 2019, Li et al., 2018a] showed that an operation we call *randomized smoothing*[1] can transform any arbitrary base classifier $f$ into a new "smoothed classifier" $g$ that is certifiably robust in $\ell_2$ norm. Let $f$ be an arbitrary classifier which maps inputs $\mathbb{R}^d$ to classes $\mathcal{Y}$. For any input $x$, the smoothed classifier's prediction $g(x)$ is defined to be the class which $f$ is most likely to classify the random variable $\mathcal{N}(x, \sigma^2 I)$ as. That is, $g(x)$ returns the most probable prediction by $f$ of random Gaussian corruptions of $x$.

If the base classifier $f$ is most likely to classify $\mathcal{N}(x, \sigma^2 I)$ as $x$'s correct class, then the smoothed classifier $g$ will be correct at $x$. But the smoothed classifier $g$ will also possess a desirable property that the base classifier may lack: one can verify that $g$'s prediction is constant within an $\ell_2$ ball around any input $x$, simply by estimating the probabilities with which $f$ classifies $\mathcal{N}(x, \sigma^2 I)$ as each class. The higher the probability with which $f$ classifies $\mathcal{N}(x, \sigma^2 I)$ as the most probable class, the larger the $\ell_2$ radius around $x$ in which $g$ provably returns that class.

Lecuyer et al. [2019] proposed randomized smoothing as a provable adversarial defense, and used it to train the first certifiably robust classifier for ImageNet. Subsequently, Li et al. [2018a] proved a stronger robustness guarantee. However, both of these guarantees are loose, in the sense that the smoothed classifier $g$ is *provably always* more robust than the guarantee indicates. In this paper, we prove the first tight robustness guarantee for randomized smoothing. Our analysis reveals

[1]Smoothing was proposed under the name "PixelDP" (for differential privacy). We use a different name since our improved analysis does not involve differential privacy.

that smoothing with Gaussian noise naturally induces certifiable robustness under the $\ell_2$ norm. We suspect that other, as-yet-unknown noise distributions might induce robustness to other perturbation sets such as general $\ell_p$ norm balls.

Randomized smoothing has one major drawback. If $f$ is a neural network, it is not possible to *exactly* compute the probabilities with which $f$ classifies $\mathcal{N}(x, \sigma^2 I)$ as each class. Therefore, it is not possible to exactly evaluate $g$'s prediction at any input $x$, or to exactly compute the radius in which this prediction is certifiably robust. Instead, we present Monte Carlo algorithms for both tasks that are guaranteed to succeed with arbitrarily high probability.

Despite this drawback, randomized smoothing enjoys several compelling advantages over other certifiably robust classifiers proposed in the literature: it makes no assumptions about the base classifier's architecture, it is simple to implement and understand, and, most importantly, it permits the use of arbitrarily large neural networks as the base classifier. In contrast, other certified defenses do not currently scale to large networks. Indeed, smoothing is the only certified adversarial defense which has been shown feasible on the full-resolution ImageNet classification task.

We use randomized smoothing to train state-of-the-art certifiably $\ell_2$-robust ImageNet classifiers; for example, one of them achieves 49% provable top-1 accuracy under adversarial perturbations with $\ell_2$ norm less than 127/255 (Table 2.1). We also demonstrate that on smaller-scale datasets like CIFAR-10 and SHVN, where competing approaches to certified $\ell_2$ robustness are feasible, randomized smoothing can deliver better certified accuracies, both because it enables the use of larger networks and because it does not constrain the expressivity of the base classifier.

## 2.2 Related Work

Many works have proposed classifiers intended to be robust to adversarial perturbations. These approaches can be broadly divided into *empirical* defenses, which empirically seem robust to known adversarial attacks, and *certified* defenses, which are *provably* robust to certain kinds of adversarial perturbations.

**Empirical defenses** The most successful empirical defense to date is *adversarial training* [Goodfellow et al., 2015, Kurakin et al., 2016, Madry et al., 2017], in which adversarial examples are found during training (often using projected gradient descent) and added to the training set. Unfortunately, it is typically impossible to tell whether a prediction by an empirically robust classifier is truly robust to adversarial perturbations; the most that can be said is that a specific attack was unable to find any. In fact, many heuristic defenses proposed in the literature were later "broken" by stronger adversaries [Carlini and Wagner, 2017, Athalye et al.,

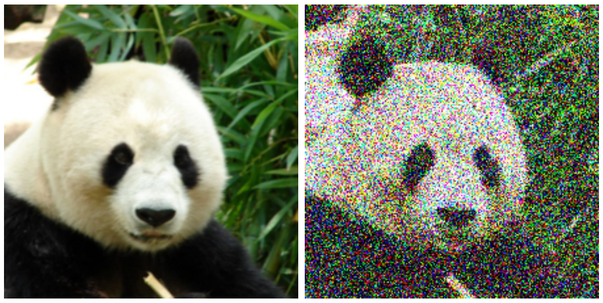


Figure 2.2: The smoothed classifier's prediction at an input $x$ (left) is defined as the most likely prediction by the base classifier on random Gaussian corruptions of $x$ (right; $\sigma = 0.5$). Note that this Gaussian noise is much larger in magnitude than the adversarial perturbations to which $g$ is provably robust.

2018, Uesato et al., 2018, Athalye and Carlini, 2018]. Aiming to escape this cat-and-mouse game, a growing body of work has focused on defenses with formal guarantees.

**Certified defenses** A classifier is said to be *certifiably robust* if for any input $x$, one can easily obtain a guarantee that the classifier's prediction is constant within some set around $x$, often an $\ell_2$ or $\ell_\infty$ ball. In most work in this area, the certifiably robust classifier is a neural network. Some works propose algorithms for certifying the robustness of generically trained networks, while others [Wong and Kolter, 2018, Raghunathan et al., 2018a] propose both a robust training method and a complementary certification mechanism.

Certification methods are either *exact* (a.k.a "complete") or *conservative* (a.k.a "sound but incomplete"). In the context of $\ell_p$ norm-bounded perturbations, exact methods take a classifier $g$, input $x$, and radius $r$, and report whether or not there exists a perturbation $\delta$ within $\|\delta\| \leq r$ for which $g(x) \neq g(x + \delta)$. In contrast, conservative methods either certify that no such perturbation exists or decline to make a certification; they may decline even when it is true that no such perturbation exists. Exact methods are usually based on Satisfiability Modulo Theories [Katz et al., 2017, Carlini et al., 2017, Ehlers, 2017, Huang et al., 2017b] or mixed integer linear programming [Cheng et al., 2017, Lomuscio and Maganti, 2017, Dutta et al., 2017, Fischetti and Jo, 2017, Bunel et al., 2018]. Unfortunately, no exact methods have been shown to scale beyond moderate-sized (100,000 activations) networks [Tjeng et al., 2017], and networks of that size can only be verified when they are trained in a manner that impairs their expressivity.

Conservative certification is more scalable. Some conservative methods bound the *global* Lipschitz constant of the neural network [Gouk et al., 2018, Tsuzuku et al., 2018, Anil et al., 2019, Cisse et al., 2017], but these approaches tend to be very loose on expressive networks. Others measure the *local* smoothness of the network in the vicinity of a particular input $x$. In theory, one could obtain a robustness guarantee via an upper bound on the local Lipschitz constant of the network [Hein and Andriushchenko, 2017], but computing this quantity is intractable for general neural networks. Instead, a panoply of practical solutions have been proposed in the literature [Wong and Kolter, 2018, Wang et al., 2018a,b, Raghunathan et al., 2018a,b, Wong et al., 2018, Dvijotham et al., 2018b,a, Croce et al., 2019, Gehr et al., 2018, Mirman et al., 2018, Singh et al., 2018, Gowal et al., 2018, Weng et al., 2018a, Zhang et al., 2018]. Two themes stand out. Some approaches cast verification as an optimization problem and import tools such as relaxation and duality from the optimization literature to provide conservative guarantees [Wong and Kolter, 2018, Wong et al., 2018, Raghunathan et al., 2018a,b, Dvijotham et al., 2018b,a]. Others step through the network layer by layer, maintaining at each layer an outer approximation of the set of activations reachable by a perturbed input [Mirman et al., 2018, Singh et al., 2018, Gowal et al., 2018, Weng et al., 2018a, Zhang et al., 2018]. None of these local certification methods have been shown to be feasible on networks that are large and expressive enough to solve modern machine learning problems like the ImageNet classification task. Also, all either assume specific network architectures (e.g. ReLU activations or a layered feedforward structure) or require extensive customization for new network architectures.

**Related work involving noise** Prior works have proposed using a network's robustness to Gaussian noise as a proxy for its robustness to adversarial perturbations [Weng et al., 2018b, Ford et al., 2019], and have suggested that Gaussian data augmentation could supplement or replace adversarial training [Zantedeschi et al., 2017, Kannan et al., 2018]. Smilkov et al. [2017] observed that averaging a classifier's input gradients over Gaussian corruptions of an image yields very interpretable saliency maps. The robustness of neural networks to random noise has been analyzed both theoretically [Fawzi et al., 2016, Franceschi et al., 2018] and empirically [Dodge and Karam, 2017]. Finally, Webb et al. [2018] proposed a statistical technique for estimating the noise robustness of a classifier more efficiently than naive Monte Carlo simulation; we did not use this technique since it appears to lack formal high-probability guarantees. While these works hypothesized relationships between a neural network's robustness to random noise and *the same network's* robustness to adversarial perturbations, randomized smoothing instead uses a classifier's robustness to random noise *to create a new classifier* robust to adversarial perturbations.

**Randomized smoothing** Randomized smoothing has been studied previously for adversarial robustness. Several works [Liu et al., 2018, Cao and Gong, 2017] proposed similar techniques as heuristic defenses, but did not prove any guarantees. Lecuyer et al. [2019] used inequalities from the differential privacy literature to prove an $\ell_2$ and $\ell_1$ robustness guarantee for smoothing with Gaussian and Laplace noise, respectively. Subsequently, Li et al. [2018a] used tools from information theory to prove a stronger $\ell_2$ robustness guarantee for Gaussian noise. However, all of these robustness guarantees are loose. In contrast, we prove a tight robustness guarantee in $\ell_2$ norm for randomized smoothing with Gaussian noise.

## 2.3 Randomized Smoothing

Consider a classification problem from $\mathbb{R}^d$ to classes $\mathcal{Y}$. As discussed above, randomized smoothing is a method for constructing a new, "smoothed" classifier $g$ from an arbitrary base classifier $f$. When queried at $x$, the smoothed classifier $g$ returns whichever class the base classifier $f$ is most likely to return when $x$ is perturbed by isotropic Gaussian noise:

$$\begin{aligned} g(x) &= \arg\max_{c \in \mathcal{Y}} \ \mathbb{P}(f(x+\varepsilon) = c) \\ \text{where } \ \varepsilon &\sim \mathcal{N}(0, \sigma^2 I) \end{aligned} \tag{2.1}$$

An equivalent definition is that $g(x)$ returns the class $c$ whose pre-image $\{x' \in \mathbb{R}^d : f(x') = c\}$ has the largest probability measure under the distribution $\mathcal{N}(x, \sigma^2 I)$. The noise level $\sigma$ is a hyperparameter of the smoothed classifier $g$ which controls a robustness/accuracy tradeoff; it does not change with the input $x$. We leave undefined the behavior of $g$ when the argmax is not unique.

We will first present our robustness guarantee for the smoothed classifier $g$. Then, since it is not possible to exactly evaluate the prediction of $g$ at $x$ or to certify the robustness of $g$ around $x$, we will give Monte Carlo algorithms for both tasks that succeed with arbitrarily high probability.

### 2.3.1 Robustness Guarantee

Suppose that when the base classifier $f$ classifies $\mathcal{N}(x, \sigma^2 I)$, the most probable class $c_A$ is returned with probability $p_A$, and the "runner-up" class is returned with probability $p_B$. Our main result is that smoothed classifier $g$ is robust around $x$ within the $\ell_2$ radius $R = \frac{\sigma}{2}(\Phi^{-1}(p_A) - \Phi^{-1}(p_B))$, where $\Phi^{-1}$ is the inverse of the standard Gaussian CDF. This result also holds if we replace $p_A$ with a lower bound $\underline{p_A}$ and we replace $p_B$ with an upper bound $\overline{p_B}$.

**Theorem 2.3.1.** *Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$. Let $g$ be defined as in (2.1). Suppose $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0, 1]$ satisfy:*

$$\mathbb{P}(f(x+\varepsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}(f(x+\varepsilon) = c) \tag{2.2}$$

*Then $g(x+\delta) = c_A$ for all $\|\delta\|_2 < R$, where*

$$R = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})) \tag{2.3}$$

We now make several observations about Theorem 2.3.1:

- Theorem 2.3.1 assumes nothing about $f$. This is crucial since it is unclear which well-behavedness assumptions, if any, are satisfied by modern deep architectures.
- The certified radius $R$ is large when: (1) the noise level $\sigma$ is high, (2) the probability of the top class $c_A$ is high, and (3) the probability of each other class is low.
- The certified radius $R$ goes to $\infty$ as $\underline{p_A} \to 1$ and $\overline{p_B} \to 0$. This should sound reasonable: the Gaussian distribution is supported on all of $\mathbb{R}^d$, so the only way that $f(x+\varepsilon) = c_A$ with probability 1 is if $f = c_A$ almost everywhere.

Both Lecuyer et al. [2019] and Li et al. [2018a] proved $\ell_2$ robustness guarantees for the same setting as Theorem 2.3.1, but with different, smaller expressions for the certified radius. However, our $\ell_2$ robustness guarantee is *tight*: if (2.2) is all that is known about $f$, then it is impossible to certify an $\ell_2$ ball with radius larger than $R$. In fact, it is impossible to certify any superset of the $\ell_2$ ball with radius $R$:

**Theorem 2.3.2.** *Assume $\underline{p_A} + \overline{p_B} \leq 1$. For any perturbation $\delta$ with $\|\delta\|_2 > R$, there exists a base classifier $f$ consistent with the class probabilities (2.2) for which $g(x+\delta) \neq c_A$.*

Theorem 2.3.2 shows that Gaussian smoothing naturally induces $\ell_2$ robustness: if we make no assumptions on the base classifier beyond the class probabilities (2.2), then the set of perturbations to which a Gaussian-smoothed classifier is provably robust is *exactly* an $\ell_2$ ball.

The complete proofs of Theorems 2.3.1 and 2.3.2 are in Appendix A.1. We now sketch the proofs in the special case when there are only two classes.

**Theorem 1 (binary case).** *Suppose $\underline{p_A} \in (\frac{1}{2}, 1]$ satisfies $\mathbb{P}(f(x+\varepsilon) = c_A) \geq \underline{p_A}$. Then $g(x+\delta) = c_A$ for all $\|\delta\|_2 < \sigma\Phi^{-1}(\underline{p_A})$.*

*Proof sketch.* Fix a perturbation $\delta \in \mathbb{R}^d$. To guarantee that $g(x+\delta) = c_A$, we need to show that $f$ classifies the translated Gaussian $\mathcal{N}(x+\delta, \sigma^2 I)$ as $c_A$ with probability $> \frac{1}{2}$. However, all we know about $f$ is that $f$ classifies $\mathcal{N}(x, \sigma^2 I)$

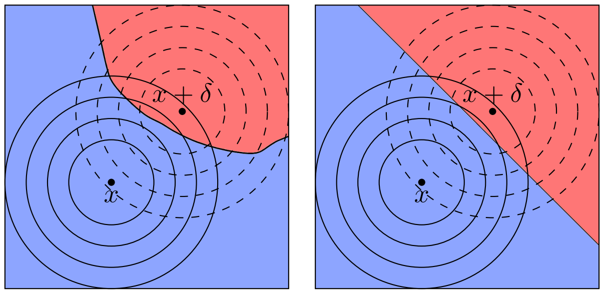


Figure 2.3: Illustration of $f^*$ in two dimensions. The concentric circles are the density contours of $\mathcal{N}(x, \sigma^2 I)$ and $\mathcal{N}(x + \delta, \sigma^2 I)$. Out of all base classifiers $f$ which classify $\mathcal{N}(x, \sigma^2 I)$ as $c_A$ (blue) with probability $\geq \underline{p_A}$, such as both classifiers depicted above, the "worst-case" $f^*$ — the one which classifies $\mathcal{N}(x + \delta, \sigma^2 I)$ as $c_A$ with minimal probability — is depicted on the right: a linear classifier with decision boundary normal to the perturbation $\delta$.

as $c_A$ with probability $\geq \underline{p_A}$. This raises the question: out of all possible base classifiers $f$ which classify $\mathcal{N}(x, \sigma^2 I)$ as $c_A$ with probability $\geq \underline{p_A}$, which one $f^*$ classifies $\mathcal{N}(x + \delta, \sigma^2 I)$ as $c_A$ with the smallest probability? One can show using an argument similar to the Neyman-Pearson lemma [Neyman and Pearson, 1933] that this "worst-case" $f^*$ is a linear classifier whose decision boundary is normal to the perturbation $\delta$ (Figure 2.3):

$$f^*(x') = \begin{cases} c_A & \text{if } \delta^T(x' - x) \leq \sigma \|\delta\|_2 \Phi^{-1}(\underline{p_A}) \\ c_B & \text{otherwise} \end{cases} \tag{2.4}$$

This "worst-case" $f^*$ classifies $\mathcal{N}(x + \delta, \sigma^2 I)$ as $c_A$ with probability $\Phi\big(\Phi^{-1}(\underline{p_A}) - \frac{\|\delta\|_2}{\sigma}\big)$. Therefore, to ensure that even the "worst-case" $f^*$ classifies $\mathcal{N}(x + \delta, \sigma^2 I)$ as $c_A$ with probability $> \frac{1}{2}$, we solve for those $\delta$ for which

$$\Phi\left(\Phi^{-1}(\underline{p_A}) - \frac{\|\delta\|_2}{\sigma}\right) > \frac{1}{2}$$

which is equivalent to the condition $\|\delta\|_2 < \sigma \Phi^{-1}(\underline{p_A})$. $\square$

Theorem 2.3.2 is a simple consequence: for any $\delta$ with $\|\delta\|_2 > R$, the base classifier $f^*$ defined in (2.4) is consistent with (2.2); yet if $f^*$ is the base classifier, then $g(x + \delta) = c_B$.

Figure 2.5 (left) plots our $\ell_2$ robustness guarantee against the guarantees derived in prior work. Observe that our $R$ is much larger than that of Lecuyer et al. [2019]

and moderately larger than that of Li et al. [2018a]. Appendix A.9 derives the other two guarantees using this paper's notation.

**Linear base classifier** A two-class linear classifier $f(x) = \text{sign}(w^T x + b)$ is already certifiable: the distance from any input $x$ to the decision boundary is $|w^T x + b|/\|w\|_2$, and no perturbation $\delta$ with $\ell_2$ norm less than this distance can possibly change $f$'s prediction. In Appendix A.2 we show that if $f$ is linear, then the smoothed classifier $g$ is identical to the base classifier $f$. Moreover, we show that our bound (2.3) will certify the true robust radius $|w^T x + b|/\|w\|$, rather than a smaller, overconservative radius. Therefore, when $f$ is linear, there always exists a perturbation $\delta$ just beyond the certified radius which changes $g$'s prediction.

**Noise level can scale with image resolution** Since our expression (2.3) for the certified radius does not depend explicitly on the data dimension $d$, one might worry that randomized smoothing is less effective for images of higher resolution — certifying a fixed $\ell_2$ radius is "less impressive" for, say, a $224 \times 224$ image than for a $56 \times 56$ image. However, as illustrated by Figure 2.4, images in higher resolution can tolerate higher levels $\sigma$ of isotropic Gaussian noise before their class-distinguishing content gets destroyed. As a consequence, in high resolution, smoothing can be performed with a larger $\sigma$, leading to larger certified radii. See Appendix A.7 for a more rigorous version of this argument.

### 2.3.2 Practical Algorithms

We now present practical Monte Carlo algorithms for evaluating $g(x)$ and certifying the robustness of $g$ around $x$. More details can be found in Appendix A.3.

#### Prediction

Evaluating the smoothed classifier's prediction $g(x)$ requires identifying the class $c_A$ with maximal weight in the categorical distribution $f(x+\varepsilon)$. The procedure described in pseudocode as PREDICT draws $n$ samples of $f(x + \varepsilon)$ by running

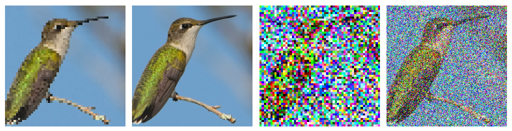


Figure 2.4: Left to right: clean 56 x 56 image, clean 224 x 224 image, noisy 56 x 56 image ($\sigma = 0.5$), noisy 224 x 224 image ($\sigma = 0.5$).

**Algorithm 1** Pseudocode for Certification and Prediction

```
# evaluate g at x
function PREDICT($f$, $\sigma$, $x$, $n$, $\alpha$)
    counts ← SAMPLEUNDERNOISE($f$, $x$, $n$, $\sigma$)
    $\widehat{c}_A, \widehat{c}_B$ ← top two indices in counts
    $n_A, n_B$ ← counts[$\widehat{c}_A$], counts[$\widehat{c}_B$]
    if BINOMPVALUE($n_A$, $n_A + n_B$, 0.5) $\leq \alpha$ return $\widehat{c}_A$
    else return ABSTAIN

# certify the robustness of g around x
function CERTIFY($f$, $\sigma$, $x$, $n_0$, $n$, $\alpha$)
    counts0 ← SAMPLEUNDERNOISE($f, x, n_0, \sigma$)
    $\widehat{c}_A$ ← top index in counts0
    counts ← SAMPLEUNDERNOISE($f, x, n, \sigma$)
    $\underline{p_A}$ ← LOWERCONFBOUND(counts[$\widehat{c}_A$], $n$, $1 - \alpha$)
    if $\underline{p_A} > \frac{1}{2}$ return prediction $\widehat{c}_A$ and radius $\sigma\, \Phi^{-1}(\underline{p_A})$
    else return ABSTAIN
```

$n$ noise-corrupted copies of $x$ through the base classifier. Let $\widehat{c}_A$ be the class which appeared the largest number of times. If $\widehat{c}_A$ appeared much more often than any other class, then PREDICT returns $\widehat{c}_A$. Otherwise, it abstains from making a prediction. We use the hypothesis test from Hung and Fithian [2019] to calibrate the abstention threshold so as to bound by $\alpha$ the probability of returning an incorrect answer. PREDICT satisfies the following guarantee:

**Proposition 2.3.3.** *With probability at least $1-\alpha$ over the randomness in* PREDICT, PREDICT *will either abstain or return $g(x)$. (Equivalently: the probability that* PREDICT *returns a class other than $g(x)$ is at most $\alpha$.)*

The function SAMPLEUNDERNOISE($f$, $x$, num, $\sigma$) in the pseudocode draws num samples of noise, $\varepsilon_1 \dots \varepsilon_{\text{num}} \sim \mathcal{N}(0, \sigma^2 I)$, runs each $x + \varepsilon_i$ through the base classifier $f$, and returns a vector of class counts. BINOMPVALUE($n_A$, $n_A + n_B$, $p$) returns the p-value of the two-sided hypothesis test that $n_A \sim \text{Binomial}(n_A + n_B, p)$.

Even if the true smoothed classifier $g$ is robust at radius $R$, PREDICT will be vulnerable in a certain sense to adversarial perturbations with $\ell_2$ norm slightly less than $R$. By engineering a perturbation $\delta$ for which $f(x + \delta + \varepsilon)$ puts mass just over $\frac{1}{2}$ on class $c_A$ and mass just under $\frac{1}{2}$ on class $c_B$, an adversary can force PREDICT to abstain at a high rate. If this scenario is of concern, a variant of Theorem 2.3.1 could be proved to certify a radius in which $\mathbb{P}(f(x+\delta+\varepsilon) = c_A)$ is larger by some margin than $\max_{c \neq c_A} \mathbb{P}(f(x + \delta + \varepsilon) = c)$.

**Certification**

Evaluating *and* certifying the robustness of $g$ around an input $x$ requires not only identifying the class $c_A$ with maximal weight in $f(x + \varepsilon)$, but also estimating a lower bound $\underline{p_A}$ on the probability that $f(x + \varepsilon) = c_A$ and an upper bound $\overline{p_B}$ on the probability that $f(x + \varepsilon)$ equals any other class. Doing all three of these at the same time in a statistically correct manner requires some care. One simple solution is presented in pseudocode as CERTIFY: first, use a small number of samples from $f(x + \varepsilon)$ to take a guess at $c_A$; then use a larger number of samples to estimate $\underline{p_A}$; then simply take $\overline{p_B} = 1 - \underline{p_A}$.

**Proposition 2.3.4.** *With probability at least $1 - \alpha$ over the randomness in* CERTIFY, *if* CERTIFY *returns a class $\widehat{c}_A$ and a radius $R$ (i.e. does not abstain), then $g$ predicts $\widehat{c}_A$ within radius $R$ around $x$: $g(x + \delta) = \widehat{c}_A \;\; \forall \; \|\delta\|_2 < R$.*

The function LOWERCONFBOUND($k$, $n$, $1 - \alpha$) in the pseudocode returns a one-sided $(1 - \alpha)$ lower confidence interval for the Binomial parameter $p$ given a sample $k \sim \text{Binomial}(n, p)$.

**Certifying large radii requires many samples** Recall from Theorem 2.3.1 that $R$ approaches $\infty$ as $\underline{p_A}$ approaches 1. Unfortunately, it turns out that $\underline{p_A}$ approaches 1 so slowly with $n$ that $R$ also approaches $\infty$ very slowly with $n$. Consider the most favorable situation: $f(x) = c_A$ everywhere. This means that $g$ is robust at radius $\infty$. But after observing $n$ samples of $f(x + \varepsilon)$ which all equal $c_A$, the tightest (to our knowledge) lower bound would say that with probability least $1 - \alpha$, $p_A \geq \alpha^{(1/n)}$. Plugging $\underline{p_A} = \alpha^{(1/n)}$ and $\overline{p_B} = 1 - \underline{p_A}$ into (2.3) yields an expression for the certified radius as a function of $n$: $R = \sigma \, \Phi^{-1}(\alpha^{1/n})$. Figure 2.5 (right) plots this function for $\alpha = 0.001, \sigma = 1$. Observe that certifying a radius of $4\sigma$ with 99.9% confidence would require $\approx 10^5$ samples.

### 2.3.3 Training the Base Classifier

Theorem 2.3.1 holds regardless of how the base classifier $f$ is trained. However, in order for $g$ to classify the labeled example $(x, c)$ correctly and robustly, $f$ needs to consistently classify $\mathcal{N}(x, \sigma^2 I)$ as $c$. In high dimension, the Gaussian distribution $\mathcal{N}(x, \sigma^2 I)$ places almost no mass near its mode $x$. As a consequence, when $\sigma$ is moderately high, the distribution of natural images has virtually disjoint support from the distribution of natural images corrupted by $\mathcal{N}(0, \sigma^2 I)$; see Figure 2.2 for a visual demonstration. Therefore, if the base classifier $f$ is trained via standard supervised learning on the data distribution, it will see no noisy images during

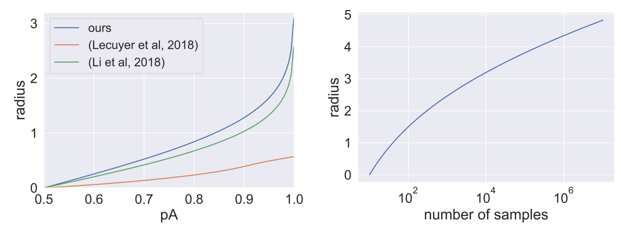


Figure 2.5: **Left**: Certified radius $R$ as a function of $\underline{p_A}$ (with $\overline{p_B} = 1 - \underline{p_A}$ and $\sigma = 1$) under all three randomized smoothing bounds. **Right**: A plot of $R = \sigma\,\Phi^{-1}(\alpha^{1/n})$ for $\alpha = 0.001$ and $\sigma = 1$. The radius we can certify with high probability grows slowly with the number of samples, even in the *best* case where $f(x) = c_A$ everywhere.

training, and hence will not necessarily learn to classify $\mathcal{N}(x, \sigma^2 I)$ with $x$'s true label. Therefore, in this paper we follow Lecuyer et al. [2019] and train the base classifier with Gaussian data augmentation at variance $\sigma^2$. A justification for this procedure is provided in Appendix A.6. However, we suspect that there may be room to improve upon this training scheme, perhaps by training the base classifier so as to maximize the smoothed classifier's certified accuracy at some tunable radius $r$.

## 2.4 Experiments

In adversarially robust classification, one metric of interest is the *certified test set accuracy* at radius $r$, defined as the fraction of the test set which $g$ classifies correctly with a prediction that is certifiably robust within an $\ell_2$ ball of radius $r$. However, if $g$ is a randomized smoothing classifier, computing this quantity exactly is not possible, so we instead report the *approximate certified test set accuracy*, defined as the fraction of the test set which CERTIFY classifies correctly (without abstaining) and certifies robust with a radius $R \geq r$. Appendix A.4 shows how to convert the approximate certified accuracy into a lower bound on the true certified accuracy that holds with high probability over the randomness in CERTIFY. However Appendix A.8.2 demonstrates that when $\alpha$ is small, the difference between these two quantities is negligible. Therefore, in our experiments we omit the step for simplicity and report approximate certified accuracies.

In all experiments, unless otherwise stated, we ran CERTIFY with $\alpha = 0.001$, so there was at most a 0.1% chance that CERTIFY returned a radius in which $g$ was not truly robust. Unless otherwise stated, when running CERTIFY we used $n_0 =$

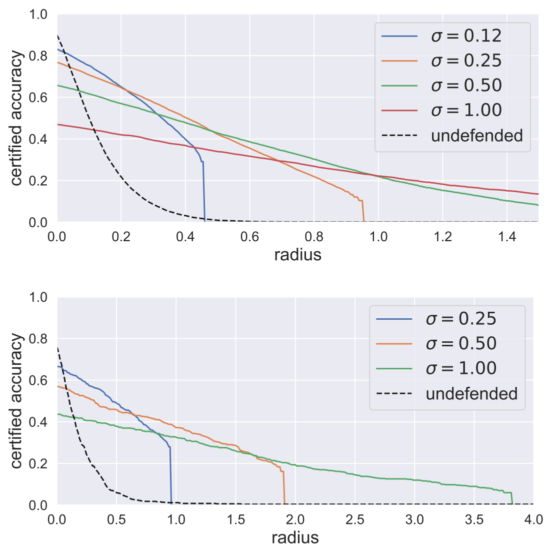


Figure 2.6: Approximate certified accuracy attained by randomized smoothing on CIFAR-10 (**top**) and ImageNet (**bottom**). The hyperparameter $\sigma$ controls a robustness/accuracy tradeoff. The dashed black line is an upper bound on the empirical robust accuracy of an undefended classifier with the base classifier's architecture.

100 Monte Carlo samples for selection and $n = 100{,}000$ samples for estimation.

In the figures above that plot certified accuracy as a function of radius $r$, the certified accuracy always decreases gradually with $r$ until reaching some point where it plummets to zero. This drop occurs because for each noise level $\sigma$ and number of samples $n$, there is a hard upper limit to the radius we can certify with high probability, achieved when all $n$ samples are classified by $f$ as the same class.

**ImageNet and CIFAR-10 results** We applied randomized smoothing to CIFAR-10 [Krizhevsky and Hinton, 2009] and ImageNet [Deng et al., 2009]. On each dataset we trained several smoothed classifiers, each with a different $\sigma$. On CIFAR-10 our base classifier was a 110-layer residual network; certifying each example took 15 seconds on an NVIDIA RTX 2080 Ti. On ImageNet our base classifier was a ResNet-50; certifying each example took 110 seconds. We also trained a neural network with the base classifier's architecture on clean data, and subjected it to a DeepFool $\ell_2$ adversarial attack [Moosavi-Dezfooli et al., 2016], in order to obtain an empirical upper bound on its robust accuracy. We certified the full CIFAR-10

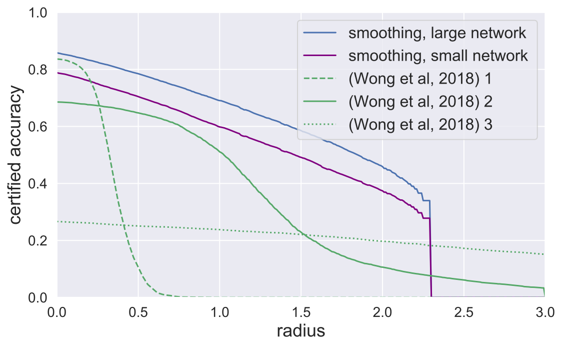


Figure 2.7: Comparison betwen randomized smoothing and Wong et al. [2018]. Each green line is a small resnet classifier trained and certified using the method of Wong et al. [2018] with a different setting of its hyperparameter $\epsilon$. The purple line is our method using the same small resnet architecture as the base classifier; the blue line is our method with a larger neural network as the base classifier. Wong et al. [2018] gives deterministic robustness guarantees, whereas smoothing gives high-probability guaranatees; therefore, we plot here the certified accuracy of Wong et al. [2018] against the "approximate" certified accuracy of smoothing.

test set and a subsample of 500 examples from the ImageNet test set.

Figure 2.6 plots the certified accuracy attained by smoothing with each $\sigma$. The dashed black line is the empirical upper bound on the robust accuracy of the base classifier architecture; observe that smoothing improves substantially upon the robustness of the undefended base classifier architecture. We see that $\sigma$ controls a robustness/accuracy tradeoff. When $\sigma$ is low, small radii can be certified with high accuracy, but large radii cannot be certified. When $\sigma$ is high, larger radii can be certified, but smaller radii are certified at a lower accuracy. This observation echoes the finding in Tsipras et al. [2018] that adversarially trained networks with higher robust accuracy tend to have lower standard accuracy. Tables of these results are in Appendix A.5.

Figure 2.8 (**left**) plots the certified accuracy obtained using our Theorem 2.3.1 guarantee alongside the certified accuracy obtained using the analogous bounds of Lecuyer et al. [2019] and Li et al. [2018a]. Since our expression for the certified radius $R$ is greater (and, in fact, tight), our bound delivers higher certified accuracies. Figure 2.8 (**middle**) projects how the certified accuracy would have changed had CERTIFY used more or fewer samples $n$ (under the assumption that the relative class proportions in `counts` would have remained constant). Finally, Figure 2.8 (**right**) plots the certified accuracy as the confidence parameter $\alpha$ is varied. Observe

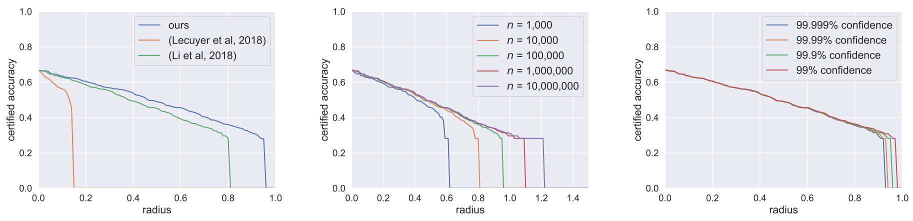


Figure 2.8: Experiments with randomized smoothing on ImageNet with $\sigma = 0.25$. **Left**: certified accuracies obtained using our Theorem 2.3.1 versus those obtained using the robustness guarantees derived in prior work. **Middle**: projections for the certified accuracy if the number of samples $n$ used by CERTIFY had been larger or smaller. **Right**: certified accuracy as the failure probability $\alpha$ of CERTIFY is varied.

that the certified accuracy is not very sensitive to $\alpha$.

**Comparison to baselines** We compared randomized smoothing to three baseline approaches for certified $\ell_2$ robustness: the duality approach from Wong et al. [2018], the Lipschitz approach from Tsuzuku et al. [2018], and the approach from Weng et al. [2018a], Zhang et al. [2018]. The strongest baseline was Wong et al. [2018]; we defer the comparison to the other two baselines to Appendix A.8.

In Figure 2.7, we compare the largest publicly released model from Wong et al. [2018], a small resnet, to two randomized smoothing classifiers: one which used the same small resnet architecture for its base classifier, and one which used a larger 110-layer resnet for its base classifier. First, observe that smoothing with the large 110-layer resnet substantially outperforms the baseline (across all hyperparameter settings) at all radii. Second, observe that smoothing with the small resnet also outperformed the method of Wong et al. [2018] at all but the smallest radii. We attribute this latter result to the fact that neural networks trained using the method of Wong et al. [2018] are "typically overregularized to the point that many filters/weights become identically zero," per that paper. In contrast, the base classifier in randomized smoothing is a fully expressive neural network.

**Prediction** It is computationally expensive to certify the robustness of $g$ around a point $x$, since the value of $n$ in CERTIFY must be very large. However, it is far cheaper to evaluate $g$ at $x$ using PREDICT, since $n$ can be small. For example, when we ran PREDICT on ImageNet ($\sigma = 0.25$) using $n = 100$, making each prediction only took 0.15 seconds, and we attained a top-1 test accuracy of 65% (Appendix A.5).

As discussed earlier, an adversary can potentially force PREDICT to abstain with high probability. However, it is relatively rare for PREDICT to abstain on the actual data distribution. On ImageNet ($\sigma = 0.25$), PREDICT with failure probability $\alpha = 0.001$ abstained 12% of the time when $n = 100$, 4% when $n = 1000$, and 1% when $n = 10{,}000$.

**Empirical tightness of bound** When $f$ is linear, there always exists a class-changing perturbation just beyond the certified radius. Since neural networks are not linear, we empirically assessed the tightness of our bound by subjecting an ImageNet smoothed classifier ($\sigma = 0.25$) to a projected gradient descent-style adversarial attack (Appendix A.10.3). For each example, we ran CERTIFY with $\alpha = 0.01$, and, if the example was correctly classified and certified robust at radius $R$, we tried finding an adversarial example for $g$ within radius $1.5R$ and within radius $2R$. We succeeded 17% of the time at radius $1.5R$ and 53% of the time at radius $2R$.

## 2.5 Conclusion

Theorem 2.3.2 establishes that smoothing with Gaussian noise naturally confers adversarial robustness in $\ell_2$ norm: if we have no knowledge about the base classifier beyond the distribution of $f(x + \varepsilon)$, then the set of perturbations to which the smoothed classifier is provably robust is precisely an $\ell_2$ ball. We suspect that smoothing with other noise distributions may lead to similarly natural robustness guarantees for other perturbation sets such as general $\ell_p$ norm balls.

Our strong empirical results suggest that randomized smoothing is a promising direction for future research into adversarially robust classification. Many empirical approaches have been "broken," and provable approaches based on certifying neural network classifiers have not been shown to scale to networks of modern size. It seems to be computationally infeasible to reason in any sophisticated way about the decision boundaries of a large, expressive neural network. Randomized smoothing circumvents this problem: the smoothed classifier is not itself a neural network, though it leverages the discriminative ability of a neural network base classifier. To make the smoothed classifier robust, one need simply make the base classifier classify well under noise. In this way, randomized smoothing reduces the unsolved problem of adversarially robust classification to the comparably solved domain of supervised learning.

Table 2.1: Approximate certified accuracy on ImageNet. Each row shows a radius $r$, the best hyperparameter $\sigma$ for that radius, the approximate certified accuracy at radius $r$ of the corresponding smoothed classifier, and the standard accuracy of the corresponding smoothed classifier. To give a sense of scale, a perturbation with $\ell_2$ radius 1.0 could change one pixel by 255, ten pixels by 80, 100 pixels by 25, or 1000 pixels by 8. Random guessing on ImageNet would attain 0.1% accuracy.

| $\ell_2$ RADIUS | BEST $\sigma$ | CERT. ACC (%) | STD. ACC(%) |
|---|---|---|---|
| 0.5 | 0.25 | 49 | 67 |
| 1.0 | 0.50 | 37 | 57 |
| 2.0 | 0.50 | 19 | 57 |
| 3.0 | 1.00 | 12 | 44 |

Chapter 3

# Certified Robustness to Label-Flipping Attacks via Randomized Smoothing

This chapter is based on Rosenfeld et al. [2020]:
Rosenfeld, E., Winston, E., Ravikumar, P., & Kolter, J. Z.
Certified Robustness to Label-Flipping Attacks via Randomized Smoothing.
In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

## 3.1 Introduction

Modern classifiers, despite their widespread empirical success, are known to be susceptible to adversarial attacks. In this paper, we are specifically concerned with so-called "data poisoning" attacks (formally, *causative* attacks [Barreno et al. 2006, Papernot et al. 2018]), where the attacker manipulates some aspects of the training data in order to cause the learning algorithm to output a faulty classifier. Automated machine-learning systems which rely on large, user-generated datasets—e.g. email spam filters, product recommendation engines, and fake review detectors—are particularly susceptible to such attacks. For example, by maliciously flagging legitimate emails as spam and mislabeling spam as innocuous, an adversary can trick a spam filter into mistakenly letting through a particular email.

Data poisoning attacks in the literature include label-flipping attacks [Xiao et al., 2012], where the labels of a training set can be adversarially manipulated to decrease performance of the trained classifier; general data poisoning, where both the training inputs and labels can be manipulated [Steinhardt et al., 2017]; and backdoor attacks [Chen et al., 2017, Tran et al., 2018], where the training set is corrupted so as to cause the classifier to deviate from its expected behavior only when triggered by a specific pattern. However, unlike the alternative test-time adversarial setting, where reasonably effective provable defenses exist, comparatively little work has been done on building classifiers that are certifiably robust to targeted data poisoning attacks.

In this work we propose a framework for building classifiers that are certifiably robust to a given class of data poisoning attacks, such as label-flipping or backdoor attacks. In particular, we propose what we refer to as a *pointwise* certified defense—this means that with each prediction, the classifier includes a certificate guaranteeing that its prediction would not be different had it been trained on adversarially manipulated data up to some "radius of perturbation" (a formal definition is presented in Section 3.3). We then demonstrate a specific instantiation of this protocol, constructing linear classifiers that are pointwise-certifiably robust to label-flipping attacks; i.e., each prediction is certified robust against a certain number of training label flips.

Prior works on certified defenses make statistical guarantees over the entire test distribution, but they make no guarantees as to the robustness of a prediction on any particular test point; thus, a determined adversary could still cause a specific test point to be misclassified. We therefore consider the threat of a worst-case adversary that can make a training set perturbation to target *each test point individually*. This motivates a defense that can certify each of its individual predictions, as we present here. Compared to traditional robust classification, this framework is superior for a task such as determining who receives a coveted resource (a loan, parole, etc.), as it provides a guarantee for each individual, rather than at the population level. This work represents the first such pointwise certified defense to *any type of data poisoning attack*; we expect significant advances can be made on both attacks and defenses within this threat model.

Our approach leverages randomized smoothing [Cohen et al., 2019], a technique that has previously been used to guarantee test-time robustness to adversarial manipulation of the input to a deep network. However, where prior uses of randomized smoothing randomize over the input to the classifier for test-time guarantees, we instead randomize over *the entire training procedure of the classifier*. Specifically, by randomizing over the labels during this training process, we obtain an overall classification pipeline that is certified to not change its prediction when some number of labels are adversarially manipulated in the training set. Previous applications of randomized smoothing perform sampling to provide probabilistic bounds, due to the intractability of integrating the decision regions of a deep network. We instead derive an analytical bound, providing truly guaranteed robustness. Although a naive implementation of this approach would not be tractable, we show how to obtain these certified bounds with minimal additional runtime complexity over standard classification, suffering only a linear cost in the number of training points.

A further distinction of our approach is that the applicability of our robustness guarantees do not rely upon stringent model assumptions or the quality of the features. Existing work on robust linear classification or regression provides certificates that only hold under specific model assumptions, e.g., recovering the best-fit linear coefficients, which is most useful when the data exhibit a linear relationship in the feature space. In contrast, our classifier makes *no assumptions* about the separability of the data or quality of the features; this means our certificates remain valid when applying our classifier to arbitrary features, which in practice allows us to leverage advances in unsupervised feature learning [Le, 2013, Chen et al., 2020] and transfer learning [Donahue et al., 2014]. We apply our classifier to pre-trained and unsupervised deep features to demonstrate its feasibility for classification of highly non-linear data such as ImageNet.

We evaluate our proposed classifier on several benchmark datasets common to the data poisoning literature. On the Dogfish binary classification challenge from ImageNet, our classifier maintains 81.3% certified accuracy in the face of an adversary who could reduce an undefended classifier to less than 1%. Additional experiments on MNIST and CIFAR10 demonstrate our algorithm's effectiveness for multi-class classification. Moreover, our classifier maintains a reasonably competitive non-robust accuracy (e.g., 94.5% on MNIST 1/7 versus 99.1% for the undefended classifier).

## 3.2 Related Work

**Data poisoning attacks** A *data poisoning attack* [Muñoz González et al., 2017, Yang et al., 2017] is an attack where an adversary corrupts some portion of the training set or adds new inputs, with the goal of degrading the performance of the learned model. The adversary is assumed to have perfect knowledge of the learning algorithm, so security by *design*—as opposed to obscurity—is the only viable defense against such attacks. The adversary is also typically assumed to have access to the training set and, in some cases, the test set.

Previous work has investigated attacks and defenses for data poisoning attacks applied to feature selection [Xiao et al., 2015], SVMs [Biggio et al., 2011, Xiao et al., 2012], linear regression [Liu et al., 2017], and PCA [Rubinstein et al., 2009], to name a few. Some attacks can even achieve success with "clean-label" attacks, inserting adversarially perturbed, seemingly correctly labeled training examples that cause the classifier to perform poorly [Shafahi et al., 2018, Zhu et al., 2019a].

Interestingly, our defense can also be viewed as (the first) certified defense to such attacks: perturbing an image such that the resulting features no longer match the label is theoretically equivalent to changing the label such that it no longer matches the image's features. For an overview of data poisoning attacks and defenses in machine learning, see Biggio et al. [2014].

**Label-flipping attacks** A *label-flipping attack* is a specific type of data poisoning attack where the adversary is restricted to changing the training labels. The classifier is then trained on the corrupted training set, with no knowledge of which labels have been tampered with. For example, an adversary could mislabel spam emails as innocuous, or flag real product reviews as fake.

Unlike random label noise, for which many robust learning algorithms have been successfully developed [Natarajan et al., 2013, Liu and Tao, 2016, Patrini et al., 2017], adversarial label-flipping attacks can be specifically targeted to exploit the structure of the learning algorithm, significantly degrading performance. Robustness to such attacks is therefore harder to achieve, both theoretically and empirically [Xiao et al., 2012, Biggio et al., 2011]. A common defense technique is *sanitization*, whereby a defender attempts to identify and remove or relabel training points that may have had their labels corrupted [Paudice et al., 2019, Taheri et al., 2019]. Unfortunately, recent work has demonstrated that this is often not enough against a sufficiently powerful adversary [Koh et al., 2018]. Further, *no existing defenses* provide pointwise guarantees regarding their robustness.

**Certified defenses** Existing works on certified defenses to adversarial data poisoning attacks typically focus on the regression case and provide broad statistical guarantees over the entire test distribution. A common approach to such certifications is to show that a particular algorithm recovers some close approximation to the best linear fit coefficients [Diakonikolas et al., 2019, Prasad et al., 2018, Shen and Sanghavi, 2019], or that the expected loss on the test distribution is bounded [Klivans et al., 2018, Chen and Paschalidis, 2018]. These results generally rely on assumptions on the data distribution: some assume sparsity in the coefficients [Karmalkar and Price, 2018, Chen et al., 2013] or corruption vector [Bhatia et al., 2015]; others require limited effects of outliers [Steinhardt et al., 2017]. As mentioned above, all of these methods fail to provide guarantees for individual test points. Additionally, these statistical guarantees are not as meaningful when their model assumptions do not hold.

**Randomized smoothing** Since the discovery of adversarial examples [Szegedy et al., 2013a, Goodfellow et al., 2015], the research community has been investigating techniques for increasing the adversarial robustness of complex models such as deep networks. After a series of heuristic defenses, followed by attacks breaking them [Athalye et al., 2018, Carlini and Wagner, 2017], focus began to shift towards the development of *provable* robustness.

One approach which has gained popularity in recent work is randomized smoothing. Rather than certifying the original classifier $f$, randomized smoothing defines a new classifier $g$ whose prediction at an input $x$ is the class assigned the most probability when $x$ is perturbed with noise from some distribution $\mu$ and passed through $f$. That is, $g(x) = \arg\max_c \mathbb{P}_{\epsilon\sim\mu}(f(x+\epsilon)=c)$. This new classifier $g$ is then certified as robust, ideally without sacrificing too much accuracy compared to $f$. The original formulation was presented by Lecuyer et al. [2019] and borrowed ideas from differential privacy. The above definition is due to Li et al. [2018a] and was popularized by Cohen et al. [2019], who derived a tight robustness guarantee.

## 3.3 A General View of Randomized Smoothing

Our first contribution is a general viewpoint of randomized smoothing, unifying all existing applications of the framework. Under our notation, randomized smoothing constructs an operator $G(\mu, \phi)$ that maps a binary-valued[1] function $\phi : \mathcal{X} \to \{0, 1\}$ and a *smoothing measure* $\mu : \mathcal{X} \to \mathbb{R}_+$, with $\int_{\mathcal{X}} \mu(x)dx = 1$, to the expected value of $\phi$ under $\mu$ (that is, $G(\mu, \phi)$ represents the "vote" of $\phi$ weighted by $\mu$). For example, $\phi$ could be a binary image classifier and $\mu$ could be some small, random pixel noise applied to the to-be-classified image. We also define a "hard threshold" version $g(\mu, \phi)$ that returns the most probable output (the majority vote winner). Formally,

$$\begin{aligned} G(\mu, \phi) &= \mathbb{E}_{x\sim\mu}[\phi(x)] = \int_{\mathcal{X}} \mu(x)\phi(x)dx, \\ g(\mu, \phi) &= \mathbf{1}\{G(\mu, \phi) \geq 1/2\}, \end{aligned}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. Where it is clear from context, we will omit the arguments, writing simply $G$ or $g$. Intuitively, for two similar measures $\mu, \rho$, we would expect that for most $\phi$, even though $G(\mu, \phi)$ and $G(\rho, \phi)$ may not be equal, the threshold function $g$ should satisfy $g(\mu, \phi) = g(\rho, \phi)$. Further, the degree to which $\mu$ and $\rho$ can differ while still preserving this property should increase as $G(\mu, \phi)$ approaches either 0 or 1, because this increases the "margin" with which

[1]For simplicity, we present the methodology here with binary-valued functions, which will correspond eventually to binary classification problems. The extension to the multi-class setting requires additional notation, and thus is deferred to the appendix.

the function $\phi$ is 0 or 1 respectively over the measure $\mu$. More formally, we define a general randomized smoothing guarantee as follows:

**Definition 3.3.1.** Let $\mu : \mathcal{X} \to \mathbb{R}_+$ be a smoothing measure over $\mathcal{X}$, with $\int_{\mathcal{X}} \mu(x)dx = 1$.[2] Then a randomized smoothing robustness guarantee is a specification of a distance function over probability measures $d(\mu, \rho)$ and a function $f : [0, 1] \to \mathbb{R}_+$ such that for all $\phi : \mathcal{X} \to \{0, 1\}$,

$$g(\mu, \phi) = g(\rho, \phi) \text{ whenever } d(\mu, \rho) \leq f(G(\mu, \phi)). \tag{3.1}$$

Informally, equation 3.1 says that the majority vote winner of $\phi$ weighted by $\mu$ and $\rho$ will be the same, so long as $\mu$ and $\rho$ are "close enough" as a function of the margin with which the majority wins. We will sometimes use $p$ in place of $G(\mu, \phi)$, representing the fraction of the vote that the majority class receives (analogous to $p_A$ in Cohen et al. [2019]).

**Instantiations of randomized smoothing** This definition is rather abstract, so we highlight concrete examples of how it can be applied to achieve certified guarantees against adversarial attacks.

*Example* 3.3.2. The randomized smoothing guarantee of Cohen et al. [2019] uses the smoothing measures $\mu = \mathcal{N}(x_0, \sigma^2 I)$, a Gaussian aroound the point $x_0$ to be classified, and $\rho = \mathcal{N}(x_0 + \delta, \sigma^2 I)$, the same measure perturbed by $\delta$. They prove that (3.1) holds for all classifiers $\phi$ if we define

$$d(\mu, \rho) = \frac{1}{\sigma}\|\delta\|_2 \equiv \sqrt{2\text{KL}(\mu \parallel \rho)}, \quad f(p) = |\Phi^{-1}(p)|,$$

where $\text{KL}(\cdot)$ denotes KL divergence and $\Phi^{-1}$ denotes the inverse CDF of the Gaussian distribution.

Although this work focused on the case of randomized smoothing of continuous data via Gaussian noise, this is by no means a requirement. Lee et al. [2019] consider an alternative approach for dealing with discrete variables.

*Example* 3.3.3. The randomized smoothing guarantee of Lee et al. [2019] uses the factorized smoothing measure in $d$ dimensions $\mu_{\alpha,K}(\mathbf{x}) = \Pi_{i=1}^d \mu_{\alpha,K,i}(\mathbf{x}_i)$, defined with respect to parameters $\alpha \in [0, 1], K \in \mathbb{N}$, and a base input $\mathbf{z} \in \{0, \ldots, K\}^d$, where

$$\mu_{\alpha,K,i}(\mathbf{x}_i) = \begin{cases} \alpha, & \text{if } \mathbf{x}_i = \mathbf{z}_i \\ \frac{1-\alpha}{K}, & \text{if } \mathbf{x}_i \in \{0, \ldots, K\},\ \mathbf{x}_i \neq \mathbf{z}_i, \end{cases}$$

[2]There is no theoretical reason to restrict $\mu$ to be a probability measure. While this and all previous works only consider probability measures, the framework we present here could easily be extended to allow for more general measures $\mu, \rho$ and functions $\phi$.

with $\mathbf{x}_i$ being the $i^{th}$ dimension of $\mathbf{x}$. $\rho_{\alpha,K}$ is similarly defined for a perturbed input $\mathbf{z}'$. They guarantee that (3.1) holds if we define

$$d(\mu, \rho) = \|\mathbf{z}' - \mathbf{z}\|_0, \quad f(p) = \mathcal{F}_{\alpha,K,d}(\max(p, 1-p)). \tag{3.2}$$

In words, the smoothing distribution is such that each dimension is independently perturbed to one of the other $K$ values uniformly at random with probability $1-\alpha$. $\mathcal{F}_{\alpha,K,d}(p)$ is a combinatorial function defined as the maximum number of dimensions—out of $d$ total—by which $\mu_{\alpha,K}$ and $\rho_{\alpha,K}$ can differ such that a set with measure $p$ under $\mu_{\alpha,K}$ is guaranteed to have measure at least $\frac{1}{2}$ under $\rho_{\alpha,K}$. Lee et al. [2019] prove that this value depends only on $\|\mathbf{z}' - \mathbf{z}\|_0$.

Finally, Dvijotham et al. [2020] consider a more general form of randomized smoothing that doesn't require strict assumptions on the distributions but is still able to provide similar guarantees.

*Example* 3.3.4 (Generic bound). Given any two smoothing distributions $\mu, \rho$, we have the generic randomized smoothing robustness certificate, ensuring that (3.1) holds with definitions

$$d(\mu, \rho) = \mathrm{KL}(\rho \parallel \mu), \quad f(p) = -\frac{1}{2}\log(4p(1-p)). \tag{3.3}$$

**Randomized smoothing in practice** For deep classifiers, the expectation $G$ cannot be computed exactly, so we must resort to Monte Carlo approximation. This is done by drawing samples from $\mu$ and using these to construct a high-probability bound on $G$ for certification. More precisely, this bound should be a *lower* bound on $G$ when the hard prediction $g = 1$ and an *upper* bound otherwise; this ensures in both cases that we under-certify the true robustness of the classifier $g$. The procedure is shown in Algorithm 8 in Appendix B.1. These estimates can then be plugged into a randomized smoothing robustness guarantee to provide a high probability certified robustness bound for the classifier.

## 3.4 Pointwise Data Poisoning Robustness

We now present the main contributions of this paper: we first describe a generic strategy for applying randomized smoothing to certify a prediction function against arbitrary classes of data poisoning attacks. We then propose a specific implementation of said strategy to certify a classifier against label-flipping attacks. We show how this approach can be made tractable using linear least-squares classification, and we use the Chernoff inequality to analytically bound the relevant probabilities for the randomized smoothing certificate. Notably, although we are employing a randomized approach, the final algorithm does not use any random sampling, but rather relies upon a convex optimization problem to compute the certified robustness.

**General data poisoning robustness** We begin by noting that in prior work, randomized smoothing was applied at test time with the function $\phi : \mathcal{X} \to \{0, 1\}$ being a (potentially deep) classifier that we wish to smooth. However, there is no requirement that the function $\phi$ be a classifier at all; the theory holds for any binary-valued function. Instead of treating $\phi$ as a trained classifier, we consider $\phi$ to be *an arbitrary learning algorithm* which takes as input a training dataset $\{x_i, y_i\}_{i=1}^n \in (\mathcal{X} \times \{0, 1\})^n$ and additional test points without corresponding labels, which we aim to predict.[3] In other words, the combined goal of $\phi$ is to first train a classifier and then predict the label of the new example. Thus, we consider test time outputs to be a function of both the test time input and the training data that produced the classifier. This perspective allows us to reason about how changes to training data affect the classifier at test time, reminiscent of work on influence functions of deep neural networks [Koh and Liang, 2017, Yeh et al., 2018].

This immediately suggests our protocol for pointwise robustness to general data poisoning attacks: randomize over the elements of the input to which we desire certified robustness, rather than over the test-time input to be classified. For example, to induce robustness to backdoor attacks, we could randomly add noise to the training points and/or their labels. Analogous to previous applications of randomized smoothing, if the majority vote of the classifiers trained with these randomly perturbed inputs has a large margin, it will confer a degree of robustness within an appropriately-defined radius of adversarially perturbed training data (as defined in Equation (3.1)).

**Specific application to label-flipping robustness** To demonstrate the effectiveness of our proposed strategy, we now present a specific implementation, providing an algorithm for tractable linear classification which is pointwise-certifiably robust to label-flipping attacks. When applying randomized smoothing in this setting, we randomize over the labels in the training set as described above—a suitably large margin in the majority vote will therefore result in pointwise robustness to adversarial label flips. In this scenario, the adversarial "radius" is defined as number of labels on which two training sets differ.

To formalize this intuition, consider two different assignments of $n$ training labels $Y_1, Y_2 \in \{0, 1\}^n$ which differ on precisely $r$ labels. Let $\mu$ (resp. $\rho$) be the distribution resulting from independently flipping each of the labels in $Y_1$ (resp. $Y_2$) with probability $q$. It is clear that as $r$ increases, $\mathrm{KL}(\mu \parallel \rho)$ will also increase. In fact, it is simple to show (see Appendix B.2.3 for derivation) that the exact KL

[3]Note that our algorithm does not require access to the test data to do the necessary precomputation. We present it here as such merely to give an intuitive idea of the procedure.

divergence between these two distributions is

$$\mathrm{KL}(\mu \parallel \rho) = \mathrm{KL}(\rho \parallel \mu) = r(1-2q)\log\left(\frac{1-q}{q}\right). \tag{3.4}$$

Plugging in the robustness guarantee (3.3), we have that $g(\mu, \phi) = g(\rho, \phi)$ so long as

$$r \leq \frac{\log(4p(1-p))}{2(1-2q)\log\left(\frac{q}{1-q}\right)}, \tag{3.5}$$

where $p = G(\mu, \phi)$. This implies that for any test point, as long as (3.5) is satisfied, $g$'s prediction (the majority vote weighted by the smoothing distribution) will not change if an adversary corrupts the training set from $Y_1$ to $Y_2$, or indeed to any other training set that differs on at most $r$ labels. We can tune the noise hyperparameter $q$ to achieve the largest possible upper bound in (3.5); more noise will likely decrease the margin of the majority vote $p$, but it will also decrease the divergence.

**Computing a tight bound** This approach has a simple closed form, but the bound is not tight. We can derive a tight bound via a combinatorial approach as in Lee et al. [2019]. By precomputing the quantities $\mathcal{F}^{-1}_{1-q,1,n}(r)$ from Equation (3.2) for each $r$, we can simply compare $p$ to each of these and thereby certify robustness to the highest possible number of label flips. This precomputation can be expensive, but it provides a significantly tighter robustness guarantee, certifying approximately twice as many label flips for a given bound on $G$ (See Figure B.4 in the Appendix).

### 3.4.1 Efficient Implementation via Least Squares Classifiers

There may appear to be one major impracticality of the algorithm proposed in the previous section, if considered naively: treating the function $\phi$ as an entire training-plus-single-prediction process would require that we train multiple classifiers, over multiple random draws of the labels $y$, all to make a prediction on a single example. In this section, we describe a sequence of tools we employ to restrict the architecture and training process in a manner that drastically reduces this cost, bringing it in line with the traditional cost of standard classification. The full procedure, with all the parts described below, can be found in Algorithm 2.

**Linear least-squares classification** The fundamental simplification we make in this work is to restrict the "training" of the classifier $\phi$ to be done via the solution of a least-squares problem. Given the training set $\{x_i, y_i\}_{i=1}^n$, we assume that there exists some feature mapping $h : \mathbb{R}^d \to \mathbb{R}^k$ (where $k < n$). If existing linear features are not available, this could instead consist of deep features learned from

**Algorithm 2** Randomized smoothing for label-flipping robustness

```
**Input:** feature mapping $h : \mathbb{R}^d \to \mathbb{R}^k$; noise parameter $q$; regularization parameter $\lambda$; training set $\{(x_i, y_i) \in \mathbb{R}^d \times \{0,1\}\}_{i=1}^n$ (with potentially adversarial labels); additional inputs to predict $\{x_j \in \mathbb{R}^d\}_{j=1}^m$.
1. Pre-compute matrix $\mathbf{M}$,
   $$\mathbf{M} = \boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}\right)^{-1}$$
where $\boldsymbol{X} \equiv h(x_{1:n})$.
**for** $j = 1, \ldots, m$ **do**
    1. Compute vector $\boldsymbol{\alpha}^j = \mathbf{M}h(x_j)^T$.
    2. Compute optimal Chernoff parameter $t$ via Newton's method
       $$t^\star = \arg\min_t \left\{ t/2 + \sum_{i:y_i=1} \log\left(q + (1-q)e^{-t\boldsymbol{\alpha}_i^j}\right) + \sum_{i:y_i=0} \log\left((1-q) + qe^{-t\boldsymbol{\alpha}_i^j}\right) \right\}$$
    and let $p^\star = \max(1 - B_{|t^\star|}, 1/2)$ where $B_{|t^\star|}$ is the Chernoff bound (3.6) evaluated at $|t^\star|$.
    **Output:** Prediction $\widehat{y}_j = \mathbf{1}\{t^\star \geq 0\}$ and certification that prediction will remain constant for up to $r$ training label flips, where
       $$r = \left\lfloor \frac{\log(4p^\star(1-p^\star))}{2(1-2q)\log\left(\frac{q}{1-q}\right)} \right\rfloor.$$
**end for**
```

a similar task—the transferability of such features is well documented [Donahue et al., 2014, Bo et al., 2010, Yosinski et al., 2014]—or features could be learned in an unsupervised fashion on $x_{1:n}$ (learning the features from poisoned labels could degrade performance). Given this feature mapping, let $\boldsymbol{X} = h(x_{1:n}) \in \mathbb{R}^{n \times k}$ be the training point features and let $\mathbf{y} = y_{1:n} \in \{0,1\}^n$ be the labels. Our training process consists of finding the least-squares fit to the training data, i.e., we find parameters $\widehat{\boldsymbol{\beta}} \in \mathbb{R}^k$ via the normal equation $\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\mathbf{y}$ and then we make a prediction on the new example via the linear function $h(x_{n+1})\widehat{\boldsymbol{\beta}}$. Although it may seem odd to fit a classification task with least-squares loss, binary classification with linear regression is equivalent to Fisher's linear discriminant [Mika, 2003] and works quite well in practice.

The real advantage of the least-squares approach is that it reduces the prediction to a linear function of $\mathbf{y}$, and thus randomizing over the labels is straightforward. Specifically, letting

$$\boldsymbol{\alpha} = \boldsymbol{X} \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} h(x_{n+1})^T,$$

the prediction $h(x_{n+1})\widehat{\boldsymbol{\beta}}$ can be equivalently given by $\boldsymbol{\alpha}^T \mathbf{y}$ (this is effectively the kernel representation of the linear classifier). Thus, we can simply compute $\boldsymbol{\alpha}$ one time and then randomly sample many different sets of labels in order to build a standard randomized smoothing bound. Further, we can pre-compute just the $\boldsymbol{X} \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1}$ term and reuse it for each test point.

**$\ell_2$ regularization for better conditioning** It is unlikely to be the case that the training points are well-behaved for linear classification in the feature space. To address this, we instead solve an $\ell_2$ regularized version of least-squares. This is a common tool for solving systems with ill-conditioned or random design matrices [Hsu et al., 2014, Suggala et al., 2018]. Luckily, there still exists a pre-computable closed-form solution to this problem, whereby we instead solve

$$\boldsymbol{\alpha} = \boldsymbol{X}(\boldsymbol{X}^T \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} h(x_{n+1})^T.$$

The other parts of our algorithm remain unchanged. Following results in Suggala et al. [2018], we set the regularization parameter $\lambda = (1+q)\frac{\widehat{\sigma}^2 k}{2n}\kappa(\boldsymbol{X}^T \boldsymbol{X})$ for all our experiments, where $\widehat{\sigma}^2 = \frac{\|\mathbf{y} - \boldsymbol{X}\widehat{\boldsymbol{\beta}}_{OLS}\|_2^2}{n-k}$ is an estimate of the variance [Dicker, 2014] and $\kappa(\cdot)$ is the condition number.

**Efficient tail bounds via the Chernoff inequality** Even more compelling, due to the linear structure of this prediction, we can forego a sampling-based approach entirely and directly bound the tail probabilities using Chernoff bounds. Because the underlying binary prediction function $\phi$ will output the label 1 for the test point whenever $\boldsymbol{\alpha}^T \mathbf{y} \geq 1/2$ and 0 otherwise, we can derive an analytical upper bound on the probability that $g$ predicts one label or the other via the Chernoff bound. By upper bounding the probability of the *opposite* prediction, we simultaneously derive a lower bound on $p$ which can be plugged in to (3.5) to determine the classifier's robustness. Concretely, we can upper bound the probability that the classifier outputs the label 0 by

$$P(\boldsymbol{\alpha}^T \mathbf{y} \leq 1/2) \leq \min_{t>0} \left\{ e^{t/2} \prod_{i=1}^{n} E[e^{-t\boldsymbol{\alpha}_i y_i}] \right\}$$
$$= \min_{t>0} \left\{ e^{t/2} \prod_{i=1}^{n} q e^{-t\boldsymbol{\alpha}_i(1-y_i)} + (1-q) e^{-t\boldsymbol{\alpha}_i y_i} \right\}. \quad (3.6)$$

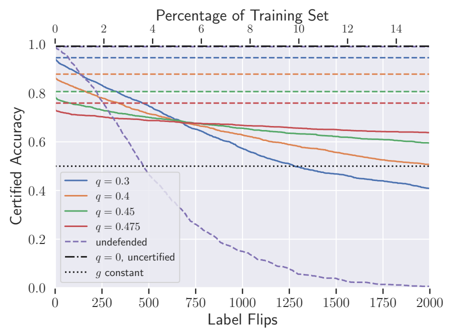
Conversely, the probability that the classifier outputs the label 1 is upper bounded by (3.6) but evaluated at $-t$. Thus, we can solve the minimization problem unconstrained over $t$, and then let the sign of $t$ dictate which label to predict and the value of $t$ determine the bound. The objective (3.6) is log-convex in $t$ and can be easily solved by Newton's method. Note that in some cases, neither Chernoff upper bound will be less than $1/2$, meaning we cannot determine the true value of $g$. In these cases, we simply define the classifier's prediction to be determined by the sign of $t$. While we can't guarantee that this classification will match the true majority vote, our algorithm will certify a robustness to 0 flips, so the guarantee is still valid. We avoid abstaining so as to assess our classifier's non-robust accuracy.

The key property we emphasize is that, unlike previous randomized smoothing applications, the final algorithm involves *no randomness whatsoever*. Instead, the probabilities are bounded directly via the Chernoff bound, without any need for Monte Carlo approximation. Thus, the method is able to generate *truly certifiable* robust predictions using approximately the same complexity as traditional predictions.

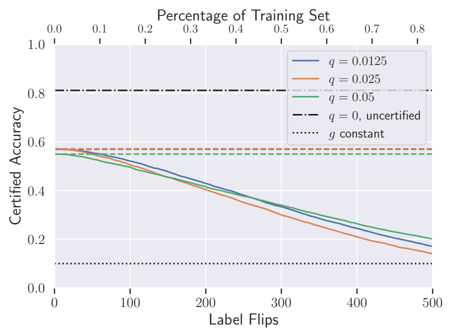
## 3.5 Experiments

Following Koh and Liang [2017] and Steinhardt et al. [2017], we perform experiments on MNIST 1/7, the IMDB review sentiment dataset [Maas et al., 2011], and the Dogfish binary classification challenge taken from ImageNet. We run additional experiments on multi-class MNIST and CIFAR10. For each dataset and each noise level $q$ we report the *certified test set accuracy* at $r$ training label flips. That is, for each possible number of flips $r$, we plot the fraction of the test set that was both correctly classified and certified to not change under at least $r$ flips.

As mentioned, our classifier suffers an additional linear cost in the number of training points due to the kernel representation $\boldsymbol{\alpha}^T \mathbf{y}$. For most datasets there was no discernible difference in the time required to certify an input via our technique versus neural network classification. For larger training sets such as CIFAR10, especially when doing pairwise comparisons for the multi-class case, the algorithm is embarassingly parallel; this parallelism brings runtime back in line with standard classification.



*(a)* Binary MNIST (classes 1 and 7)



*(b)* Full MNIST

Figure 3.1: MNIST 1/7 ($n = 13007$, top) and full MNIST ($n = 60000$, bottom) test set certified accuracy to adversarial label flips as $q$ is varied. The bottom axis represents the number of adversarial label flips to which *each individual prediction* is robust, while the top axis is the same value expressed as a percentage of the training set size. The solid lines represent certified accuracy; dashed lines of the same color are the overall non-robust accuracy of each classifier. The black dotted line is the (infinitely robust) performance of a constant classifier, while the black dash-dot line is the (uncertified) performance of our classifier with no label noise.

For binary classification, one could technically achieve a certified accuracy of 50% at $r = \infty$ (or 10% for MNIST or CIFAR10) by letting $g$ be constant—a constant classifier would be infinitely robust. Though not a very meaningful baseline, we include the accuracy of such a classifier in our plots (black dotted line) as a reference. We also evaluated our classifier with $q = 0$ (black dash-dot line); this cannot certify robustness, but it indicates the quality of the features.

To properly justify the need for such certified defenses, and to get a sense of the scale of our certifications, we generated label-flipping attacks against the undefended binary MNIST and Dogfish models. Following previous work, the undefended models were implemented as convolutional neural networks, trained on the clean data, with all but the top layer frozen—this is equivalent to multinomial logistic regression on the learned features. For each test point we recorded how many flips were required to change the network's prediction. This number serves as an upper bound for the robustness of the network on that test point, but we note that our attacks were quite rudimentary and could almost certainly be improved upon to tighten this upper bound. Appendix B.3.1 contains the details of our attack implementations. Finally, we implemented attacks on our own defense to derive an empirical upper bound and found that it reasonably tracks our lower bound. Plots and details of this attack can be found in Appendix B.3.2.

In all plots, the solid lines represent certified accuracy (except for the undefended classifier, which is an upper bound), while the dashed lines of the same color are the overall non-robust accuracy of each classifier.

**Results on MNIST** The MNIST 1/7 dataset [LeCun et al., 1998] consists of just the classes 1 and 7, totalling 13,007 training points and 2,163 test points. We trained a simple convolutional neural network on the other eight MNIST digits to learn a 50-dimensional feature embedding and then calculated Chernoff bounds for $G$ as described in Section 3.4.1. Figure 3.1a displays the certified accuracy on the test set for varying probabilities $q$. As in prior work on randomized smoothing, the noise parameter $q$ balances a trade-off; as $q$ increases, the required margin $|G - \frac{1}{2}|$ to certify a given number of flips decreases. On the other hand, this results in more noisy training labels, which reduces the margin and therefore results in lower robustness and often lower accuracy. Figure 3.1b depicts the certified accuracy for the full MNIST test set—see Appendix B.2 for derivations of the bounds and optimization algorithm in the multi-class case. In addition to this being a significantly more difficult classification task, our classifier could not rely on features learned from other handwritten digits; instead, we extracted the top 30 components with ICA [Hyvarinen, 1999] independently of the labels. Despite the lack of fine-tuned features, our algorithm still achieves significant certified accuracy under a large number of adversarial label flips.

See Figure B.2 in the Appendix for the effect of $\ell_2$ regularization for the binary case. At a moderate cost to non-robust accuracy, the regularization results in substantially higher certified accuracy at almost all radii. We observed that regularization did not make a large difference for the multi-class case, possibly due to the inaccuracy of the residual term in the noise estimate.

**Results on CIFAR10** To further demonstrate the effectiveness of our classifier with unsupervised features, we used SimCLR [Chen et al., 2020] to learn unsupervised features for CIFAR10. We used PCA to reduce the features to 128 dimensions to reduce overfitting. Figure 3.2 shows the results: our classifier with $q = 0.12$ achieves 50% certified accuracy up to 175 labels flips (recall there are ten classes, not two) and decays gracefully. Further, the classifier maintains better than random chance certified accuracy up to 427 label flips, which is approximately 1% of the training set.

Because the "votes" are changed by flipping so few labels, high values of $q$ reduce the models' predictions to almost pure chance—this means we are unable to achieve the margins necessary to certify a large number of flips. We therefore found that smaller levels of noise achieved higher certified test accuracy. This suggests

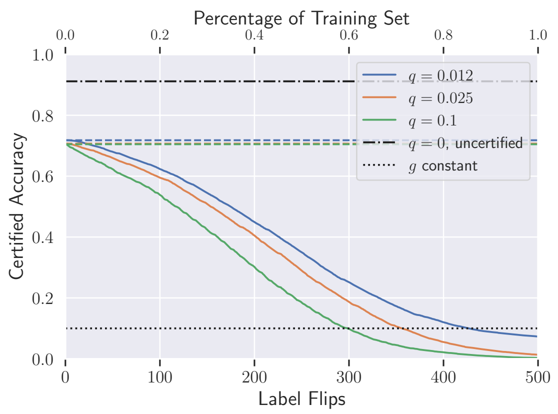


Figure 3.2: CIFAR10 ($n = 50000$) test set certified accuracy to adversarial label flips as $q$ is varied.

that the more susceptible the original, non-robust classifier is to label flips, the lower $q$ should be set for the corresponding randomized classifier.

For much smaller values of $q$, slight differences did not decrease the non-robust accuracy—they did however have a large effect on certified robustness. This indicates that the sign of $t^\star$ is relatively stable, but the margin of $G$ is much less so. This same pattern was observed with the IMDB and Dogfish datasets. We used a high-precision arithmetic library [Johansson et al., 2013] to achieve the necessary lower bounds, but the precision required for non-vacuous bounds grew extremely fast for $q < 10^{-4}$; optimizing (3.6) quickly became too computationally expensive.

**Results on Dogfish** The Dogfish dataset contains images from the ImageNet dog and fish synsets, 900 training points and 300 test points from each. We trained a ResNet-50 [He et al., 2016] on the standard ImageNet training set but removed all images labeled dog or fish. Our pre-trained network therefore learned meaningful image features but had no features specific to either class. We again used PCA to reduce the feature space dimensionality. Figure 3.3 displays the results of our poisoning attack along with our certified defense. Under the undefended model, more than 99% of the test points can be successfully attacked with no more than 23 label flips, whereas our model with $q = 10^{-4}$ can certifiably correctly classify 81.3% of the test points under the same threat model. It would take more than four times as many flips—more than 5% of the training set—for *each test point individually* to reduce our classifier to less than 50% certified accuracy.

Here we observe the same pattern, where reducing $q$ does not have a large effect

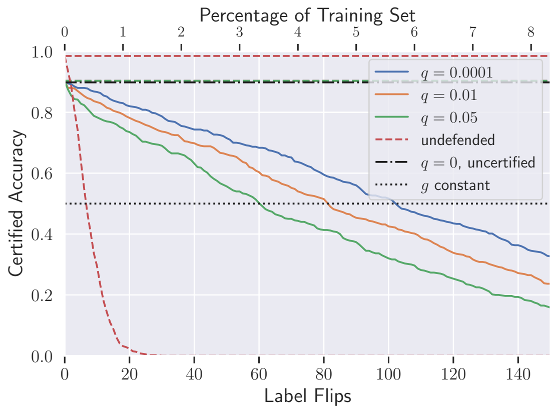


Figure 3.3: Dogfish ($n = 1800$) test set certified accuracy to adversarial label flips as $q$ is varied.

on non-robust accuracy but does increase robustness significantly. This provides further evidence for the hypothesis that more complex datasets/classifiers are more susceptible to attacks and should be smoothed with less label noise.

Figure B.3 in the Appendix shows our classifier's performance with unsupervised features. Because Dogfish is such a small dataset ($n = 1800$), deep unsupervised feature learning techniques were not feasible—we instead learned overcomplete features on 16x16 image patches using RICA [Le, 2013].

**Results on IMDB** Figure 3.4 plots the result of our randomized smoothing procedure on the IMDB review sentiment dataset. This dataset contains 25,000 training examples and 25,000 test examples, evenly split between "positive" and "negative". To extract the features we applied the Google News pre-trained Word2Vec to all the words in each review and averaged them. This feature embedding is considerably noisier than that of an image dataset, as most of the words in a review are irrelevant to sentiment classification. Indeed, Steinhardt et al. [2017] also found that the IMDB dataset was much more susceptible to adversarial corruption than images when using bag-of-words features. Consistent with this, we found smaller levels of noise resulted in larger certified accuracy. We expect significant improvements could be made with a more refined choice of feature embedding.

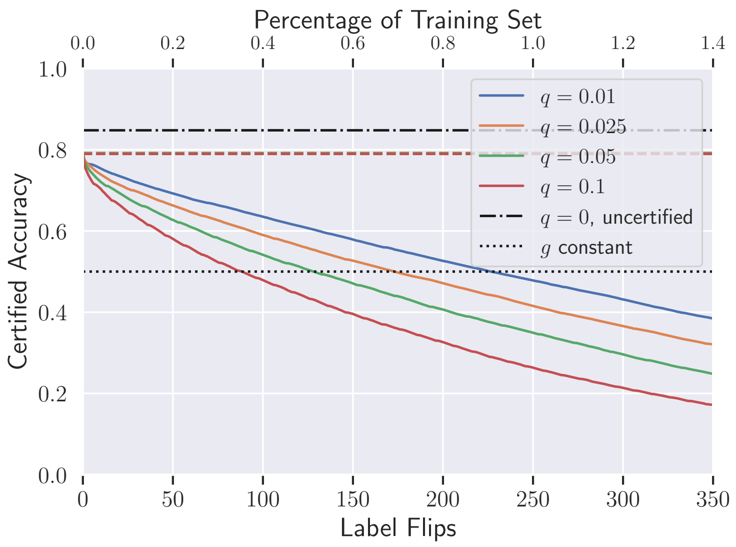


Figure 3.4: IMDB Review Sentiment ($n = 25000$) test set certified accuracy. The non-robust accuracy slightly decreases as $q$ increases; for $q = 0.01$ the non-robust accuracy is 79.11%, while for $q = 0.1$ it is 78.96%.

## 3.6 Conclusion

In this work we presented a unifying view of randomized smoothing, which borrows from the literature of differential privacy in order to provide black-box certificates of robustness. Based on the observation that this framework is applicable more broadly than just defenses to adversarial examples, we used it to derive a framework for certified defenses against arbitrary data poisoning attacks—we dub such defenses "pointwise" because they provide a certificate of robustness for each test point.

We next implemented this protocol as a specific classifier which is robust to a strong class of label-flipping attacks, where an adversary can flip labels to target each test point individually. This contrasts with previous data poisoning defenses which have typically only considered an adversary who wishes to degrade the classifier's accuracy on the test distribution as a whole. Finally, we offered a tractable algorithm for evaluating this classifier which, despite being rooted in randomization, can be computed with no Monte Carlo sampling whatsoever, resulting in a truly certifiably robust classifier. This work represents the first classification algorithm that is pointwise-certifiably robust to *any type* of data poisoning attack; we anticipate many possible new directions within this framework.

A particular strength of this framework is when we specifically care about robustly classifying each input individually. Compared to traditional robust classification, this technique is superior for determining who receives a coveted resource (a loan, parole, etc.) or for making some other sensitive classification, as it provides a guarantee for each individual. Other works only ensure that they correctly classify some fraction $p$ of the population, which is often not acceptable as that still leaves the $1 - p$ fraction who could be misclassified, with no indication of which ones belong to the test set.

There are several avenues for improvements to this line of work. Most immediately, our protocol could be implemented with other types of smoothing distributions applied to the training data, such as randomizing over the input data or features, to derive specific algorithms that are pointwise-certifiably robust to other types of data poisoning attacks. Additionally, the method for learning the input features in an unsupervised, semi-supervised, or self-supervised manner could be improved. Finally, we hope that our defense to this threat model will inspire the development of more powerful (e.g., pointwise) train-time attacks, against which future defenses can be evaluated.

# Part II

# Learning Robust Classifiers via Invariance

# Chapter 4

# The Risks of Invariant Risk Minimization

This chapter is based on [Rosenfeld et al., 2021]:
Rosenfeld, E., Ravikumar, P., & Risteski, A.
The Risks of Invariant Risk Minimization.
In *International Conference on Learning Representations*, 2021.

## 4.1 Introduction

Prediction algorithms are evaluated by their performance on unseen test data. In classical machine learning, it is common to assume that such data are drawn i.i.d. from the same distribution as the data set on which the learning algorithm was trained—in the real world, however, this is often not the case. When this discrepancy occurs, algorithms with strong in-distribution generalization guarantees, such as Empirical Risk Minimization (ERM), can fail catastrophically. In particular, while deep neural networks achieve superhuman performance on many tasks, there is evidence that they rely on statistically informative but non-causal features in the data [Beery et al., 2018, Geirhos et al., 2018, Ilyas et al., 2019]. As a result, such models are prone to errors under surprisingly minor distribution shift [Su et al., 2019, Recht et al., 2019]. To address this problem, researchers have investigated alternative objectives for training predictors which are robust to possibly egregious shifts in the test distribution.

The task of generalizing under such shifts, known as *Out-of-Distribution (OOD) Generalization*, has led to many separate threads of research. One approach is Bayesian deep learning, accounting for a classifier's uncertainty at test time [Neal, 2012]. Another technique that has shown promise is data augmentation—this includes both automated data modifications which help prevent overfitting [Shorten and Khoshgoftaar, 2019] and specific counterfactual augmentations to ensure invariance in the resulting features [Volpi et al., 2018, Kaushik et al., 2020].

A strategy which has recently gained particular traction is Invariant Causal Prediction (ICP; Peters et al. 2016), which views the task of OOD generalization through the lens of causality. This framework assumes that the data are generated according to a Structural Equation Model (SEM; Bollen 2005), which consists of a set of so-called mechanisms or structural equations that specify variables given their parents. ICP assumes moreover that the data can be partitioned into *environments*, where each environment corresponds to interventions on the SEM [Pearl, 2009] on just the *non-causal mechanisms*, and without modifying those mechanisms that are causal. The justification for this is that the causal mechanisms of the data generating process are unchanging but other effects can vary. Therefore, learning relationships that are invariant across environments ensures recovery of the invariant features which generalize under arbitrary interventions. In this work, we consider objectives that attempt to learn what we refer to as the "invariant classifier"—this is the classifier which uses and is optimal with respect to only the invariant features in the SEM. By definition, such a classifier does not overfit to environment-specific properties of the data distribution, so it will generalize even under major distribution shift at test time. In particular, we focus our analysis on one of the more popular objectives, Invariant Risk Minimization (IRM; Arjovsky et al. [2019]), but our results can easily be extended to similar recently proposed alternatives (see Appendix Appendix C.5).

Various works on invariant prediction [Muandet et al., 2013, Ghassami et al., 2017, Heinze-Deml et al., 2018, Rojas-Carulla et al., 2018, Subbaswamy et al., 2019, Christiansen et al., 2020] consider regression in both the linear and non-linear setting, but they exclusively focus on learning with partially observed covariates or instrumental variables. Under such a condition, results from causal inference [Maathuis et al., 2009, Peters et al., 2017] allow for formal guarantees of the identification of the invariant features, or at least a strict subset of them. With the rise of deep learning, more recent literature has developed objectives for learning invariant representations when the data are a non-linear function of unobserved latent factors, a common assumption when working with complex, high-dimensional data such as images. Causal discovery and inference with unobserved confounders or latents is a much harder problem [Peters et al., 2017], so while empirical results seem encouraging, these objectives are presented with few formal guarantees.

IRM is one such objective for invariant representation learning. The goal of IRM is to learn a feature embedder such that the optimal linear predictor on top of these features is the same for every environment—the idea being that only the invariant features will have an optimal predictor that is invariant. Thus, IRM hopes to "extrapolate" to new test environments, unlike ERM which excels at "interpolating". Recent works have pointed to shortcomings of IRM and have suggested modifications which they claim prevent these failures. However, these alternatives are compared in broad strokes, with little in the way of theory.

In this work, we present the first formal analysis of classification under the IRM objective under a fairly natural and general model which is similar to the original work. Our results show that despite being inspired by invariant prediction, this objective can frequently be expected to perform *no better than ERM*. In the linear setting, we present simple conditions under which solving to optimality succeeds or, more often, breaks down in recovering the invariant classifier. We additionally demonstrate a fundamental failure case—under mild conditions, there exists a feasible point that uses only non-invariant features and achieves lower empirical risk than the invariant classifier; thus it will appear as a more attractive solution, yet its reliance on non-invariant features mean it will completely fail to generalize. As corollaries, we present similar cases where Risk Extrapolation (REx; Krueger et al. 2020) and similar objectives likewise fail. Futhermore, we present the *first results in the non-linear regime*: we demonstrate the existence of a classifier with exponentially small sub-optimality which nevertheless heavily relies on non-invariant features on most test inputs, resulting in worse-than-chance performance on distributions that are sufficiently dissimilar from the training environments. These findings strongly suggest that existing approaches to ICP for high-dimensional latent variable models do not cleanly achieve their stated objective and that future work would benefit from a more formal treatment.

## 4.2 Related Work

Works on learning deep invariant representations vary considerably in their model assumptions. Some works search for a *domain-invariant* representation [Muandet et al., 2013, Ganin et al., 2016], i.e. invariance of the covariate distribution $p(\Phi(x))$, but this is typically used for domain adaptation [Ben-David et al., 2010a, Ganin and Lempitsky, 2015, Zhang et al., 2015, Long et al., 2018], with assumed access to labeled or unlabeled data from the target distribution. Further, there is evidence that this condition may not be sufficient for effective domain adaptation or OOD generalization [Zhao et al., 2019, Johansson et al., 2019]. Other works instead hope to find representations that are *conditionally* domain-invariant, expecting invariance of $p(\Phi(x) \mid y)$ [Gong et al., 2016, Li et al., 2018c].

More recent works, including the objectives discussed in this paper, suggest invariance of the *feature-conditioned label distribution*. In particular, Arjovsky et al.

[2019] only assume invariance of $\mathbb{E}[y \mid \Phi(x)]$; follow-up works rely on a stronger assumption of invariance of $p(y \mid \Phi(x))$ [Krueger et al., 2020, Xie et al., 2020, Jin et al., 2020, Mahajan et al., 2020, Bellot and van der Schaar, 2020]. Though this approach has become popular in the last year, it is somewhat similar to the existing concept of *covariate shift* [Shimodaira, 2000, Bickel et al., 2009], which considers the same setting. The main difference is that these more recent works assume that the shifts in $p(\Phi(x))$ occur between discrete, labeled environments, as opposed to more generally from train to test distributions.

## 4.3 Our Results

We consider a structural causal model with explicit separation of *invariant* features $z_c$, whose joint distribution with the label is fixed for all environments, and *environmental* features $z_e$ ("non-invariant"), whose distribution can vary across different environments. We assume that data are drawn from a set of $E$ training environments $\mathcal{E} = \{e_1, e_2, \ldots, e_E\}$ and that we know from which environment each sample is drawn. For a given environment $e$, the data are defined by the following process: first, a label $y \in \{\pm 1\}$ is drawn according to a fixed probability:

$$y = \begin{cases} 1, & \text{w.p. } \eta \\ -1, & \text{otherwise.} \end{cases} \tag{4.1}$$

Next, both invariant features and environmental features are drawn according to a Gaussian[1]:

$$z_c \sim \mathcal{N}(y \cdot \mu_c, \sigma_c^2 I), \qquad z_e \sim \mathcal{N}(y \cdot \mu_e, \sigma_e^2 I), \tag{4.2}$$

with $\mu_c \in \mathbb{R}^{d_c}$, $\mu_e \in \mathbb{R}^{d_e}$—typically, for complex, high-dimensional data we would expect $E < d_c \ll d_e$. Finally, the observation $x$ is generated as a function of the latent features:

$$x = f(z_c, z_e). \tag{4.3}$$

[1]Note the deliberate choice to have $z_e$ depend on $y$. Much work on this problem models spurious features which correlate with the label but are not causal. However, the term "spurious" is often applied incongruously; historically, a spurious correlation is one which (a) appears by chance and would disappear given enough samples or (b) is due to an unobserved confounder. In recent work, the term has been co-opted to refer to any feature that correlates with the label but does not cause it. Thus there is a subtle distinction: if we allow for *the label to cause the features* (e.g. as in natural images), the resulting correlation is *not* spurious. We therefore avoid using the term "spurious" in this work, opting instead for "non-invariant" or "environmental".

We assume $f$ is injective, so that it is in principle possible to recover the latent features from the observations, i.e. there exists a function $\Phi$ such that $\Phi(f(z_c, z_e)) = [z_c, z_e]^\top$. We write the joint and marginal distributions as $p^e(x, y, z_c, z_e)$. Where it is clear from context, we omit the specific arguments.

**Remarks on the model.** This model is natural and flexible; it generalizes several existing models used to analyze learning under the existence of adversarial distribution shift or non-invariant correlations [Schmidt et al., 2018, Sagawa et al., 2020b]. The fundamental facet of this model is the constancy of the invariant parameters $\eta, \mu_c, \sigma_c^2, f$ across environments—the dependence of $\mu_e, \sigma_e$ on the environment allows for varying distributions, while the true causal process remains unchanged. Here we make a few clarifying remarks:

- We do not impose any constraints on the model parameters—in particular, we do not assume a prior. Observe that $\mu_c, \sigma_c^2$ are the same for all environments, hence the subscript indicates the invariant (causal) relationship. In contrast, with some abuse of notation, the environmental subscript is used to indicate both dependence on the environment and the index of the environment itself (e.g., $\mu_i$ represents the mean specific to environment $i$).
- While we have framed the model as $y$ causing $z_c$, the causation can just as easily be viewed in the other direction. The log-odds of $y$ are a linear function of $z_c$—this matches logistic regression with an invariant regression vector $\beta_c = 2\mu_c/\sigma_c^2$ and bias $\beta_0 = \log \frac{\eta}{1-\eta}$. We present the model as above to emphasize that the causal relationships between $y$ and the $z_c, z_e$ are a priori indistinguishable, and because we believe this direction is more intuitive.
- The assumption of a constant marginal is necessary—in logistic regression, the optimal bias term is $\log \frac{p(y=1)}{p(y=-1)}$. Thus the optimal classifier can only be invariant if this value is the same for all environments.
- Modeling class-conditional means as direct opposites is made for clarity: it matches existing models and it greatly simplifies the calculations and results. None of our proofs require this condition, and it is straightforward to extend our results to the general case of arbitrary means.

We consider the setting where we are given infinite samples from each environment; this allows us to isolate the behavior of the objectives themselves, rather than finite-sample effects. Upon observing samples from this model, our objective is thus to learn a feature embedder $\Phi$ and regression vector $\widehat{\beta}$ to minimize the risk on an unseen environment $e$:

$$\mathcal{R}^e(\Phi, \widehat{\beta}) := \mathbb{E}_{(x,y)\sim p^e}\left[\ell(\sigma(\widehat{\beta}^\top \Phi(x)), y)\right].$$

The function $\ell$ can be any loss appropriate to classification: in this work we consider the logistic and the 0-1 loss. Note, however, that we are *not* hoping to minimize this risk in expectation over the environments; this is already accomplished via ERM or distributionally robust optimization (DRO; Bagnell 2005, Ben-Tal et al. 2009). Rather, we hope to extract and regress on invariant features while ignoring environmental features, such that our classifier generalizes to all unseen environments regardless of their parameters. In other words, the focus is on minimizing risk for the worst-case environment. We refer to the classifier which will minimize worst-case risk under arbitrary distribution shift as the *invariant classifier*. To discuss this formally, we define precisely what we mean by this term.

**Definition 4.3.1.** Assume the existence of a true set of "invariant features", whose joint or conditional distribution with the target variable is unchanging across environments. Then, the *invariant classifier* consists of: (a) a feature embedder which recovers exactly these features, and (b) the optimal predictor on top of these features.

This definition is intentionally broad; in this paper, we consider logistic regression and thus we assume the optimal predictor $\widehat{\beta}$ is a vector—note that $\Phi$ may be non-linear. Thus, the invariant classifier comprises:

$$\Phi(x) = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \circ f^{-1}(x) = [z_c], \quad \widehat{\beta} = \begin{bmatrix} \beta_c \\ \beta_0 \end{bmatrix} := \begin{bmatrix} 2\mu_c/\sigma_c^2 \\ \log \frac{\eta}{1-\eta} \end{bmatrix}.$$

This classifier is Bayes with respect to the invariant features and so it achieves the minimum possible risk without using environmental features—it is therefore also minimax optimal. Observe that *the invariant classifier is distinct from the Bayes classifier*. The Bayes classifier uses informative but non-invariant features; the invariant classifier does not.

### 4.3.1 Informal Results

With the model defined, we can informally present our results; we defer the formal statements to first give a background on the IRM objective in the next section. For the full theorem statements, see Section Section 4.5 (linear) and Section Section 4.6 (non-linear). With a slight abuse of notation, we identify a classifier by the tuple $\Phi, \widehat{\beta}$ which parametrizes it.

First, we show that the usefulness of IRM exhibits a "threshold" behavior, depending on whether $E > d_e$ or $E \leq d_e$:

**Theorem 4.3.2** (Informal, Linear)**.** *For linear $f$, consider solving the IRM objective to learn a linear $\Phi$ with invariant optimal coefficients $\widehat{\beta}$. If $E > d_e$, then $\Phi, \widehat{\beta}$ is precisely the invariant classifier; it uses only invariant features and generalizes to all environments with minimax-optimal risk. If $E \leq d_e$, then $\Phi, \widehat{\beta}$ relies upon non-invariant features.*

In fact, when $E \leq d_e$ it is even possible to learn a classifier *solely* relying on environmental features that achieves lower risk on the training environments than the invariant classifier:

**Theorem 4.3.3** (Informal, Linear)**.** *For linear $f$ and $E \leq d_e$, under mild conditions there exists a linear classifier $\Phi, \widehat{\beta}$ which uses* only environmental features*, yet achieves lower risk than the optimal invariant classifier.*

Finally, in the non-linear case, we show that IRM fails unless the training environments approximately "cover" the space of possible environments, and therefore it behaves similarly to ERM:

**Theorem 4.3.4** (Informal, Non-linear)**.** *For non-linear $f$, there exists a non-linear classifier $\Phi, \widehat{\beta}$ which is nearly optimal under the penalized objective and furthermore is nearly identical to the invariant classifier on the training distribution. However, for any test environment with a mean sufficiently different from the training means, this classifier will be equivalent to the ERM solution on nearly all test points. For test distributions where the environmental feature correlations with the label are reversed, this classifier has almost trivial performance.*

**Extensions to other objectives.** Several follow-up works have suggested alternatives to IRM (see Section 4.4). Though these objectives perform better on various baselines, there are few formal guarantees and no results beyond the linear case. Due to their collective similarities, it is straightforward to extend every theorem in this paper to these objectives, demonstrating that they all suffer from the same shortcomings. Appendix Appendix C.5 contains example corollaries for each of the main results presented in this work.

## 4.4 Background on IRM and its Alternatives

During training, a classifier will learn to leverage correlations between features and labels in the training data to make its predictions. If a correlation varies with the environment, it may not be present in future test distributions—worse yet, it may be *reversed*—harming the classifier's predictive ability. IRM [Arjovsky et al., 2019] is a recently proposed approach to learning environmentally invariant representations to facilitate invariant prediction.

**The IRM objective.** IRM posits the existence of a feature embedder $\Phi$ such that the optimal classifier on top of these features is the same for every environment. The authors argue that such a function will use only invariant features, since non-invariant features will have different conditional distributions with the label and therefore a fixed linear classifier on top of them won't be optimal in all environments.

To learn this featurizer, the IRM objective is the following constrained optimization problem:

$$\min_{\Phi,\widehat{\beta}} \quad \frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}} \mathcal{R}^e(\Phi,\widehat{\beta}) \qquad \text{s.t.} \quad \widehat{\beta} \in \arg\min_{\beta} \mathcal{R}^e(\Phi,\beta) \quad \forall e \in \mathcal{E}. \tag{4.4}$$

This bilevel program is highly non-convex and difficult to solve. To find an approximate solution, the authors consider a Langrangian form, whereby the sub-optimality with respect to the constraint is expressed as the squared norm of the gradients of each of the inner optimization problems:

$$\min_{\Phi,\widehat{\beta}} \quad \frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}} \left[\mathcal{R}^e(\Phi,\widehat{\beta}) + \lambda\|\nabla_{\widehat{\beta}}\mathcal{R}^e(\Phi,\widehat{\beta})\|^2\right]. \tag{4.5}$$

Assuming the inner optimization problem is convex, achieving feasibility is equivalent to the penalty term being equal to 0. Thus, Equations Equation (4.4) and Equation (4.5) are equivalent if we set $\lambda = \infty$.

**Alternative objectives.** IRM is motivated by the existence of a featurizer $\Phi$ such that $\mathbb{E}[y \mid \Phi(x)]$ is invariant. Follow-up works have proposed variations on this objective, based instead on the strictly stronger desideratum of the invariance of $p(y \mid \Phi(x))$. Krueger et al. [2020] suggest penalizing the variance of the risks, while Xie et al. [2020] give the same objective but taking the square root of the variance. Many papers have suggested similar alternatives [Jin et al., 2020, Mahajan et al., 2020, Bellot and van der Schaar, 2020]. These objectives are compelling—it seems natural to expect that solving them will yield the desired invariance. Indeed, it is easy to show that the invariant classifier constitutes a stationary point of each of these objectives:

**Proposition 4.4.1.** *Suppose the observed data are generated according to Equations Equation (4.1)-Equation (4.3). Then recovering the (parametrized) invariant classifier $\Phi(x) = [z_c]$ and $\widehat{\beta} = [\beta_c, \beta_0]$ is a stationary point for Equation (4.4).*

The stationarity of the invariant classifier for the other objectives is a trivial corollary. However, in the following sections we will demonstrate that such a result is misleading and that a more careful investigation is necessary.

## 4.5 The Difficulty of IRM in the Linear Regime

In their work proposing IRM, Arjovsky et al. [2019] present specific conditions for an upper bound on the number of training environments needed such that a feasible linear featurizer $\Phi$ will have an invariant optimal regression vector $\widehat{\beta}$. Our first result is similar in spirit but presents a substantially stronger (and simplified) upper bound in the classification setting, along with a matching lower bound: we demonstrate that observing a large number of environments—linear in the number of environmental features—is *necessary* for generalization in the linear regime.

**Theorem 4.5.1** (Linear case)**.** *Assume $f$ is linear. Suppose we observe $E$ training environments. Then the following hold:*

1. *Suppose $E > d_e$. Under mild non-degeneracy conditions, any linear featurizer $\Phi$ with an invariant optimal regression vector $\widehat{\beta}$ uses only invariant features, and it therefore has identical risk on all possible environments.*
2. *If $E \leq d_e$ and the environmental means $\mu_e$ are linearly independent, then there exists a linear $\Phi$ with $\text{rank}(\Phi) = d_c + d_e + 1 - E$ whose output depends on the environmental features, yet the optimal classifier on top of $\Phi$ is invariant. Further, both the logistic and 0-1 risks of this $\Phi$ and its corresponding $\widehat{\beta}$ are strictly lower than those of the invariant classifier.*

Similar to Arjovsky et al. [2019], the set of environments which do not satisfy Theorem 4.5.1 has measure zero under any absolutely continuous density over environmental parameters. Further details, and the full proof, can be found in Appendix C.3.1. Since the invariant classifier is Bayes with respect to the invariant features, by the data-processing inequality the only way a classifier can achieve lower risk is by relying on environmental features. Thus, Theorem 4.5.1 directly implies that when $E \leq d_e$, the global minimum necessarily uses these non-invariant features and therefore will not universally generalize to unseen environments. On the other hand, in the (perhaps unlikely) case that $E > d_e$, any feasible solution will generalize, and the invariant classifier has the minimum (and minimax) risk of all such classifiers:

**Corollary 4.5.2.** *For both logistic and 0-1 loss, the invariant classifier is a global minimum of the IRM objective if and only if $E > d_e$.*

Let us compare our theoretical findings to those of Arjovsky et al. [2019]. Suppose the observations $x$ lie in $\mathbb{R}^d$. Roughly, their theorem says that for a learned $\Phi$ of rank $r$ with invariant optimal coefficient $\widehat{\beta}$, if the training set contains $d - r + d/r$ "non-degenerate" environments, then $\widehat{\beta}$ will be optimal for all environments. There are several important issues with this result: first, they present no result tying the rank of $\Phi$ to their actual objective; their theory thus motivates the objective, but does not provide any performance guarantees for its solution. Next, observe when $x$ is high-dimensional (i.e. $d \gg d_e + d_c$)—in which case $\Phi$ will be low-rank (i.e. $r \leq d_e + d_c$)—their result requires $\Omega(d)$ environments, which is extreme. For example, think of images on a low-dimensional manifold embedded in very high-dimensional space. Even when $d = d_c + d_e$, the "ideal" $\Phi$ which recovers precisely $z_c$ would have rank $d_c$, and therefore their condition for invariance would require $E > d_e + d_e/d_c$, a stronger requirement than ours; this inequality also seems unlikely to hold in most real-world settings. Finally, they give no lower bound on the number of required environments—prior to this work, there were no existing results for the performance of the IRM objective when their conditions are not met. We also run a simple synthetic experiment to verify our theoretical results, drawing samples according to our model and learning a classifier with the IRM objective. Details and results of this experiment can be found in Appendix Appendix C.3.2.

We now sketch a proof of part 2 of the theorem for when $E = d_e$; this involves an explicit construction of a $\Phi$ of rank $d_c + 1$ whose optimal $\widehat{\beta}$ is invariant:

*Proof Sketch.* Since $f$ has an inverse over its range, we can define $\Phi$ as a linear function directly over the latents $[z_c, z_e]$. Specifically, we define $\Phi(x) = [z_c, p^\top z_e]$. Here, $p$ is a unit-norm vector such that $\forall e \in \mathcal{E},\ p^\top \mu_e = \sigma_e^2 \tilde{\mu}$, where $\tilde{\mu}$ is a fixed scalar that depends on the geometry of the $\mu_e, \sigma_e^2$—such a vector exists so long as the means are linearly independent. Observe that this $\Phi$ also has the desired rank. Since this is a linear function of a multivariate Gaussian, the label-conditional distribution of each environment's non-invariant latents has a simple closed form: $p^\top z_e \mid y \sim \mathcal{N}(y \cdot p^\top \mu_e, \|p\|^2 \sigma_e^2) \overset{d}{=} \mathcal{N}(y \cdot \sigma_e^2 \tilde{\mu}, \sigma_e^2)$.

For separating two Gaussians, the optimal linear classifier weights are $\Sigma^{-1}(\mu_1 - \mu_0)$—here, the optimal classifier weight on $p^\top z_e$ is precisely $2\tilde{\mu}$, which does not depend on the environment (and neither do the optimal coefficients for $z_c$). Though the distribution varies across environments, the optimal classifier is the same! Thus, $\Phi$ directly depends on the environmental features, yet the optimal regression vector $\widehat{\beta}$ for each environment is constant. To see that it has lower risk than the invariant classifier, note that this classifier is Bayes with respect to its features and that the invariant classifier uses a strict subset of these features, and therefore it has less information for its predictions. □

**A purely environmental classifier.** The precise value of $\tilde{\mu}$ in the proof sketch above represents how strongly this non-invariant feature is correlated with the label. In theory, a classifier that achieves a lower objective value could do so by a very small margin—incorporating an arbitrarily small amount of information from a non-invariant feature would suffice. This result would be less surprising, since achieving low empirical risk might still ensure that we are "close" to the invariant classifier. Our next result shows that this is not the case: there exists a feasible solution which uses *only the environmental features* yet performs better than the invariant classifier on all environments for which $\tilde{\mu}$ is large enough.

**Theorem 4.5.3.** *Suppose we observe $E \leq d_e$ environments, such that all environmental means are linearly independent. Then there exists a feasible $\Phi, \widehat{\beta}$ which uses* only environmental features *and achieves lower 0-1 risk than the invariant classifier on every environment* $e$ *such that* $\sigma_e \tilde{\mu} > \sigma_c^{-1} \|\mu_c\|$ *and* $2\sigma_e \tilde{\mu} \sigma_c^{-1} \|\mu_c\| \geq |\beta_0|$.

The latter of these two conditions is effectively trivial, requiring only a small separation of the means and balance in class labels. From the construction of $\tilde{\mu}$ in the proof of Lemma C.3.1, we can see that the former condition is more likely to be met when $E \ll d_e$ and in environments where some non-invariant features are reasonably correlated with the label—both of which can be expected to hold in the high-dimensional setting. Figure C.2 in the appendix plots the results for a few toy examples for various dimensionalities and variances to see how often this condition holds in practice. For all settings, the number of environments observed before the condition ceases to hold is quite high—on the order of $d_e - d_c$.

## 4.6 The Failure of IRM in the Non-Linear Regime

We've demonstrated that OOD generalization is quite difficult in the linear case, but it is achievable given enough training environments. Our results—and those of Arjovsky et al. [2019]—intuitively proceed by observing that each environment reduces a "degree of freedom" of the invariant solution, such that only the invariant features remain feasible if enough environments are seen. In contrast, in the non-linear case, it's not clear how to capture this idea of restricting the "degrees of freedom"—and in fact our results imply that this intuition is simply wrong. Instead, we show that the solution generalizes only to test environments that are sufficiently similar to the training environments. Thus, these objectives present no real improvement over ERM or DRO.

Non-linear transformations of the latent variables $z_c, z_e$ make it hard to directly characterize the optimal linear classifier $\widehat{\beta}$, which makes reasoning about the constrained solution to Equation (4.4) difficult. Instead, we turn our attention to Equation (4.5), the penalized IRM objective used in practice. By analyzing this objective we can reason about solutions that are sufficiently close to optimal, which allows for concrete results on the resulting classifier. In this section we demonstrate a foundational flaw of IRM and related objectives in the non-linear regime: unless we observe enough environments to "cover" the space of non-invariant features, a solution that appears to be invariant can still wildly underperform on a new test distribution. We begin with a definition about the optimality of a coefficient vector $\widehat{\beta}$:

**Definition 4.6.1.** For $0 < \gamma < 1$, a coefficient vector $\widehat{\beta}$ is $\gamma$-*close to optimal* for a label-conditional feature distribution $z \sim \mathcal{N}(y \cdot \mu, \Sigma)$ if

$$\widehat{\beta}^\top \mu \geq (1-\gamma) 2\mu^\top \Sigma^{-1} \mu.$$

Since the optimal coefficient vector is precisely $2\Sigma^{-1}\mu$, being $\gamma$-close implies that $\widehat{\beta}$ is reasonably aligned with that optimum. Observe that the definition does not account for magnitude—the set of vectors which is $\gamma$-close to optimal is therefore a halfspace which is normal to the optimal vector. One of our results in the non-linear case uses the following assumption, which says that the observed environmental means are sufficiently similar to one another.

**Assumption 4.6.2.** *There exists a $0 \leq \gamma < 1$ such that the ERM-optimal coefficients for the non-invariant features,*

$$\beta_{e;ERM} := \arg\min_{\widehat{\beta}_e} \frac{1}{|\mathcal{E}|} \sum_{e\in\mathcal{E}} \mathbb{E}_{z_c,z_e,y\sim p^e}\left[\ell(\sigma(\beta_c^\top z_c + \widehat{\beta}_e^\top z_e + \beta_0), y)\right], \tag{4.6}$$

*is $\gamma$-close to optimal for every environmental feature distribution in $\mathcal{E}$.*

This assumption says the environmental distributions are similar enough such that the optimal "average classifier" is reasonably predictive for each environment individually. This is a natural expectation: we are employing IRM precisely *because* we expect the ERM classifier to do well on the training set but fail to generalize. If the environmental parameters are sufficiently orthogonal, we might instead expect ERM to ignore the features which are not at least moderately predictive across all environments. Finally, we note that if this assumption only holds for a subset of features, our result still applies by marginalizing out the dimensions for which it does not hold.

We are now ready to give our main result in the non-linear regime. We present a simplified version, assuming that that $\sigma_e^2 = 1\ \forall e$. This is purely for clarity of presentation; the full theorem, for which the result still holds, is presented in Appendix Appendix C.4. We make use of two constants in the following proof—the average squared norm of the environmental means, $\overline{\|\mu\|^2} := \frac{1}{E}\sum_{e\in\mathcal{E}} \|\mu_e\|^2$; and the standard deviation of the response variable of the ERM-optimal classifier, $\sigma_{\text{ERM}} := \sqrt{\|\beta_c\|^2\sigma_c^2 + \|\beta_{e;\text{ERM}}\|^2\sigma_e^2}$.

**Theorem 4.6.3** (Non-linear case, simplified)**.** *Suppose we observe $E$ environments $\mathcal{E} = \{e_1, e_2, \ldots, e_E\}$, where $\sigma_e^2 = 1\ \forall e$. Then, for any $\epsilon > 1$, there exists a featurizer $\Phi_\epsilon$ which, combined with the ERM-optimal classifier $\widehat{\beta} = [\beta_c, \beta_{e;ERM}, \beta_0]^\top$, satisfies the following properties, with $p_{\epsilon,d_e} := \exp\{-d_e \min((\epsilon-1), (\epsilon-1)^2)/8\}$:*

1. *The regularization term of $\Phi_\epsilon, \widehat{\beta}$ as in Equation (4.5) is bounded as*

$$\frac{1}{E}\sum_{e\in\mathcal{E}} \|\nabla_{\widehat{\beta}} \mathcal{R}^e(\Phi_\epsilon, \widehat{\beta})\|^2 \in \mathcal{O}\left(p_{\epsilon,d_e}^2 \left(c_\epsilon d_e + \overline{\|\mu\|^2}\right)\right),$$

*for some constant $c_\epsilon$ that depends only on $\epsilon$.*

2. *$\Phi_\epsilon, \widehat{\beta}$ exactly matches the invariant classifier on at least a $1 - p_{\epsilon,d_e}$ fraction of the training set. On the remaining inputs, it matches the ERM-optimal solution.*

*Further, for any test distribution, suppose its environmental mean $\mu_{E+1}$ is sufficiently far from the training means:*

$$\forall e \in \mathcal{E}, \min_{y\in\{\pm 1\}} \|\mu_{E+1} - y \cdot \mu_e\| \geq (\sqrt{\epsilon} + \delta)\sqrt{d_e} \tag{4.7}$$

*for some $\delta > 0$, and define $q := \frac{2E}{\sqrt{\pi}\delta}\exp\{-\delta^2\}$. Then the following holds:*

3. *$\Phi_\epsilon, \widehat{\beta}$ is equivalent to the ERM-optimal classifier on at least a $1 - q$ fraction of the test distribution.*
4. *Under Assumption Assumption 4.6.2, suppose it holds that $\mu_{E+1} = -\sum_{e\in\mathcal{E}} \alpha_e\mu_e$ for some set of coefficients $\{\alpha_e\}_{e\in\mathcal{E}}$. Then so long as*

$$\sum_{e\in\mathcal{E}} \alpha_e\|\mu_e\|^2 \geq \frac{\|\mu_c\|^2/\sigma_c^2 + |\beta_0|/2 + \sigma_{ERM}}{1-\gamma}, \tag{4.8}$$

*the 0-1 risk of $\Phi_\epsilon, \widehat{\beta}$ on the new environment is greater than $.975 - q$.*

Before giving a proof sketch, we give a brief intuition for each of the claims made in this theorem:

1. The first claim says that the classifier we construct will have a gradient squared norm penalty scaling as $p^2_{\epsilon,d_e}$ which is exponentially small in $d_e$. Thus, in high dimensions, the penalty term for our classifier will shrink rapidly towards 0, meaning it will appear as a perfectly reasonable solution to the objective (Equation (4.5)).

2. The second claim says that this "fake" invariant classifier is identical to the true one on all but an exponentially small fraction of the training data; on the remaining fraction, it matches the ERM-optimal solution, which has lower risk. The correspondence between constrained and penalized optimization implies that for large enough $d_e$, the "fake" classifier will often be a preferred solution. In the finite-sample setting, we would need exponentially many samples to even distinguish between the two!

3. The third claim is the crux of the theorem; it says that this classifier we've constructed will completely fail to use invariant prediction on most environments. Recall, the intent of IRM is to perform well precisely when ERM breaks down: when the test distribution differs greatly from the training distribution. If we assume a Gaussian prior on the training environment means, they will have an $\ell_2$-norm separation in $\mathcal{O}(\sqrt{d_e})$ with high probability. Observe that $q$

will be vanishingly small so long as $\delta \geq \text{polylog}(E)$. Thus, part 3 says that IRM fails to use invariant prediction on any environment that is even slightly outside the high probability region of the training prior; even a separation of $\Omega(\sqrt{d_e \log E})$ suffices. If we instead expect the new environments to be similar, ERM already guarantees reasonable average performance at test-time; thus, IRM fundamentally *does not improve* over ERM in the non-linear regime.

4. The final statement demonstrates a particularly egregious failure case of this classifier: just like ERM, if the correlation between the non-invariant features and the label reverses at test-time, our classifier will have significantly worse than chance performance.

A back-of-the-envelope calculation for Equations Equation (4.7), Equation (4.8) and Assumption Assumption 4.6.2 makes clear just how broadly we can expect them to hold; for details, see Appendix Appendix C.4.2. With this intuition, we now present a sketch of the proof—the full proof can be found in Appendix Appendix C.4.

*Proof Sketch.* The key is a construction of $\Phi$ which is almost identical to the invariant classifier on the observed environments, yet behaves like the ERM solution at test time. We partition the environmental feature space into two sets, $\mathcal{B}, \mathcal{B}^c \subset \mathbb{R}^{d_e}$. $\mathcal{B}$ is the union of balls centered at each $\mu_e$, each with a large enough radius that it contains the majority of samples from that environment; therefore $\mathcal{B}$ represents the vast majority of the training distribution. On this set, define $\Phi(x) = [z_c]$, so our construction is equal to the invariant classifier. Now consider $\mathcal{B}^c = \mathbb{R}^{d_e} \setminus \mathcal{B}$. We use standard concentration results to upper bound the measure of $\mathcal{B}^c$ under the training distribution by $p_{\epsilon,d_e}$. Next, we show how choosing $\Phi(x) = f^{-1}(x) = [z_c, z_e]^\top$ on this set results in the gradient squared norm sub-optimality bound, which is of order $p^2_{\epsilon,d_e}$ (part 1). It is also clear that our constructed classifier is equivalent to the ERM-optimal solution on $\mathcal{B}^c$ (part 2). Thus, our classifier will have often have lower empirical risk on $\mathcal{B}^c$, counter-weighting the regularization penalty.

The second part of the proof is to demonstrate that while $\mathcal{B}$ has large measure under the training environments, it will have very small measure (upper bounded by $q$) under any moderately different test environment. We can see this by considering the separation of means (Equation (4.7)); the measure of each ball in $\mathcal{B}$ can be bounded by the measure of the halfspace containing it; if each ball is far enough away from $\mu_{E+1}$, then the total measure of these halfspaces must be small. At test time, our classifier will therefore match the ERM solution on all but $q$ of the observations (part 3). Finally, we lower bound the 0-1 risk of the ERM classifier under such a distribution shift by analyzing the distribution of the response variable. The proof is completed by observing that our classifier's risk can differ from this by at most $q$. $\square$

Theorem 4.6.3 shows that it's possible for the IRM solution to perform poorly on environments which differ even moderately from the training data. We can of course guarantee generalization if the training distributions "cover" (or approximately cover) the full space of environments in order to tie down the performance on future distributions. But in such a scenario, there would no longer be a need for ICP; we could expect ERM or DRO to perform just as well. Once more, we find that our result trivially extends to the alternative objectives; we again refer to Appendix Appendix C.5.

## 4.7 Conclusion

Out-of-distribution generalization is an important direction for future research, and Invariant Causal Prediction remains a promising approach. However, formal results for latent variable models are lacking, particularly in the non-linear setting. This paper demonstrates that Invariant Risk Minimization and subsequent works have significant under-explored risks and issues with their formulation. This raises the question: what is the correct formulation for invariant prediction when the observations are complex, non-linear functions of unobserved latent factors? It would be interesting to investigate ours or similar models further; some possible directions include (a) characterizing in which settings specifically an "invariance-like" constraint can lead to improved performance over ERM and (b) formulating an objective that encourages invariance in the non-linear setting in a way that can be formally demonstrated and quantified. We hope that this work will inspire further theoretical study on the effectiveness of IRM and similar objectives.

Chapter 5

# Iterative Feature Matching: Toward Provable Domain Generalization with Logarithmic Environments

This chapter is based on Chen et al. [2022]:
Chen, Y., Rosenfeld, E., Sellke, M., Ma, T. & Risteski, A.
Iterative Feature Matching: Toward Provable Domain Generalization with Logarithmic Environments.
In *Thirty-fifth Conference on Neural Information Processing Systems*, 2022.

## 5.1 Introduction

Domain generalization aims at performing well on unseen environments using labeled data from a limited number of training environments [Blanchard et al., 2011]. In contrast to transfer learning or domain adaptation, domain generalization assumes that neither labeled or unlabeled data from the test environments is available at training time. For example, a medical diagnostic system may have access to training datasets from only a few hospitals, but will be deployed on test cases from many other hospitals [Choudhary et al., 2020]; a traffic scene semantic segmentation system may be trained on data from some specific weather conditions, but will need to perform well under other conditions [Yue et al., 2019].

There are many algorithms for domain generalization, including Invariant Risk Minimization (IRM) [Arjovsky et al., 2019] and several variants. IRM is inspired by the principle of *invariance of causal mechanisms* [Pearl, 2009], which, under sufficiently strong assumptions, allows for provable identifiability of the features that achieve minimax domain generalization [Peters et al., 2016, Heinze-Deml et al., 2018]. However, empirical results for these algorithms are mixed; Gulrajani and Lopez-Paz [2021], Aubin et al. [2021] present experimental evidence that these methods do not consistently outperform ERM for either realistic or simple linear data models.

Recent theoretical works [Rosenfeld et al., 2021, Kamath et al., 2021] also question the theoretical foundations of IRM and its variants, shedding light on their failure conditions. These works study specific data generative models; a common assumption is that, conditioned on the label, some *invariant features* have identical distribution for all environments, and other *spurious features* have varying distributions across environments. The goal of domain generalization is then to obtain an *invariant predictor*, i.e. a classifier which uses only the invariant features. These works also often assume each training environment contains infinite samples. Thus, the central measure of domain generalization is the number of *environments* needed to recover an invariant predictor—we refer to this measure as the *environment complexity* of a learning algorithm. Rosenfeld et al. [2021] prove that even for a simple generative model and linear classifiers, the environment complexity of IRM—and other objectives based on the same principle of invariance—is at least as large as the dimension of the spurious latent features, $d_s$. Further results by Kamath et al. [2021], Ahuja et al. [2021] also point to a linear environment complexity.

Such a linear environment complexity is prohibitive for realistic applications. We usually expect there to be many more spurious dimensions than signal dimensions, whereas the number of environments observed is presumed to be much fewer. Thus, we aim to study whether it is possible to achieve an environment complexity sub-linear or even logarithmic in dimension, at least for the structured cases studied in prior theoretical works. Although the models in these works are simple, they help elucidate why existing algorithms fail and can therefore help inform better algorithmic design. Indeed, in this paper, our algorithm provably generalizes with logarithmic environment complexity.

Rosenfeld et al. [2021] showed that in their linear data model with isotropic spurious covariances, IRM has $o(d_s)$ environment complexity. We show that there is a simple algorithm that achieves an upper bound of $O(1)$ environment complexity for this baseline model. Therefore we study a more natural "smoothed covariance" extension of the data model which allows for dependence between invariant and spurious coordinates. Instead of assuming that the spurious features have isotropic covariances [Rosenfeld et al., 2021], we model their covariances as generic random positive definite matrices with adversarial biases, i.e., the covariances can be arbitrary with added noise. Under this new model, we show that

ERM and IRM still do not generalize after seeing fewer than $d_s$ environments (Theorem 5.4.2, Theorem 5.4.3). On the other hand, we propose a conceptually simple algorithm based on iterative feature matching (IFM) that is guaranteed with high probability to recover only the invariant features with environment complexity $E = O(\log d_s)$. Our method therefore *provably* achieves generalization to the worst-case test environment with a much more reasonable number of observed environments (Theorem 5.4.1).

The main idea behind IFM is to iteratively project the features to a lower dimension, in each round matching the label-conditioned feature distributions on a small, disjoint subset of the training environments. As a projection which induces invariance in the non-invariant features across one subset of environments is unlikely to do so for a different subset, we can show that with high probability, each projection removes only spurious feature dimensions. By avoiding an end-to-end training scheme, we effectively prevent the different environments from "colluding" to create a misleading solution which which depends on spurious features. As a result, IFM recovers optimal invariant predictor after $O(\log d_s)$ iterations, requiring $O(\log d_s)$ environments.

To corroborate the advantages of the proposed method, we perform experiments on a Gaussian dataset and a semi-synthetic *Noised MNIST* [LeCun et al., 1998] dataset, where the background noise spuriously correlates with the label. Our results suggest that practitioners may benefit from feature matching algorithms when the distinguishing property of the signal feature is indeed conditional distributional invariance, and may get additional advantage via matching at multiple layers with diminishing dimensions, echoing existing empirical observations [Long et al., 2015, Luo et al., 2017].

### 5.1.1 Additional Related Works

For a domain generalization problem, it is crucial to make carefully reasoned assumptions on what remains constant and what varies across environments, as different domain shift assumptions call for different algorithms. One could consider modeling the signal features $\Phi(x)$ as satisfying invariance of $P(\Phi(x) \mid y)$ ("label shift") or $P(y \mid \Phi(x))$ ("covariate shift"), among other possibilities. The model we consider allows for both of these invariances.

**Distribution matching and conditional invariance.** In the empirical literature, invariance of the conditional signal feature distribution $P(\Phi(x) \mid y)$ is the underlying assumption in widely adopted algorithms such as Correlation Alignment (CORAL) [Sun et al., 2016a, Sun and Saenko, 2016], Maximum Mean Discrepancy (MMD, Gretton et al. [2012]) [Li et al., 2018b], and (Conditional) Domain Adversarial Networks [Ganin et al., 2016, Long et al., 2018]. These algorithms are popular and enjoy empirical success in both domain adaptation and generalization, but they lack formal guarantees. Previous empirical works usually justify these algorithms using the generalization bounds based on $\mathcal{H}$-divergence [Ben-David et al., 2010a], but those bounds are generally vacuous and thus cannot explain their success. We instead study a specific data model, which is necessary for concrete guarantees on environment complexity. Prior works attempting to theoretically characterize the performance of feature matching algorithms emphasize lower bounds [Zhao et al., 2019, Tachet des Combes et al., 2020]. In contrast, our work gives the first positive theoretical justification for feature distribution matching algorithms.

The other major alternative assumption in the literature is invariance of the label distribution conditioned on the signal features. Arjovsky et al. [2019] assume invariant $\mathbb{E}[y \mid \Phi(x)]$, and follow-up works assume invariance of higher moments [Xie et al., 2020, Jin et al., 2020, Mahajan et al., 2020, Krueger et al., 2020, Bellot and van der Schaar, 2020].

**Broader theoretical study of domain generalization.** Other works analyze the task of domain generalization more generally in different settings. Blanchard et al. [2011], Muandet et al. [2013] assume a fixed prior over environments and present classification algorithms with generalization bounds that depend on properties of the prior. Considering instead convex combinations of domain likelihoods, Albuquerque et al. [2020] give a generalization bound for distributions with sufficiently small $\mathcal{H}$-divergence, while Rosenfeld et al. [2022c] model domain generalization as an online game, showing that generalizing beyond the convex hull is NP-hard.

## 5.2 Preliminaries

### 5.2.1 Domain Generalization

In domain generalization, we are given a set of $E$ training environments $\mathcal{E}_{tr}$ indexed by $e \in [E]$,[1] and a set of test environments $\mathcal{E}_{ts}$. For environment $e$ we have $n$ examples $\{(X_i^e, Y_i^e)\}_{i=1}^n$ drawn from the distribution $P_e$. In this work we study the infinite sample limit $n \to \infty$ so as to separate the effect of limited training environments from that of limited samples *per* environment, as is done in previous theoretical works [Rosenfeld et al., 2021, Kamath et al., 2021]. Let $\mathcal{X}$, $\mathcal{P}$, $\mathcal{Y}$ denote the space of inputs, intermediate features, and labels. For a featurizer $\Phi : \mathcal{X} \to \mathcal{P}$ and classifier $w : \mathcal{P} \to \mathcal{Y}$, their risk on environment $e$ is denoted

[1]We define $[n] = \{1, \dots, n\}$; $\mathbf{0}_{n\times m} \in \mathbb{R}^{n\times m}$ denotes an all-zero matrix; $\mathbb{S}^d$ is the unit sphere in $\mathbb{R}^{d+1}$; sign $(c) \in \{\pm 1, 0\}$ is the sign of scalar $c \in \mathbb{R}$. $\dagger$ denotes the Moore-Penrose pseudo-inverse.

by $R^e_{\Phi,w} = \mathbb{E}_{(X,Y)\sim P_e}[l(w \circ \Phi(X), Y)]$ for any common loss function $l$. In this paper we focus on $\mathcal{Y} = \{\pm 1\}$, linear featurizers $\Phi(X) = UX$ for $U \in \mathbb{R}^{k\times d}$, and unit-norm predictors $\hat{Y} = \text{sign}\,(w^\top Ux)$ where $w \in \mathbb{R}^k$ and $\|w^\top U\|_2 = 1$ for some feature dimension $k \leq d$ chosen by the algorithm. A predictor's 0-1 risk on environment $e$ is denoted by $R^e_{U,w} = \Pr_{(X,Y)\sim P_e}[\text{sign}\,(w^\top UX) \neq Y]$. We focus on unit-norm predictors because we evaluate on the 0-1 risk on test environments, which are invariant to the scaling of $w^\top U$ under our data model.

### 5.2.2 Baseline Algorithms

We analyze the performance of our proposed method and compare it to two baseline algorithms, ERM and IRM.

ERM learns a classifier that minimizes the average loss over all training environments, where $l$ is any common training loss such as the logistic loss:

$$\min_{w\in\mathbb{S}^{d-1}} \frac{1}{E}\sum_{e\in[E]} \mathbb{E}_{(X,Y)\sim P_e}[l(w^\top X, Y)].$$

IRM learns a featurizer $\Phi(X) \in \mathbb{R}^k$ such that the optimal classifier on top of the featurizer is invariant across training environments. As we focus on linear classifier, it is equivalent to learning a linear transformation $U \in \mathbb{R}^{k\times d}$ such that it induces a classifier $w$ that is optimal for all $e \in \mathbb{E}_{tr}$:

$$\begin{aligned} &\min_{U\in\mathbb{R}^{k\times d}, w\in\mathbb{R}^k, \|w^\top U\|_2=1} \frac{1}{E}\sum_{e\in[E]} \mathbb{E}_{(X,Y)\sim P_e} l((w^\top UX), Y) \\ \text{s.t.} \quad & w \in \arg\min_{w'\in\mathbb{R}^k} \mathbb{E}_{(X,Y)\sim P_e}[l((w'^\top UX), Y)], \forall e \in \mathcal{E}_{tr}. \end{aligned}$$

Note that this is objective is *not* the same as feature distribution matching; IRM only tries to match the first moment. Observe that this constrained objective is intended to solve a minimax domain generalization problem, as opposed to ERM which is typically viewed as minimizing the risk in expectation.

## 5.3 Problem Setup

We first recall the data model from Rosenfeld et al. [2021]. We assume without loss of generality that the label $Y$ is uniformly randomly drawn from $\{\pm 1\}$ (extension of our theorems to $Y = 1$ with probability $\eta \neq 0.5$ is straightforward). Latent variable $Z$ consists of invariant features $Z_1 \in \mathbb{R}^r$ and spurious features $Z_2 \in \mathbb{R}^{d_s}$

where $d_s = d - r$. The number of spurious features can be much larger than the number of invariant features, i.e. $d_s \gg r$. The input $X \in \mathbb{R}^d$ is generated via a linear transformation of latent variable $Z$, i.e. $X = SZ$ for an invertible matrix $S \in \mathbb{R}^{d \times d}$.

For each training environment indexed by $e \in [E]$, the features conditioned on $Y$ are drawn from a Gaussian distribution with mean $Y \cdot \mu_e \in \mathbb{R}^r$ and nonsingular covariance $\Sigma_e \in \mathbb{R}^{d \times d}$. The assumption of symmetric class center with respect to the origin can also be relaxed. Define $\mu^e = [\mu_1, \mu_2^e]$. The invariant features have constant means $\mu_1$ and covariances $\Sigma_1$ for all environments. The overall data model for training environments is summarized below:

$$\begin{aligned} Y &\sim \text{unif}\{\pm 1\} \\ Z|Y &\sim N(Y \cdot \mu_e, \Sigma_e) \in \mathbb{R}^d \\ X &= SZ. \end{aligned}$$

Since the goal of invariant feature learning is to learn a predictor that only uses the invariant features, one reasonable measure for domain generalization is a predictor's performance on test environments where the spurious features $Z_2$ are drawn from a different distribution—in particular, they are usually chosen adversarially. A classifier that predicts using the spurious features will perform badly on such test environments. When modeling the test environments, we consider the difficult scenario where there is one corresponding test environment for each training environment, whose parameters are the same except that the spurious means are flipped. Formally, for each environment $e \in \mathcal{E}_{tr}$ we construct a corresponding test environment $e' \in \mathcal{E}_{test}$ where

$$Z|Y \sim N(Y \cdot [\mu_1, -\mu_2^e], \Sigma_e).$$

Crucially, in this setting where the observations $X$ are a linear function of the latents $Z$, Rosenfeld et al. [2021] assume that the covariances of spurious features are isotropic and vary only in magnitude:

**Assumption 5.3.1** (Data model for covariances in Rosenfeld et al. [2021])**.** $\Sigma^e = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2^e \end{bmatrix}$, $\Sigma_2^e = \sigma_e^2 I_{d_s}$, *where $\sigma_e$ is a different scalar for any environment indexed by $e$.*

Rosenfeld et al. [2021] proved that the environment complexity of IRM is $o(d_s)$ in this data model. In fact, this data model is easy, as we show a simple algorithm that finds the invariant predictor using only 2 environments (Theorem 5.4.4). To study a more interesting setting with potentially higher environment complexity, we generalize the above data model in two aspects: first, the covariance of spurious features for each environment is a generic random PSD matrix, instead of only random in scaling; second, the invariant and spurious features are dependent, even though the covariance of invariant features remains constant.

**Assumption 5.3.2** (Data model for covariances in this work)**.**

$$\Sigma^e = \begin{bmatrix} \Sigma_1 & \Gamma_e^\top \\ \Gamma_e & \Sigma_2^e \end{bmatrix}, \textit{ where } \Gamma_e \in R^{d_s \times r} \textit{ differs for different } e,\ \Sigma_2^e \sim \overline{\Sigma_2^e} + G_e G_e^\top,$$

*where* $\overline{\Sigma_2^e} \in \mathbb{R}^{d_s \times d_s}$ *is arbitrary (and can be adversarial), and* $[G_e]_{i,j} \overset{iid}{\sim} N(0,1)$ *for all* $i,j \in [d_s]$*. Furthermore,*$\max_e \|\overline{\Sigma_2^e}\|_2^2 \leq D$*.*

Here $\Sigma_2^e$ can be arbitrary up to a small perturbation, which is almost always satisfied in practice. This form of assumption is common in smoothed analysis [Spielman and Teng, 2004]. In the next section, we show that baseline algorithms ERM and IRM still have $o(d_s)$ environment complexity for under Assumption 5.3.2. We propose a new algorithm (Algorithm 3) that drastically reduces the required number of training environments from $O(d_s)$ to $O(\log d_s)$. Note that the environment complexity of our algorithm only depends logarithmically on the norm bound $D$.

## 5.4 Main Results

Armed with Assumption 5.3.2, we are now ready to present our main results. We begin by presenting our algorithm based on iterative feature matching. In the following subsections, we provide formal guarantees for its environment complexity and compare it to ERM and IRM.

### 5.4.1 Iterative Feature Matching Algorithm

We hope to recover the invariant features by imposing constraints that are satisfied by them but not the spurious ones. A natural idea is to match the feature means and covariances across $\mathcal{E}_{tr}$. Since $\mu_1, \Sigma_1$ are constant, any orthonormal featurizer $U \in \mathbb{R}^{r \times d}$ such that $US$ has only non-zero entries in the first $r$ rows yields invariant means $US\mu^e$ and covariances $US\Sigma^e S^\top U^\top$. Thus we need $E$ large enough such that any $U' \in \mathbb{R}^{r \times d}$ using spurious dimensions cannot match the means and covariances. Informally, for each $e \in [E]$ we get $r \times r$ equations from matching covariances $U\Sigma^e U^\top = C$, and we have $r \times d$ parameters to estimate in $U$. Rough parameter counting suggests that if we match covariances of all $E$ environments jointly, we need at least $E > d/r$ environments to find a unique solution. Our key observation is that, due to the independence of randomness in $\Sigma_2^e$, we can split $E$ environments into disjoint groups $\mathcal{E}_1, \ldots, \mathcal{E}_T$, and use $\mathcal{E}_t$ to train an orthonormal featurizer that shrinks the feature dimensions from $r_{t-1}$ to $r_t$. Thus, in each round we shrink the dimension by a constant factor using a constant number of

**Algorithm 3** Iterative Feature Matching (IFM) algorithm

```
Require: Invariant feature dimension $r$, target feature dimensions $r_0 = d > r_1 > \cdots > r_T = r$, number of training environments $E = |\mathcal{E}_{tr}|$, infinite samples $\{(X_i^e, Y_i^e)\}_{i=1}^{\infty} \sim P_e$ from each environment $e \in \mathcal{E}_{tr}$.
  Let $\{\mathcal{E}_t\}_{t=1}^{T}$ be a partition of $\mathcal{E}_{tr}$ such that for $t < T$, $|\mathcal{E}_t| = \tilde{\Omega}((r_{t-1} - r_t)/(r_t - r - 1))$, and $|\mathcal{E}_T| = 3$.
  for $t = 1$ to $T$ do
    Find orthonormal $U_t \in \mathbb{R}^{r_t \times r_{t-1}}$ and $C_t \in \mathbb{R}^{r_t \times r_t}$ such that for all $e \in \mathcal{E}_t$ (letting $\bar{U} := U_t U_{t-1} \dots U_2 U_1$),
        $\mathbb{E}_{(X,Y)\sim P_e}[\bar{U} X X^\top \bar{U}^\top | Y] = C_t$        (5.1)
    via minimizing
        $\sum_{e,e'} \left[ Cov_e[\bar{U} X X^\top \bar{U}^\top | Y] - Cov_{e'}[\bar{U} X X^\top \bar{U}^\top | Y] \right] + \|U_t^\top U_t - I\|_F^2.$
  end for
  Return classifier $\widehat{w} = \min_{w \in \mathbb{S}^{r-1}} \frac{1}{E} \sum_{e \in [E]} \mathbb{E}_{(X,Y)\sim P_e} l(U_t \dots U_1 X, Y)$.
```

environments. The main theoretical challenge that remains is to show that in each iteration, with high probability, *only* spurious features are projected out.

This brings us to IFM (Algorithm 3), which proceeds in $T = O(\log d_s)$ rounds. Starting with an input dimension $r_0 = d$, each round we learn an orthonormal matrix $U_t$ projecting features from $r_{t-1}$ to $r_t$ dimensions so that the feature covariances after projection match across a fresh set of training environments. Although IFM requires the invariant dimension $r$ to be known, in practice it can be treated as a hyperparameter and selected via holding out some training environments for validation.

CORAL [Sun et al., 2016a] also matches feature means and covariances. However, there are several salient differences between IFM and CORAL: first, CORAL does not enforce that the featurizer is orthonormal; second, IFM learns to extract features in an unsupervised manner, whereas CORAL jointly minimizes the supervised loss and feature distribution discrepancy; third, IFM matches the feature distributions at multiple layers and uses a disjoint set of environments for each layer—this iterative process is necessary for the theoretical guarantees we provide. Despite these differences, our theoretical results serve as a justification for using feature matching algorithms in general, when the distinguishing attribute of signal vs. spurious features is that the former have invariant distributions across all environments. In Section 5.6 we empirically evaluate whether bridging the gap between IFM and CORAL may improve test accuracy.

The following theorem states that the environment complexity of IFM is logarithmic in the spurious feature dimension. A proof sketch is given in Section 5.5.

**Theorem 5.4.1** (IFM upper bound)**.** *Under Assumption 5.3.2, suppose IFM takes in* $\{r_t\}_{t=1}^T$ *as inputs, where* $r_0 = r + d_s$, $r_t - r = \lfloor (r_{t-1} - r)/2 \rfloor$ *for all* $t < T$*, and* $r_T = r$*. Suppose* $E = O(\log d_s)$*. With probability* $1 - \exp(-\Omega(d_s))$*, IFM outputs* $\widehat{w} = w^*$*.*

### 5.4.2 ERM and IRM Still Have Linear Environment Complexity

We've demonstrated that IFM has low environment complexity thanks to the additional structure assumed in our model. However, it could also be the case that this assumption allows ERM and IRM to succeed as well; perhaps they only fail because of the fixed covariance structure studied by Rosenfeld et al. [2021]. These next two results demonstrate that this is not the case: even with this structure, these algorithms are still unable to generalize to worst-case test environments.

**ERM has low test accuracy** In contrast to IFM, ERM still suffers from linear environment complexity under Assumption 5.3.2. The first theorem says there are hard instances where the ERM solution has worse-than-random performance on the test environments.

**Theorem 5.4.2** (ERM lower bound)**.** *Under Assumption 5.3.2, suppose* $E \leq d_s$*. Then any unit-norm linear classifier which achieves accuracy* $\geq \Phi\left(\frac{2\|\mu_1\|}{\sqrt{\sigma_{\min}(\Sigma^e)}}\right)$ *on all training environments will suffer 0-1 error at least* $\frac{1}{2}$ *on every test environment with flipped spurious mean, where* $\Phi$ *is the standard Normal CDF.*

A complete proof of Theorem 5.4.2 is in Appendix D.2. Note that it is quite reasonable to assume that the ERM solution satisfies the accuracy condition. In particular, it is common to model the spurious features as having much greater magnitude than the invariant features, since they have much greater dimensionality. For example, with a unit-norm mean we would expect $\|\mu_1\|^2 \approx r/d$, $\|\mu_2^e\|^2 \approx d_s/d$. Then for $r \ll d$ and $\sigma_{\min}(\Sigma^e) = \Omega(1/d)$ we have that $2\|\mu_1\|/\sqrt{\sigma_{\min}(\Sigma^e)} = O(\sqrt{r/d})$ is very close to 0, meaning the lower bound $\Phi\left(\frac{2\|\mu_1\|}{\sqrt{\sigma_{\min}(\Sigma^e)}}\right)$ is only slightly larger than $\frac{1}{2}$.

**IRM fails to learn invariant features** Our next theorem proves that even under Assumption 5.3.2, IRM is still not guaranteed to find an invariant predictor. We can show this by proving that when $E \leq d_s$, we can find a featurizer that only uses spurious dimensions, i.e., $u_s \in \mathbb{R}^{d_s}$, such that $u_s^\top \Sigma_2^e u_s = u_s^\top \mu_2^e$ for all $e \in \mathcal{E}_{tr}$. If so, the optimal predictor on top of features $u_s^\top Z_2$, $\widehat{w^e} = (u_s^\top \Sigma_2^e u_s)^{-1} u_s^\top \mu^e$ is invariant across all $e$, and can therefore be the preferred solution IRM when the spurious features have larger margin on the training environments.

**Theorem 5.4.3** (IRM lower bound)**.** *Suppose $E \leq d_s$. If $\mu_2^1, \ldots, \mu_2^E \in \mathbb{R}^{d_s}$ are linearly independent, then there exists $u_s \in \mathbb{R}^{d_s}$, $\|u_s\|_2 > 0$, such that $u_s^\top \Sigma_2^e u_s = u_s^\top \mu_2^e$ for all $e \in [E]$.*

Observe that each environment provides an ellipsoidal constraint $E_e = \{u_s \in \mathbb{R}_s^d : u_s^\top \Sigma_2^e u_s - u_2^\top \mu_2^e = 0\}$. The origin is a trivial intersection. We prove the existence of a non-trivial intersection using tools from differential topology. The key lemma is that the total number of intersection points between two manifolds of complementary dimensions $k, d-k$ is even when certain tranversality conditions hold. Using these techniques, we show that $|\bigcap_e E_e| \geq 2$ for almost all matrices $\Sigma_2^1, \ldots, \Sigma_2^E$, as long as the means are linearly independent. A complete proof of Theorem 5.4.3 is in Appendix D.3.

### 5.4.3 A Simple Algorithm Achieves O(1) Environment Complexity for the Isotropic Covariance Model in Rosenfeld et al. [2021]

We improve the upper bound of the environment complexities of the data model in Rosenfeld et al. [2021] via a new and simple Algorithm 4. Intuitively, subtracting the label-conditional covariances of any two environments yield the subspace of spurious coordinates $Q$. Once we obtain $Q$, we can transform all observations $(X_i^e, Y_i^e)$ to $(X_i^{e\prime}, Y_i^e)$ where $X_i^{e\prime} = (I - QQ^\top) X_i^e$ is the projection of $X_i^e$ onto the orthogonal subspace of $Q$. The transformed inputs have no signals in any spurious dimensions, so performing logistic regression on data $(X_i^{e\prime}, Y_i^e)_{i=1}^\infty$ from any environment $e$ yields the optimal invariant predictor $w^*$.

The following theorem provides formal guarantees for the environment complexity of Algorithm 4. The proof of is in Appendix D.4.

**Theorem 5.4.4.** *Under Assumption 5.3.1, Algorithm 4 satisfies $\widehat{w} = w^*$.*

Even though Algorithm 4 achieves $O(1)$ environment complexity when invariant and spurious features are independent, it does not work under Assumption 5.3.2, as subtracting the label-conditional covariances from two environments fails to yield the subspace of spurious features.

**Algorithm 4** A simple algorithm when invariant and spurious features are independent

```
Require: Invariant feature dimension $r$, spurious feature dimensions $d_s$, 2 training environments with infinite samples $\{(X_i^e, Y_i^e)\}_{i=1}^{\infty} \sim P_e$.
1: Taking the difference between the covariances of class 1 examples from the two environments $B = Cov_e[X|Y=1] - Cov_{e'}[X|Y=1]$.
2: Perform SVD $B = Q\Gamma Q^\top$ for orthonormal $Q \in \mathbb{R}^{d \times d_s}$ and diagonal $\Gamma \in \mathbb{R}^{d_s \times d_s}$.
3: Project the mean of class 1 examples $\mathbb{E}[X|Y=1]$ unto orthogonal subspace of $B$: $\mu' = (I - QQ^\top)\mathbb{E}[X|Y=1]$.
4: Project the covariance of class 1 examples $\Sigma' = (I - QQ^\top)Cov_e[X|Y=1](I - QQ^\top)$.
5: Return classifier $\widehat{w} = \Sigma'^{\dagger}\mu'$.
```

## 5.5 Proof Sketch for the Main Upper Bound Theorem 5.4.1

To argue that IFM outputs a featurizer $U_1 \dots U_T$ that does not use the spurious features, we need to show that the right $d_s$ columns of matrix $U_T \dots U_1 S$ are all-zero. The main lemma below says that this happens with high probability if we match $\tilde{\Omega}(1)$ [2] environments at every iteration,

**Lemma 5.5.1.** *If for all $1 \le t < T$, $|\mathcal{E}_t| = E_t = \Omega\big(\frac{r_{t-1}-r_t}{r_t-1}\max\big\{1, \log\left(\frac{D}{(r_t-1)d_s}\right), \log\left(\frac{d_s}{r_t-1}\right)\big\}\big)$, and $E_T \ge 3$, and $U_1, \dots, U_T$ are the orthonormal matrices returned by IFM, then with probability $1 - \exp\left(-\Omega(d_s)\right)$, if we write $U_T \dots U_1 S = [A, B]$, where $B \in \mathbb{R}^{r \times d_s}$, then $B = \mathbf{0}_{r \times d_s}$.*

Theorem 5.4.1 follows from Lemma 5.5.1 as follows: when we set $r_t = \lfloor (r_{t-1} + r)/2 \rfloor$ for $t < T$, we have $E_t = \tilde{\Omega}(1)$ for all $t$, and $T = O(\log d_s)$. Therefore with an environment complexity of $O(\log d_s)$, we learn a feature extractor $U = U_T \dots U_1$ that does not use any spurious dimensions. Since $U$ is orthonormal, it must contain all signal dimensions. The predictor on top of this representation uses all and only signal dimensions, so with high probability, IFM outputs $\widehat{w} = w^*$.

The first step towards proving Lemma 5.5.1 is to show that with high probability, any *one-layer* featurizer $Q_1 \in \mathbb{R}^{k_1 \times d_s}$ that uses *only* spurious dimensions cannot match feature covariances from $\tilde{\Omega}(d_s/k_1)$ environments. If a featurizer $U_1 \in \mathbb{R}^{r_1 \times d}$ uses $k_1$ spurious dimensions, there is a corresponding rank-$k_1$ featurizer $Q_1 \in \mathbb{R}^{k_1 \times d_s}$ that uses *only* spurious dimensions. So Lemma 5.5.2 implies that any $U_1$ that matches covariances in $\mathcal{E}_1$ must use at most $d_s/E_1$ spurious dimensions. We

[2] $\tilde{\Omega}(\cdot)$ hides logarithmic factors in $D, r, d_s$.

will then apply this argument recursively until we have 0 spurious dimensions.

**Lemma 5.5.2** (Informal version of Lemma D.1.2)**.** *For any integer* $2 \leq k_1 \leq d_s/2$*, when* $|\mathcal{E}_1| = E_1 = \Omega\left(\frac{d_s - k_1}{k_1 - 1} \max\left\{1, \log\left(\frac{D}{(k_1-1)d_s}\right), \log\left(\frac{d_s}{k_1-1}\right)\right\}\right)$*, with probability* $1 - O(\exp(-d_s))$*, no orthonormal* $Q \in \mathbb{R}^{k_1 \times d_s}$ *satisfies that for some constant* $C_1 \in \mathbb{R}^{k_1 \times k_1}$*.*

$$\forall e \in [E_1], \quad Q \Sigma_2^e Q^\top = C_1. \tag{5.2}$$

The formal statement Lemma D.1.2 and its proof can be found in Appendix D.1. On a high level, we discretize over the space of $Q$, and show that for fixed $Q$, denoting by $q_i$ its $i$-th row, the probability that $q_i^\top G_1 G_1^\top q_j - q_i^\top G_2 G_2^\top q_j = 0$ for all $i \neq j$ is small, so Equation (5.2) cannot be true for this fixed $Q$. Union bounding over the covering, with high probability, no $Q$ can satisfy Equation (5.2).

The next claim says that Lemma 5.5.2 can be applied iteratively, i.e. fixing a featurizer from previous iterations that uses $k_{t-1}$ spurious dimensions, with high probability, any $U_t$ that matches features from $\Omega(k_{t-1}/k_t)$ new environments uses at most $k_t$ spurious dimensions.

**Corollary 5.5.3** (Informal version of Corollary D.1.6)**.** *Suppose* $2 \leq k_t \leq k_{t-1}/2 \leq d_s/2$*. When* $|\mathcal{E}_t| = E_t = \Omega\left(\frac{k_{t-1} - k_t}{k_t - 1} \max\left\{1, \log\left(\frac{D}{(k_t-1)d_s}\right), \log\left(\frac{d_s}{k_t-1}\right)\right\}\right)$*, for fixed orthonormal* $P \in \mathbb{R}^{k_{t-1} \times d_s}$*, with probability* $1 - O(\exp(-d_s))$*, no orthonormal* $Q \in \mathbb{R}^{k_t \times k_{t-1}}$ *satisfies* $\forall e \in [E_t]$*,* $QP\Sigma_2^e P^\top Q^\top = C_t$ *for some constant* $C_t \in \mathbb{R}^{r_t \times r_t}$*.*

The formal statement Corollary D.1.6 and its proof can be found in Appendix D.1. Lemma 5.5.1 follows from iterative application of Corollary 5.5.3, as shown below.

*Proof of Lemma 5.5.1.* We shall prove that for all $t < T$, if we write $U_t \dots U_1 S = [A_t, B_t]$ where $A_t, B_t$ are the left $r$ and right $d_s$ columns of $U_t \dots U_1 S$, then $k_t = rank(B_t) = r_t - r$ with probability $1 - O(t \exp(-d_s))$. Since $T = O(d_s)$, for $t = T - 1$, the probability $1 - O(T \exp(-d_s)) = 1 - \exp(-\Omega(d_s))$.

We prove by induction on $t$. For any matrix $X$, define $P_X = X(X^\top X)^{-1} X^\top$ as the projection unto the column span of $X$. For the base case $t = 1$, since $rank(B_1) = k_1$, we have $rank((I - P_{A_1})B_1) = rank(U_1(I - P_{A_0})B_0) = k_1$. Therefore matching the covariances implies

$$\begin{aligned}
&U_1 S \begin{bmatrix} \Sigma_1 & \Gamma_e^\top \\ \Gamma_e & \Sigma_2^e \end{bmatrix} S^\top U_1^\top = C \\
&\implies (I - P_{A_1})\left(A_1 \Sigma_1 A_1^\top + A_1 \Gamma_e B_1^\top + B_1 \Gamma_e^\top A_1 + B_1 \Sigma_2^e B_1^\top\right)(I - P_{A_1}) = C' \\
&\implies (I - P_{A_1}) B_1 \Sigma_2^e B_1^\top (I - P_{A_1}) = C'' \\
&\implies Q_1 \Sigma_2^e Q_1^\top = C''',
\end{aligned}$$

where $(I - P_{A_1})B_1 = L_1 Q_1$ for some orthonormal $Q_1 \in \mathbb{R}^{k_1 \times d_s}$. This event happens with probability $O(\exp(-d_s))$ by Lemma 5.5.2.

For $t \geq 2$, suppose to the contrary that there is orthonormal $U_t \in \mathbb{R}^{r_t \times r_{t-1}}$ satisfying equation 5.1 such that $U_t \ldots U_1 S = [A_t, B_t]$ where $B_t \in \mathbb{R}^{r_t \times d_s}$, and $rank(B_t) = k_t > r_t - r$. By induction hypothesis, with probability $1 - O((t-1)\exp(-d_s))$, we can write $U_{t-1} \ldots U_1 S = [A_{t-1}, B_{t-1}]$ where $B_{t-1} \in \mathbb{R}^{r_{t-1} \times d_s}$ has rank $k_{t-1} \leq r_{t-1} - r$. Below we condition on this event.

Suppose $(I - P_{A_{t-1}})B_{t-1} = P_{t-1} Q_{t-1}$ for orthonomral $Q_{t-1} \in \mathbb{R}^{k_{t-1} \times d_s}$. Therefore we can write $(I - P_{A_t})B_t = U_t P_{t-1} Q_{t-1} = P_t Q_t Q_{t-1}$ for some orthonormal $Q_t \in \mathbb{R}^{k_t \times k_{t-1}}$.

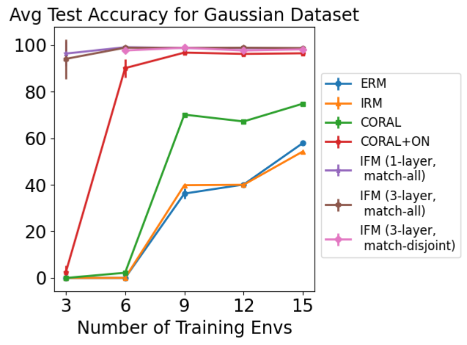
Therefore matching covariances implies $\forall e \in \mathcal{E}_t, Q_t Q_{t-1} \Sigma_2^e Q_{t-1}^\top Q_t^\top = C_t''$. Applying Corollary 5.5.3 with $P = Q_{t-1}, Q = Q_t$, with probability $1 - O(\exp(-d_s))$, no $Q_t$ satisfies $\forall e \in [E]$, $Q_t Q_{t-1} \Sigma_2^e Q_{t-1}^\top Q_t^\top = C_t$ for some constant $C_t \in \mathbb{R}^{r_t \times r_t}$.

For the last iteration $t = T$, $E_T = 3$. We assume without loss of generality $rank(B_{T-1}) = k_{T-1} \in \{1, 2\}$, since we can always half the spurious dimensions $r_t - r \leq (r_{t-1} - r)/2$ until $r_{t-1} - r = 2$.

Lemma D.1.7 and Lemma D.1.8 in Appendix D.1 deal with the cases when $k_{T-1} = 2$ and $k_{T-1} = 1$, respectively. Suppose $rank(B_{T-1}) = 2$, its associated orthonormal matrix $Q_{T-1} \in \mathbb{R}^{2 \times d_s}$. Lemma D.1.7 says that with yields that, with probability 1, no vector on the unit circle $q_T \in \mathbb{S}^1$ satisfies $q_T^\top Q_{T-1}(\Sigma_2^e - \Sigma_2^{e+1}) Q_{T-1}^\top q_T = 0$ for $e \in \{1, 2\}$. Suppose $rank(B_{T-1}) = 1$, its associated unit-norm vector $q_{T-1} \in \mathbb{R}^{d_s}$. Lemma D.1.8 says that with probability 1, no non-zero scalar $q_T$ satisfies $q_T^2 q_{T-1}^\top (\Sigma_2^1 - \Sigma_2^2) q_{T-1} = 0$. Combining the two cases, with probability $1 - O((T-1)\exp(-d_s))$, $rank(B_T) = k_T = 0$. □

## 5.6 Experiments

In light of the differences between IFM and CORAL discussed in Section 5.4.1, we test several questions inspired by our theory: (Q1) Do feature matching algorithms (IFM and CORAL) have much smaller environment complexity compared to ERM and IRM, with finite samples drawn from data models similar to our assumptions? (Q2) Can decoupling feature matching and supervised training of the classifier (IFM) improve over joint training (CORAL)? (Q3) For neural network featurizers, can matching feature distributions at multiple layers improve over matching at only the last layer (naive CORAL)? (Q4) Can matching disjoint sets of environments at each layer perform as well as matching all environments at all layers? (Q5) Is it important to shrink feature dimensions? We use two tasks to investigate those questions empirically. Appendix D.5 contains additional details.



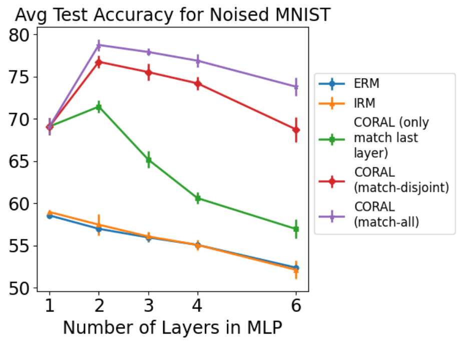


Figure 5.1: For Gaussian dataset, our algorithm IFM achieves highest test accuracy with the same number of training environments.

Figure 5.2: For Noised MNIST, matching feature distributions from multiple layers improves over naive CORAL across different architectures.

**Gaussian dataset** is a binary classification task that closely reflects our assumptions in Section 5.3. We take $r = 3$, $d_s = 32$, $\mu_1 = \mathbf{1}_r$, $\Sigma_1 = I_r$, $\mu_2^e \sim \mathcal{N}(0, 10I_{d_s})$, and $\Sigma_2^e = G_e G_e^\top$. We use $1k$ samples per environment and vary the number of training / test environments from $E = 3$ to $E = 15$.

**Noised MNIST** is a 10-way semi-synthetic classification task modified from LeCun et al. [1998] to test generalization of our theory to multi-class classification and different neural network architectures. The construction is inspired by the situation where certain background features spuriously correlate with labels ("most cows appear in grass and most camels appear in sand") [Beery et al., 2018, Arjovsky et al., 2019, Aubin et al., 2021], but the covariance of the background features changes across environments. Concretely, we divide the 60k images into $E = 12$ groups. Each group is further divided into a training and a test environment with ratio 9:1. We add an additional row of noise (28 pixels) to the original grayscale digits of dimension $28 \times 28$. In training environments, the added noise is the spurious feature that, conditioned on the label, has identical mean but changing covariances across environments. In test environments, the noise is uncorrelated with the label.

**Algorithms and architectures.** For Gaussian dataset we use linear predictors. **IRM** follows the implementation in Arjovsky et al. [2019]; **CORAL** jointly minimizes average supervised loss on training environments and $L_{coral}$, which is the average of squared distances in conditonal feature means (in $l_2$ norm) and covariances (in Frobenius norm) between adjacent training environments; **CORAL+ON**

adds orthonormal penalty loss $L_{on}(U) = \|UU^\top - I\|_F^2$ where $U$ is the featurizer; **IFM** is our Algorithm 3, where for each layer $U_t$, the training objective is $L_t(U_t) = \lambda_1 L_{coral} + \lambda_2 L_{on}$. We test IFM with 1 vs. 3-layer featurizers, either matching all (**match-all**) or a disjoint set of training environments (**match-disjoint**) at each layer.

For Noised MNIST we use ReLU networks with 1 to 6 layers. Here the unsupervised feature matching stage of IFM would fail to extract features informative of the label; nonetheless, our theory inspires us to test whether bridging the gap between IFM and CORAL can improve test accuracy. Thus, we compare naive **CORAL** which only matches feature distributions at the last layer, to variants that match at all layers post-activation, using either all (**match-all**) or a disjoint subset of training environments (**match-disjoint**) per layer.

**Results.** (Q1) Figures 5.1 and 5.2 show that IFM and CORAL have much smaller environment complexity compared to ERM and IRM in both datasets. (Q2) In Gaussian dataset, IFM improves over CORAL. (Q3) In Noised MNIST dataset, matching feature distributions at multiple layers (CORAL match-all, CORAL match-disjoint) improves over matching at only the last layer (CORAL). (Q4) In both datasets, matching disjoint sets of environments at each layer (IFM match-disjoint, CORAL match-disjoint) is almost as good as matching all environments at all layers (IFM match-all, CORAL match-all) while saving computation. (Q5) In Noised MNIST dataset (Table D.2 in Appendix D.5), shrinking feature dimensions is crucial for the advantage of feature matching at multiple layers, e.g. matching features at 3 layers with widths $[24, 24, 24]$ does not significantly improve over matching features at the last layer (CORAL). Overall, our results suggest that practitioners may benefit from feature matching algorithms when the data is similar to our assumed model, and may get additional advantage via matching at multiple layers with diminishing dimensions, echoing existing empirical works [Long et al., 2015, Luo et al., 2017].

# Chapter 6

# An Online Learning Approach to Interpolation and Extrapolation in Domain Generalization

This chapter is based on Rosenfeld et al. [2022c]:
Rosenfeld, E., Ravikumar, P., & Risteski, A.
An Online Learning Approach to Interpolation and Extrapolation in Domain Generalization.
In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, 2022.

## 6.1 Introduction

Modern machine learning algorithms excel when the training and test distributions match but often fail under even moderate distribution shift [Beery et al., 2018]; learning a predictor which generalizes to distributions which differ from the training data is therefore an important task. This objective, broadly referred to as out-of-distribution (OOD) generalization, was classically explored in a setting where there is a single "source" training distribution and a different "target" test distribution. Achieving good performance in this setting is impossible in general, so researchers have formalized several possible frameworks to study. One common choice is to make specific assumptions about covariate or label shift [Widmer and Kubat, 1996, Bickel et al., 2009, Lipton et al., 2018]; another approach is Distributionally Robust Optimization (DRO), where the test distribution is assumed to lie in some uncertainty set around the training distribution [Bagnell, 2005, Rahimian and

Mehrotra, 2019].

There has been considerable recent interest in moving beyond a single source distribution, instead assuming that the set of training data is comprised of a collection of "environments" [Blanchard et al., 2011, Muandet et al., 2013, Peters et al., 2016] or "groups" [Hu et al., 2018, Duchi et al., 2019, Sagawa et al., 2020a], each representing a distinct distribution,[1] where the group identity of each sample may be known. Such a setting is referred to as *domain generalization*. The hope is that by cleverly training on such a collection of groups, one can derive a robust predictor which will better transfer to unseen test data. Previous literature has focused exclusively on worst-case domain generalization, where the test environment is chosen to be the worst choice among a constrained set of possible test environments. It is useful to cast such a task as solving a one-shot min-max game, where the learner selects the predictor and then an adversary selects the test environment. A key specification for this game is how future test distributions depend on the training domains (i.e., the action space for the adversary).

The most immediate choice for the set of possible test environments is simply the set of training environments. More broadly, researchers have considered how to perform well when the adversary is allowed to present test distributions which "interpolate" the training distributions or "extrapolate" beyond them, but it is unclear what is the ideal formalization of such interpolations and extrapolations. A popular choice for modeling interpolation is to allow any convex combination of the training environments—this is referred to as *group/sub-population shift*, and the resulting objective is known as *Group Distributionally Robust Optimization* (DRO). Duchi et al. [2019], Sagawa et al. [2020a] give efficient algorithms for solving the Group DRO objective, but a key point is that the resulting min-max objective is *exactly equivalent* to when the adversary is limited to playing only the training environments. For modeling extrapolation, Krueger et al. [2020] consider "extrapolating" the training likelihoods (we make this formal in Section 6.2), but in this game the adversary's choice will still always be a vertex of the playable region. Thus, solving the one-shot min-max game under likelihood reweighting is always equivalent to simply minimizing worst-case risk on a discrete set.

In addition to this, formal analyses of these games are sparse. A common belief is that Empirical Risk Minimization (ERM) excels at interpolation but not extrapolation; it is also generally held as folklore that extrapolation is a much harder task, which is why generalization is so difficult—but these claims are understood intuitively, rather than mathematically. Further, Sagawa et al. [2020a] find that when using modern neural networks in the interpolation regime, explicitly solving the

[1]Throughout this work, we use the terms "domain", "distribution", and "environment" interchangeably.

Group DRO objective does not yield better solutions than simple ERM with strong regularization. Thus the relative optimality of ERM and other domain generalization algorithms remains unclear. In light of these points, we begin by considering the question: **Is there an alternative to the single-round min-max game which might allow for a more in-depth analysis of the statistical and algorithmic properties of the task of domain generalization?**

One final additional caveat with this line of research is its emphasis on worst-case optimality over all possible test environments, which is often unnecessarily conservative. This is exemplified by empirical evaluations in the OOD literature: these works train a predictor on the source data and then evaluate it on *a single test set* which is chosen adversarially with respect to the predictor. Such a protocol often misses the mark for realistically comparing the expected performance of different algorithms. For example, Gulrajani and Lopez-Paz [2021] point out that many recent works deliberately evaluate on a single train/test environment split with an unreasonably difficult distribution shift. When averaging performance over multiple environment splits, they find that no algorithm outperforms ERM. This adversarial analysis can indeed be appropriate for quantifying how an algorithm will perform in the worst possible case (particularly in safety-critical applications), but this frequently does not reflect a predictor's quality in the real world: when the test environments are *not* chosen adversarially, a reasonable learning algorithm should be able to do significantly better. Thus the crucial distinction is that **existing frameworks are minimax because they demand good performance of an algorithm even in the worst case, not because we actually expect the test environments to be chosen adversarially.**[2] This suggests there is room for a more nuanced measure of OOD generalization, one which adequately captures the purpose of such algorithms—to achieve consistently good performance on all possible test distributions—and allows for a formal comparison of their performances.

In this work, we aim to address the two main gaps identified above: formalizing the difference, if any (statistical *and* computational), between ERM and other OOD algorithms in both interpolation and extrapolation group shift settings; and doing so in a framework that allows us to analyze a predictor's performance on potentially non-adversarial (e.g., stochastic) future test environments. To do this, we take inspiration from the literature of online convex optimization [Hazan, 2016] and ask what can be achieved in a game where the learner is allowed to repeatedly refine their predictor upon observing new environments. Our analysis therefore captures an algorithm's ability to *learn and adapt* from multiple training distributions to suffer less under distribution shift and consequently perform better, on average, on future test sets. Our multi-round game generalizes existing work on domain

[2]This is a subtle point which we discuss in greater detail in Section 6.3.1.

generalization, providing new insights into the quantifiable effects of observing different environments as a function of both their number and their geometric diversity. Further, this new perspective allows for a theoretical analysis of the computational and statistical complexity of interpolation versus extrapolation, formalizing and verifying the answers to several outstanding questions which until now have only been stated intuitively.

Concretely, this work makes the following contributions:

- We recast domain generalization as a repeated online game between an adversary presenting test distributions and a player minimizing *cumulative regret*. This framework enables meaningful analysis beyond the single-round minimax setting, and we expect it can serve as a new approach to the formal study of the efficacy of robust OOD generalization algorithms.
- Under an existing notion of inter- and extrapolation, we tightly characterize their respective complexities. Specifically, we prove that i) extrapolation is indeed exponentially more difficult than interpolation in a computational sense, but ii) the statistical complexity of extrapolation is not significantly higher.
- For both inter- and extrapolation, we show that ERM—or a noisy variant—is *provably minimax-optimal* with respect to regret, as a function of the number of environments observed. For minimizing regret over any time horizon, it is impossible to improve over ERM without additional assumptions. This result supplements recent works which support the same idea theoretically [Rosenfeld et al., 2021] and empirically [Gulrajani and Lopez-Paz, 2021] for the single-round setting.

## 6.2 The Single-Round Domain Generalization Game

The key assumption of domain generalization is that the training set comprises a set of distinct domains $\mathcal{E} = \{e_i\}_{i=1}^{E}$, each of which indexes a probability distribution $p^e$, and that the test environment will relate to these domains in some pre-specified way. Let us denote the set of such possible test distributions by $\mathcal{E}_{\text{test}}$. It's common to use a minimax formulation, wherein the learner's goal is to minimize the worst-case error over the possible test distributions $\mathcal{E}_{\text{test}}$. For a set of predictors $\mathcal{F}$ and loss $\ell$, our goal is thus to solve the objective

$$\min_{f \in \mathcal{F}} \max_{e \in \mathcal{E}_{\text{test}}} \mathbb{E}_e[\ell(f)].$$

In an adversarial framework, $\mathcal{E}_{\text{test}}$ is the "playable region" of the adversary, similar to the uncertainty set in traditional DRO. A critical ingredient of the game as noted earlier is how this set of test distributions $\mathcal{E}_{\text{test}}$ depends on the training domains $\mathcal{E}$. It is typically presented as belonging to one of two distinct settings: *interpolation* and *extrapolation*. Intuitively, the interpolation setting should consist of environments which do not vary "beyond" the observed training environments, while the extrapolation setting should allow for such variation to some degree. However, these terms do not have a single agreed-upon meaning.

**Formally modeling interpolation.** Given a collection of environments, there are many possible ways to consider interpolating them. In this work, we limit our analysis to the notion of likelihood reweighting which has been used previously in several works [Duchi et al., 2019, Albuquerque et al., 2020, Sagawa et al., 2020a].[3] We model the interpolation of a set of domains as all convex combinations (i.e., mixtures) of their likelihoods. Formally, an interpolation of the domains in $\mathcal{E}$ is any distribution which is written

$$p^{\lambda} := \sum_{e \in \mathcal{E}} \lambda_e p^e, \tag{6.1}$$

where $\lambda \in \Delta_E$ is a vector of convex coefficients ($\Delta_E$ is the $(E-1)$-simplex). This is a fairly natural definition, as the space of interpolations is defined as the convex hull of the environments $\mathcal{E}$ in distribution-space. We will denote this convex hull $\text{Conv}(\mathcal{E})$.

Observe that this definition is mathematically equivalent to the set of environments which can be generated via group shift, and solving the above min-max objective is precisely Group DRO. However, this notion of single-round interpolation, while perhaps intuitive, does not actually induce a more meaningful playable region for the adversary. This is because for any predictor, the optimal choice for the adversary will be whichever training environment produces the highest risk; that is, the adversary will *always* play a vertex of the simplex. Thus, these two games are equivalent:

**Proposition 6.2.1** (Equivalence of interpolation and the discrete one-shot game)**.**

$$\min_{f \in \mathcal{F}} \max_{e \in \mathit{Conv}(\mathcal{E})} \mathbb{E}_e[\ell(f)] \;=\; \min_{f \in \mathcal{F}} \max_{e \in \mathcal{E}} \mathbb{E}_e[\ell(f)].$$

We note that in some prior work on Group DRO, learning models that minimize worst-case sub-population risk is indeed the goal—that is, they only care about test domains that match one of the source domains. In the broader domain generalization

[3]As another possibility, we could directly interpolate between two samples, but this is unlikely to be meaningful for highly complex data such as images. If we were to pose a generative model, it would instead be natural to consider interpolations of the generative parameters.

literature, however, it does not seem that this form of interpolation provides any additional constraint on OOD learning without additional regularization [Hu et al., 2018].

**Generalizing to extrapolation.** It is not immediately obvious how to extend this concept to include extrapolation. Krueger et al. [2020] suggest allowing for combinations in which the coefficients are still restricted to sum to 1, but may be slightly negative, where the minimum coefficient is given as a hyperparameter $\alpha$: $\sum_{e\in\mathcal{E}} \lambda_e = 1,\ \lambda_e \geq -\alpha\ \forall e \in \mathcal{E}$. We refer to such combinations as "bounded affine" combinations, and the objective they induce is equivalent to a fixed linear combination of the average loss plus the worst-case loss. It is immediate that the adversary's optimal choice is still on a vertex, so this game also reduces to minimizing over a discrete set:

**Proposition 6.2.2** (Equivalence of constraint set for extrapolation and the discrete one-shot game)**.**

$$\min_{f\in\mathcal{F}} \max_{e\in \mathit{Extr}_\alpha(\mathcal{E})} \mathbb{E}_e[\ell(f)] = \min_{f\in\mathcal{F}} \max_{e\in\mathcal{E}} \left[(1+E\alpha)\mathbb{E}_e[\ell(f)] - \alpha \sum_{e'\in\mathcal{E}} \mathbb{E}_{e'}[\ell(f)]\right],$$

*where $\mathit{Extr}_\alpha(\cdot)$ is all $\alpha$-bounded affine combinations.*

Thus we find that for a single round, the precise meaning of these objectives is unclear: the adversary is still choosing from a discrete set, and this model does not seem to capture the intuition that extrapolation should be fundamentally "harder" than interpolation. This shortcoming motivates our modified approach based on long-term regret, which we introduce shortly.

For extrapolating likelihoods, note that the resulting function is not guaranteed to be a probability distribution, as it could result in negative measure—one can instead frame it as reweighting of the environment *risks* (thus in Proposition 6.2.2 above, $\mathbb{E}[\cdot]$ refers to general Lebesgue integration). We study this reweighting of risks in Section 6.4.2, and we find that generalizing well over all such combinations is NP-hard. This provable difficulty in extrapolating validates our proposed sequential game, but it also indicates that additional assumptions may be necessary for modeling domain generalization. This raises interesting questions about what is the correct or most useful model of "extrapolation", which we do not address here.

## 6.3 The Sequential Domain Generalization Game

We consider recasting the task of domain generalization as a continuous game of online learning in which the player is presented with sequential test domains and

**Algorithm 5 : Domain Generalization Game** (likelihood reweighting)

```
**Input:** Convex parameter space $B$, distributions $\{p^e\}_{e\in\mathcal{E}}$ over $\mathcal{X}\times\mathcal{Y}$,
   strongly convex loss $\ell : B\times(\mathcal{X}\times\mathcal{Y})\to\mathbb{R}$, playable region $\Delta$.
**for** $t = 1\dots T$ **do**
   1. Player chooses parameters $\widehat{\beta}_t \in B$.
   2. Adversary chooses coefficients $\lambda_t \in \Delta$.
   3. Define
         $f_t(\beta) := \mathbb{E}_{(x,y)\sim p^{\lambda_t}}[\ell(\beta,(x,y))] = \sum_{e\in\mathcal{E}} \lambda_{t,e}\mathbb{E}_{(x,y)\sim p^e}[\ell(\beta,(x,y))]$.
**end for**
Player suffers regret
         $R_T = \sum_{t=1}^{T} f_t(\widehat{\beta}_t) - \min_{\beta\in B}\sum_{t=1}^{T} f_t(\beta)$.
```

must refine their predictor at each round. We're therefore interested in the player's ability to *learn continuously* and improve in each round. We would expect that any good learning algorithm will suffer less per distribution as we observe more of them—that is, the *per-round regret* should decrease over time. Specifically, we'd like to prove a rate at which our regret goes down as a function of the number of distributions we've observed. Our game allows for an analysis of the average loss (over time) of a learning algorithm across all possible test sequences—in order to bound this performance, we consider the worst such sequence. In Section 6.3.1 we expound upon this idea, comparing in detail our game to existing single-round minimax settings and discussing the benefits it affords.

We now describe the game which will allow a formal analysis of the efficacy of various domain generalization strategies. The full game can be found in the box titled Algorithm 5. Note we describe a specific instance where the adversary is limited to group mixtures as described in Section 6.2; the general game allows for any formally specified action space for the adversary and we expect this will enable future analyses involving rich classes of distribution shift threat models such as $f$-divergence or $\mathcal{H}$-divergence balls [Bagnell, 2005, Ben-David et al., 2006].

**Game Setup.** Before the game begins, we define a family of predictors parameterized by $\beta$ lying in a convex set $B$. For some observation space $\mathcal{X}$ and label space $\mathcal{Y}$, nature provides a fixed loss function $\ell : B\times(\mathcal{X}\times\mathcal{Y})\to\mathbb{R}$, strongly convex in the first argument, as well as a set of $E$ environments $\mathcal{E} = \{e_i\}_{i=1}^E$, each of which indexes a distribution $p^e$ over $\mathcal{X} \times \mathcal{Y}$. We assume that $B$ is large enough such that for any $\lambda \in \Delta_E$, the parameter which minimizes risk on $p^\lambda$ lies in $B$. We further assume that for all $\beta \in B$ and $e \in \mathcal{E}$, the expected loss of $\beta$ under $p^e$ is finite. The game proceeds as follows:

On round $t$, the player chooses parameters $\widehat{\beta}_t \in B$. Next, the adversary chooses a set of coefficients $\lambda_t := \{\lambda_{t,e}\}_{e\in\mathcal{E}}$, which defines the distribution $p^{\lambda_t}$ as the weighted combination of the likelihoods of environments in $\mathcal{E}$ with coefficients $\lambda_t$, as in Equation (6.1). For now, we assume that every choice of $\lambda$ by the adversary is a set of convex coefficients—that is, an interpolation—which ensures that $p^{\lambda_t}$ is a valid probability distribution; we will relax this restriction in Section 6.4.2. At the end of the round, the player suffers loss $f_t(\widehat{\beta}_t) = \mathcal{R}^{\lambda_t}(\widehat{\beta}_t)$, defined as the risk of the predictor parameterized by $\widehat{\beta}_t$ on the adversary's chosen distribution:

$$\mathcal{R}^{\lambda_t}(\beta) := \mathbb{E}_{(x,y)\sim p^{\lambda_t}}[\ell(\beta, (x, y))]$$

(we write $f_e = \mathcal{R}^e$ for the analogous risk on distribution $p^e$). For clarity, when using the above notation we will drop the subscript $t$ when it is not necessary.

It's important to note that in this game the player does not begin "training" until the first round; the initial environments $\mathcal{E}$ serve only to define the playable region for the adversary. Thus to recover the existing notion of single-round domain generalization, where the estimator has already seen the source environments $\mathcal{E}$ and next faces an unseen test environment, the online game would actually begin with the adversary playing each of the environment distributions in $\mathcal{E}$ once. As in standard online learning, our goal is to minimize *regret* with respect to the best fixed predictor in hindsight after $T$ rounds. That is, we hope to minimize

$$\sum_{t=1}^{T} f_t(\widehat{\beta}_t) - \min_{\beta\in B} \sum_{i=1}^{T} f_t(\beta). \tag{6.2}$$

Observe that this notion of regret straightforwardly generalizes previous work on single-round domain generalization. By allowing $T \to \infty$, we have a meaningful measure of success: each time we are presented with a new environment, we update our predictor in the hopes of improving our average performance. Crucially, this modification allows us to ask questions about the rate at which our regret decreases as a function of the number of environments observed. It also better reflects the idea that our algorithm's performance should not be evaluated in a vacuum: we aim to perform well relative to how we *could* have performed over all timesteps with a single predictor.

### 6.3.1 The Benefits of Online Regret vs. Single-Round Loss

Our focus on regret in the online setting as opposed to loss in a single round is important; it will be instructive to carefully consider the benefits to such an analysis.

**Significance of regret with respect to a fixed baseline.** The second term in Equation (6.2) is crucial; the comparison to the best *fixed* parameter prevents the adversary from forcing constant regret at each round and reflects the idea that we hope to eventually perform favorably compared to a single predictor which does reasonably well on all environments. Without this baseline, the player's objective would be to simply minimize the sum of the risks on all environments: $\sum_{t=1}^{T} f_t(\widehat{\beta}_t)$. In the adversarial setting,[4] the game therefore reduces to repeated, independent instances of the single-round version; clearly, the best we can do to minimize worst-case loss each single round is to play the minimax-optimal parameters $\beta^* := \arg\min_{\beta \in B} \max_{\lambda \in \Delta_E} \mathcal{R}^{\lambda}(\beta)$. In response, the adversary would always choose $\lambda^* := \arg\max_{\lambda \in \Delta_E} \mathcal{R}^{\lambda}(\beta^*)$. This game is uninteresting beyond the first round and does not adequately capture an algorithm's performance in a real-world setting where the environments are *not* chosen adversarially. As mentioned in the introduction, the key observation here is that the single-round minimax framework is used to guarantee good performance even in the worst-case scenario, but we do not actually expect future test environments to be chosen in this way.

As a simple example, if we were to repeatedly play $\beta^*$ and repeatedly face the test distribution $p^*$, we should consider it more likely that this is representative of future test environments (i.e., we will continue to encounter $p^*$) than that Nature is actively trying to give us the largest possible loss. Consequently we should switch strategies and play $\arg\min_{\beta \in B} \mathcal{R}^{p^*}(\beta)$, which will have better performance if the pattern continues. Thus, existing frameworks overemphasize minimax performance in individual rounds—even though in reality, distribution shift is rarely adversarial—while ignoring possible improvements over time via *adaptation to the changing environments*. In contrast, our longitudinal analysis allows for an algorithm to occasionally suffer preventable loss in any given turn, so long as the per-turn regret is guaranteed to decrease over time.

One particular setting where the benefits of this new framework are readily apparent is under gradual distribution shift. The single-round minimax formulation is intended for safety-critical applications where even a tiny mistake is fatal; however, when this is not the case, such an approach is far too conservative, and regret-based analyses provide a much clearer picture of expected performance. Our framework is thus not intended to supplant the single-round setting, but rather to supplement

[4]By this we mean the setting where the next environment is always the one which maximizes risk for the parameter chosen by the player.

it with a new, more realistic method of formal analysis of domain generalization algorithms.

**Implications of sublinear regret.** For any sequence of environments, there will be some parameter $\tilde{\beta}$ which *would have* achieved the least possible cumulative loss. Sublinear regret implies that as $T \to \infty$ we will eventually recover the per-round loss of $\tilde{\beta}$, but without committing beforehand and with *no prior knowledge* of the test environment sequence. Thus in the limit we are guaranteeing the lowest possible average loss against a fixed sequence of environments—at the same time, our analysis is minimax so as to guarantee our regret bound holds even against the worst such sequence.

Further, sublinear regret is a very powerful guarantee when the environments are stochastic, as might be expected in any real-world setting. For any prior over environment distributions $\pi(p^e)$, it is easy to see that sublinear regret implies convergence to the performance of the parameter which minimizes loss over the marginal distribution:

$$\arg\min_{\beta \in B} \int_{\mathcal{P}} \pi(p^e)\, \mathbb{E}_{p^e}[\ell(\beta, (x, y))]\; dp^e,$$

where $\mathcal{P}$ is the set of all distributions over $\mathcal{X} \times \mathcal{Y}$. This is because as $T \to \infty$, the $\pi$-weighted average of the sum of losses will converge to the loss on the marginal distribution—the baseline will then be whatever parameter minimizes this loss. Observe that this is strictly stronger than the guarantee of ERM, which ensures the same result only in the limit: sublinear regret implies that *for every* $T$, our regret with respect to the best predictor so far is bounded as $o(T)$. Thus if by chance the distributions we've seen are not representative of the prior $\pi$ (an oft-stated motivation for OOD generalization), we are still ensuring convergence to the loss of the optimal fixed predictor in hindsight, whatever it may be. In particular, if the sequence of environments is so unfavorable that the optimal predictor in hindsight is an invariant predictor [Peters et al., 2016, Arjovsky et al., 2019, Rosenfeld et al., 2021], which ignores meaningful signal to ensure broad generalization, sublinear regret guarantees that our algorithm's loss converges to this invariant predictor's loss.

We emphasize again that while the above example considers a stochastic adversary, **we do not in general assume a prior over environments**. Instead, we perform a minimax analysis to guard against the worst possible sequence of test distributions. We are measuring average regret with respect to *time*.

## 6.4 Theoretical Results

Before presenting our main theoretical results, we begin with a lemma which greatly simplifies the analysis by recharacterizing the adversary's playable region.

**Lemma 6.4.1.** *Recall $\mathcal{R}^e(\beta)$ is defined as the risk of $\beta$ on the distribution $p^e$. Then for all $\lambda \in \Delta_E$, it holds that $\mathcal{R}^\lambda(\beta) = \sum_{e\in\mathcal{E}} \lambda_e \mathcal{R}^e(\beta)$.*

This reframing allows us to generalize our analysis to extrapolation without worrying that the resulting measure is not a probability distribution. Lemma 6.4.1 implies that when the adversary chooses convex coefficients $\lambda_t$, they are equivalently choosing a loss function $f_t$ which is a combination of $\{f_e\}_{e=1}^E$, the individual environments' risks. Each choice of $\lambda_t$ uniquely defines the resulting loss function $f_t$; moving forward we will drop this explicit dependency in our notation.

### 6.4.1 Convex Combinations

Similar to Abernethy et al. [2008], we evaluate the performance of an algorithm by defining the *value* of the game after $T$ timesteps as the player's regret under optimal play by both player and adversary:

$$V_T := \min_{\widehat{\beta}_1\in B} \max_{\lambda_1\in\Delta_E} \ldots \min_{\widehat{\beta}_T\in B} \max_{\lambda_T\in\Delta_E} \left( \sum_{t=1}^T f_t(\widehat{\beta}_t) - \min_{\beta\in B} \sum_{t=1}^T f_t(\beta) \right).$$

For fixed $T$, this allows us to formalize minimax bounds on the regret. In the traditional literature, the adversary is allowed to play losses $f_t$ from a much more general class, such as all strongly convex functions. In this setting, the value of the game in any given round $t$ is known to be exactly $V_t = \sum_{s=1}^t \frac{G_s^2}{2s\sigma_{\min}}$, where $G_s$ is the Lipshitz constant of $f_s$ at the parameter chosen by the player and $\sigma_{\min}$ is the minimum curvature of $f$.[5] This means the minimax-optimal rate for regret is $\Theta(\log t)$ [Hazan et al., 2007, Bartlett et al., 2007].

In contrast to traditional online learning, where the adversary is free to choose its loss from a large non-parametric class such as all strongly convex functions, our interpolation game severely restricts the adversary, allowing only convex combinations of the risks of the $E$ distributions. We might expect that such a restriction, especially when known to the player, would allow for a faster convergence to zero regret, even if the strategy which attains it is intractable. Our first result demonstrates that this is not the case.

**Theorem 6.4.2.** *Suppose $\sigma_{\max} \geq \sigma_{\min} > 0$ such that $\forall e \in \mathcal{E},\ \sigma_{\min} I \preceq \nabla^2 f_e \preceq \sigma_{\max} I$. Define $g$ as the minimum gradient norm that is guaranteed to be forceable*

[5] We've omitted some details; see Abernethy et al. [2008] for the full result.

*by the adversary:* $g := \min_{\beta \in B} \max_{\lambda \in \Delta_E} \|\nabla f(\beta)\|_2$*. Then for all* $t \in \mathbb{N}$ *it holds that* $V_t > \frac{g^2 \sigma_{\min}}{16\sigma_{\max}^2} \log t$*.*

*Proof Sketch.* The general idea of the proof is to lower bound the regret on round $t$ by the optimal regret on round $t-1$ plus some additional loss suffered on round $t$. This loss depends on the distance from the chosen parameter on round $t$ to the regret minimizer for round $t-1$, as well as the adversary's choice on round $t$, and it can be bounded as $\Omega(1/t)$. By unrolling the recursion we derive an overall lower bound of order $\sum_{i=1}^{t} \frac{1}{i} > \log t$. The full proof can be found in Appendix E.1. $\square$

Theorem 6.4.2 provides insight into how the statistical complexity of generalizing to domain interpolations depends on the geometry of the source domains. Observe that the minimum forceable gradient norm $g$ encodes a sort of "radius" of the convex hull of loss gradients—it is easy to see that if a ball of radius $r$ can be embedded in $\text{Conv}(\{\nabla f_e(\beta)\}_{e=1}^{E})$ then $g > r$. Thus, the restriction of the adversary to the convex hull of distributions entails a restriction on the geometry of the convex hull of the corresponding loss gradients, which subsequently determines the regret our player can be forced to suffer. The bound does not directly depend on the *number* of training environments $E$; rather it scales quadratically with the size of this region, which appropriately captures the intuition that a smaller regret should be achievable for a collection of sub-distributions whose optimal parameters are very similar to one another.

With respect to the asymptotic rate of regret, this theorem provides a somewhat surprising conclusion. Even with full knowledge of the adversary's limited selection, Theorem 6.4.2 shows that no algorithm can do asymptotically better than if we were playing against the more powerful adversary playing any strongly convex function. Even more interesting, this rate can be achieved with a very simple algorithm known as Follow-The-Leader (FTL), which just plays the minimizer of the sum of all previously seen functions [Hazan et al., 2007]. In our game, this means playing the predictor which minimizes risk over all environments seen so far—after observing $t$ environments, FTL would therefore play

$$\beta_{\text{FTL}} = \arg\min_{\beta} \sum_{s=1}^{t} f_s(\beta).$$

Observe that this strategy is precisely ERM! In other words, *ERM is provably minimax-optimal for interpolation*. As the adversary's playable region is a strict subset of all strongly convex functions, it is immediate that the regret suffered by playing ERM is upper bounded as $\sum_{s=1}^{t} G_s^2 / 2s\sigma_{\min} = O(\log t)$. While Theorem 6.4.2 applies to the multi-round game, it has useful implications for the single-round setting. A simple corollary provides a tight bound on the attainable regret as a function of the number of environments seen. To our knowledge, this is the first such bound for single-round domain generalization.

**Corollary 6.4.3.** *Suppose we've seen $t$ environments. Then under the same setting as Theorem 6.4.2, the additional regret suffered due to one more round is $\Omega\left(\frac{1}{t}\right)$. This lower bound is attained by ERM.*

### 6.4.2 Bounded Affine Combinations

One could argue that allowing the adversary only convex combinations of domains is perhaps too good to hope for. Indeed, as we've seen, ERM is optimal for such a setting, but it has been widely observed that ERM fails under minor distribution shift. We might expect that future environments would fall outside of this hull—if combinations within the hull represent a formal notion of "interpolating" the training distributions, then it seems our goal instead should be to "extrapolate" beyond them.

As discussed in Section 6.2, Krueger et al. [2020] consider allowing the adversary to play bounded affine combinations of the environments; while they provide no formal results for their proposed algorithm, this conceptualization of extrapolation seems a natural extension. Clearly, this game is no easier for the player—in fact, we will demonstrate that it is *significantly* harder. For general Lipschitz functions, it is known that against the worst-case sequence, no deterministic strategy can guarantee sublinear regret, and attaining sublinear regret with a randomized strategy is NP-hard. Further, there is a regret lower bound of $\Omega(\sqrt{T})$ which was recently shown to be achievable with Follow-The-Perturbed-Leader (FTPL), assuming access to an optimization oracle for approximately minimizing a non-convex function [Suggala and Netrapalli, 2020]. As in the previous subsection, we extend these results to the task of domain generalization—that is, we demonstrate that despite the (seemingly restrictive) requirement that the adversary play bounded affine combinations of strongly convex losses that are fully known to the player, *the game remains equally hard*. These results are also surprising, as an adversary that can play arbitrary Lipschitz functions is significantly more powerful than the adversary in our game.

**Theorem 6.4.4.** *No algorithm can guarantee sublinear regret against bounded affine combinations of a finite set of strongly convex losses.*

*Proof.* We'll show that for any algorithm, there exists a sequence of loss functions chosen in response by the adversary for which the regret is bounded as $\Omega(T)$. Assume the adversary can use coefficients greater than $-\alpha$. Define

$$f_{e_1}(\beta) = \beta^2, \qquad f_{e_2}(\beta) = \beta^4 + \frac{1}{2\alpha}\beta^2.$$

On round $t$, our player will choose to play $\beta \in \mathbb{R}$. We now describe our construction of the $t$th loss in the sequence: If $|\beta| < 1$, then we choose $f_t = (1+\alpha) f_{e_1} - \alpha f_{e_2}$, and if $|\beta| \geq 1$, we choose $f_t = f_{e_1}$. In the first case, the player suffers loss $f_t(\beta) \geq 0$, and in the second case, the player suffers loss $\geq 1$. Suppose the player plays the first option $a$ times and the second option $b$ times, for a total of $a + b = T$ rounds, and suffers $\geq b$ loss.

Consider the possible best actions in hindsight. If $a \leq \frac{T}{2}$, then $\beta^* = 0$ suffers 0 loss, meaning the player's regret is at least $b = T - a \geq \frac{T}{2}$. If, on the other hand, $a > \frac{T}{2}$, then note that for any choice $\beta$ the loss suffered is

$$\begin{aligned} -a\alpha\beta^4 + (a/2 + a\alpha + b)\beta^2 &\leq a\alpha(\beta^2 - \beta^4) + (a+b)\beta^2 \\ &= \big(a\alpha(1-\beta^2) + T\big)\,\beta^2. \end{aligned}$$

Choosing $\beta^* = \sqrt{1 + \frac{3}{\alpha}}$ results in regret $\geq \frac{T}{2}$. In either case, the player suffers $\Omega(T)$ regret.

For completeness's sake, in Appendix E.2 we also include a proof of the existence of a regression task and a set of environments which could give rise to such a set of loss functions. $\square$

Thus we find that just as in the general non-convex case, a weaker adversary is necessary. In the following we consider a relaxed version with an "oblivious" adversary: this adversary is forced to select the entire sequence of loss functions at the beginning of the game (our lower bounds hold despite this relaxation). We might hope that against such a restricted adversary, the computational requirements of achieving sublinear regret would be lessened—perhaps there would be no need for an optimization oracle. However, Theorem 6.4.5 proves otherwise:

**Theorem 6.4.5.** *Against an oblivious adversary playing bounded affine combinations, achieving sublinear regret is NP-hard.*

*Proof.* Consider the problem of identifying the maximum size of a stable set of a graph on $|V|$ vertices; such a problem is not approximable in polynomial time to within a factor $|V|^{(1/2-\epsilon)}$ for any $\epsilon > 0$ unless $NP = P$ [Håstad, 1999, De Klerk, 2008]. We will demonstrate that solving this problem up to a constant factor reduces to achieving sublinear regret on an online strongly convex game with bounded affine coefficients. Let $-\alpha$ represent the minimum negative coefficient allowed for the adversary. Given the graph $G$ on $|V| > 1$ vertices, denote by $A$ its adjacency matrix. Then the maximum stable set size $\gamma(G)$ can be written $\frac{1}{\gamma(G)} = \min_{\beta \in \Delta_{|V|}} \beta^T (I + A)\beta$ by a result of Motzkin and Straus [1965]. We define a game where the adversary has two functions:

$$f_{e_1}(\beta) = \frac{1}{1+\alpha}\beta^T(|V|I + A)\beta, \qquad f_{e_2}(\beta) = \frac{|V|-1}{\alpha}\|\beta\|_2^2.$$

Note that $f_{e_1}$ is strongly convex because $(|V|-1)I + A$ is diagonally dominant and therefore PSD. Each round, the player plays some $\beta \in \Delta_{|V|}$, and the (oblivious) adversary chooses the loss

$$(1+\alpha)f_{e_1} - \alpha f_{e_2} = \beta^T(|V|I + A)\beta - (|V|-1)\|\beta\|_2^2 = \beta^T(I+A)\beta.$$

Define $L_T$ as the loss suffered by the player after $T$ rounds. Clearly, the optimal choice would be to play $\beta$ such that $\beta^T(I+A)\beta = \frac{1}{\gamma(G)}$ each round, implying that $\frac{T}{\gamma(G)} \leq L_T$ and also that regret can be written $L_T - \frac{T}{\gamma(G)}$. Suppose there exists a polynomial-time strategy with regret growing sublinearly with $T$. Then by definition, there exists a constant $T_0 \in \text{poly}(|V|)$ such that on all rounds $T > T_0$, the player's regret is upper bounded as

$$L_T - \frac{T}{\gamma(G)} \leq \frac{1}{|V|}T \leq \frac{T}{\gamma(G)} \implies L_T \leq \frac{2T}{\gamma(G)}.$$

Putting these inequalities together, we get $\frac{1}{\gamma(G)} \leq \frac{L_T}{T} \leq \frac{2}{\gamma(G)}$, which implies $\frac{1}{2}\gamma(G) \leq \frac{T}{L_T} \leq \gamma(G)$. Recall that this holds for all $T > T_0$, so our polynomial-time algorithm has attained a 2-approximation to the maximum stable set size. $\square$

Computationally, our game of extrapolation is just as difficult as achieving sublinear regret on arbitrary Lipschitz functions. These results present, for the first time, proof of *an exponential computational complexity gap between interpolation and extrapolation in the domain generalization setting*, formally verifying existing intuition.

We now turn our attention to the statistical complexity of regret minimization under bounded affine combinations. Recall that for the case of convex combinations (i.e. interpolations), Theorem 6.4.2 shows a minimax lower bound of $\Omega(\log t)$ which can be achieved with standard ERM. Before we consider the bounded affine setting (i.e. extrapolations), we again note that for an adversary playing arbitrary Lipschitz functions, Suggala and Netrapalli [2020] demonstrate that with access to a non-convex optimization oracle, FTPL can achieve the minimax lower bound of $\Omega(\sqrt{T})$. The FTPL strategy is to play the parameter which minimizes the sum of the observed environments plus a noise term—specifically, FTPL takes the sum of existing risks, samples a random linear function of the parameters, and solves for the parameters which minimize this "perturbed" sum. In our game, then, FTPL is just a noisy variant of ERM. Computational limitations notwithstanding, the natural next question is if playing against an oblivious adversary is enough of a relaxation that we can surpass this lower bound. That is, can we outperform ERM in this setting *at all?* Our final result answers this question in the negative:

**Theorem 6.4.6.** *Against an oblivious adversary playing bounded affine combinations, the achievable regret is lower bounded as $\Omega(\sqrt{T})$.*

*Proof.* For a fixed, convex loss $\ell$ and convex parameter space $\Theta$, predicting with expert advice is known to have an information-theoretic minimax regret lower bound of $\Omega(\sqrt{T})$ [Cesa-Bianchi and Lugosi, 2006, Theorem 3.7]. We will give a reduction which demonstrates that the same lower bound holds for bounded affine combinations of strongly convex losses.

Assume a fixed convex loss $\ell : \Theta \times \Theta \mapsto \mathbb{R}$ over convex $\Theta$ and fix the adversary's coefficient lower bound as $-\alpha$. Suppose on round $t$, we are presented with $E$ experts' predictions, which we imagine as an $E$-dimensional vector $\tilde{\theta}_t$ whose $i$th entry is the prediction of the $i$th expert. Define the following functions over elements $\delta \in \Delta_E$:

$$f_{e_1}(\delta, \theta^*) = \frac{1}{1+\alpha}\left[\ell(\delta^T\tilde{\theta}_t, \theta^*) + \|\delta\|_2^2\right], \qquad f_{e_2}(\delta, \theta^*) = \frac{1}{\alpha}\|\delta\|_2^2.$$

Note that both these functions are both strongly convex in $\delta$. Consider what happens if the adversary plays $(1+\alpha)f_{e_1} - \alpha f_{e_2} = \ell$. Suppose for the sake of contradiction there exists an algorithm playing $\widehat{\delta}_t$ which achieves $o(\sqrt{T})$ regret with respect to $\delta^*$, defined as the best fixed $\delta \in \Delta_E$ in hindsight:

$$\delta^* := \arg\min_{\delta\in\Delta_E} \sum_{t=1}^{T} \ell(\delta^T\tilde{\theta}_t, \theta^*).$$

As this represents a convex combination of the experts' predictions, it is clear that the loss suffered by $\delta^*$ will be less than or equal to the loss suffered by the best expert. This implies that by taking this algorithm's choice $\widehat{\delta}_t$ each round and playing $\widehat{\delta}_t^T\tilde{\theta}_t$, we will achieve $o(\sqrt{T})$ regret with respect to the best expert, defying the known lower bound. It follows that the lower bound of $\Omega(\sqrt{T})$ holds even for bounded affine combinations of strongly convex functions. □

This theorem implies two crucial points: firstly, that *ERM remains minimax optimal for this model of extrapolation*; and secondly, that *proper regularization is essential for good OOD generalization*. This provides theoretical justification for the empirical findings of Sagawa et al. [2020a] and complements existing results on the value of explicit regularization for group shift [Hu et al., 2018]. Additionally, we find that even though there is an exponential computational complexity gap between the two tasks, the statistical gap is not too large—$\Theta(\log T)$ versus $\Theta(\sqrt{T})$ regret.

## 6.5 Related Work

Many works provide formal guarantees for OOD generalization by assuming invariances in the causal structure of the data: a set of interventions is assumed to result in separate fixed environments [Peters et al., 2016, Heinze-Deml et al., 2018, Heinze-Deml and Meinshausen, 2020, Christiansen et al., 2020] or distribution shift over time [Tian and Pearl, 2001, Didelez et al., 2006], and the test distribution will likewise represent such an intervention. Under sufficiently strong conditions it is then possible to identify which features have invariant relationships with the target variable; recovery of these features ensures reasonable performance despite arbitrary future interventions on the other variables. However, these works assume full or partial observation of the covariates, and therefore they do not apply to the setting where the data is a complex function of unobserved latent variables.

Works which eschew a direct causal formalization often still depend upon the intuition of "invariance" within the context of causality. The IRM objective [Arjovsky et al., 2019] was designed for such a setting assuming the target variables' causal mechanisms remain invariant, but it lacked serious theoretical justification; Krueger et al. [2020] likewise suggest an algorithm for extrapolation but similarly fail to provide any formal guarantees. Rosenfeld et al. [2021] subsequently showed that, while these and other similar objectives may work under strong conditions in the linear setting, the same cannot be said for more complex data. Albuquerque et al. [2020] theoretically analyze extrapolation beyond the convex hull of domain likelihoods and give generalization bound via $\mathcal{H}$-divergences. Unfortunately, this bound scales linearly with both the maximum discrepancy between pairs of training distributions and between the test distribution and training environment hull.

This work relates the nascent study of domain generalization theory to prior work on online and lifelong learning [Thrun, 1998, Mitchell et al., 2015, Hazan, 2016], for which there already exist provable regret bounds and efficiency guarantees [Balcan et al., 2015, Alquier et al., 2017]. The main difference is that those works—which are for more general online learning—present new algorithms and give upper bounds, while this work focuses on OOD generalization and proves lower bounds which match rates already known to be achievable for more general classes of losses [Hazan et al., 2007, Abernethy et al., 2008, Suggala and Netrapalli, 2020] , implying that existing algorithms (ERM and a noisy variant) are already optimal.

## 6.6 Conclusion and Future Directions

This work presents the first formal results demonstrating an exponential computational gap between interpolation and extrapolation in domain generalization, a claim which has until now only been given vague intuitive justification. Perhaps more importantly, we've shown that ERM remains statistically minimax-optimal for both tasks—given the observed failure of ERM in practice, this suggests that there is quite a bit more subtlety to distribution shift in the real world. Taken together, our results present strong evidence that the "likelihood reweighting" model of distribution shift, while perhaps appropriate for specific settings involving subpopulations, might not be appropriate for the more general study of extrapolation to new domains. It could instead be beneficial to reconsider existing notions of inter- and extrapolation—particularly those involving linearity or generic likelihood reweighting—in the context of online learning, where the notions of regret and stochastic adversaries allow for more a nuanced study of statistical and algorithmic complexity.

We see two important directions for further research. First, the proposed domain generalization game serves as a standalone framework for the theoretical analysis of learning algorithms. As discussed in Section 6.3.1, considering regret in the online setting provides a more nuanced signal of an algorithm's expected performance, especially when we are not too worried about the *literal worst case test distribution*. We hope that this new perspective will better enable future work to provide formal OOD generalization guarantees for their proposed methods. We note that this work considers only strongly convex functions, but using the same techniques one could extend the analysis to more general classes such as all convex losses; this setting might eliminate the statistical complexity gap and could lead to additional insight into the differences between inter- and extrapolation.

Second, there still remains significant flexibility in how we define "interpolation" and "extrapolation" with respect to training environments; we consider one specific notion in this work, and we show that ERM remains optimal—implying that alternative formulations may be preferable. However, it seems likely that different restrictions on the adversary could allow for stronger generalization guarantees. Furthermore, our analysis reveals that the *geometry of the environmental loss functions* is a critical element for generalization. This suggests additional improvements can be achieved with careful representation learning.

# Part III

# Understanding Heavy-Tailed Data, and How to Handle it

Chapter 7

# Domain-Adjusted Regression or: ERM May Already Learn Features Sufficient for Out-of-Distribution Generalization

This chapter is based on Rosenfeld et al. [2022b]:
Rosenfeld, E., Ravikumar, P., & Risteski, A.
Domain-Adjusted Regression or: ERM May Already Learn Features Sufficient for Out-of-Distribution Generalization.
In *NeurIPS 2022 Workshop on Distribution Shifts (DistShift)*.

## 7.1 Introduction

The historical motivation for deep learning focuses on the ability of deep neural networks to automatically learn rich, hierarchical features of complex data [LeCun et al., 2015, Goodfellow et al., 2016]. Simple Empirical Risk Minimization (ERM), with appropriate regularization, results in high-quality representations which surpass carefully hand-selected features on a wide variety of downstream tasks. Despite these successes, or perhaps because of them, the dominant focus of late is on the shortcomings of this approach: recent work points to the failure of networks trained with ERM to generalize under even moderate distribution shift [Recht et al., 2019, Miller et al., 2020a]. A common explanation for this phenomenon is reliance on "spurious correlations" or "shortcuts", where a network makes predictions based on structure in the data which generalizes on average in the training set but may not persist in future test distributions [Poliak et al., 2018, Kaushik and Lipton, 2018, Geirhos et al., 2019, Xiao et al., 2021].

Many proposed solutions *implicitly assume* that this problem is due to the entire neural network: they suggest an alternate objective to be minimized over a deep network in an end-to-end fashion [Sun et al., 2016a, Ganin et al., 2016, Arjovsky et al., 2019]. These objectives are complex, poorly understood, and difficult to optimize. Indeed, the efficacy of many such objectives was recently called into serious question [Zhao et al., 2019, Rosenfeld et al., 2021, Gulrajani and Lopez-Paz, 2021]. Though a neural network is often viewed as a deep feature embedder with a final linear predictor applied to the features, it is still unclear—and to our knowledge *has not been directly asked or tested*—whether these issues are primarily because of (i) learning the wrong features or (ii) learning good features but failing to find the best-generalizing linear predictor on top of them.

We begin with a simple experiment (Figure 7.1) to try to distinguish between these two possibilities: we train a deep network with ERM on several domain generalization benchmarks, where the task is to learn a predictor using a collection of distinct training domains and then perform well on a new, unseen domain. After training, we freeze the features and separately learn a linear classifier on top of them. Crucially, when training this classifier (i.e., retraining just the last linear layer), we give it an unreasonable advantage by optimizing on both the train and test domains—henceforth we refer to this as "cheating". Since we use just a linear classifier, this process establishes a lower bound on what performance we could plausibly achieve using standard ERM features, with access to data from the target distribution. We then separately cheat while training the full network end-to-end, simulating the idealized setting with no distribution shift. Note that in neither case do we train on the test *points*; our cheating entails training on (different) samples from the test *domain*, which are assumed unavailable in domain generalization.

Notably, we find that simple (cheating) logistic regression on frozen deep features learned via ERM results in enormous improvements over current state of the art, on the order of 10-15%. In fact, it usually performs comparably to the full cheating method—which learns both features and classifier end-to-end with test domain access—sometimes even outperforming it. Put another way, cheating while training the entire network rarely does significantly better than cheating while training just the last linear layer. One possible explanation for this is that the pretrained model is so overparametrized as to effectively be a kernel with universal approximation power (such as the Neural Tangent Kernel [Jacot et al., 2018, Du et al., 2019]); in this case, the outstanding performance of a cheating linear classifier

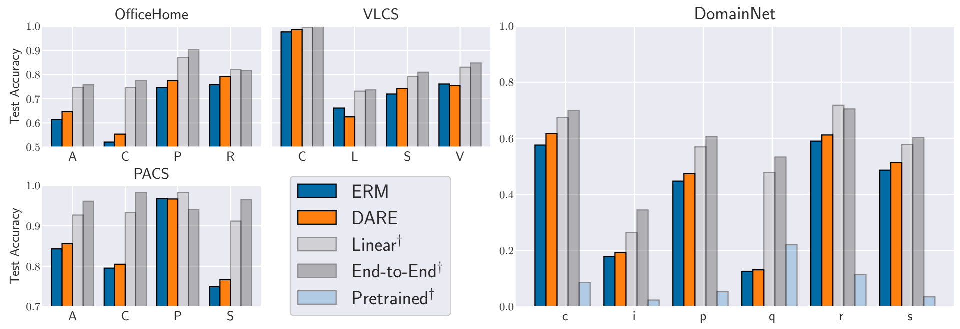


Figure 7.1: Accuracy via "cheating": (mean over 3 trials) dagger (†) denotes access to test domain at train-time. Each letter is a domain. Dark blue is approximate SOTA, orange is our proposed DARE objective, light grey represents cheating while retraining the linear classifier only. All three methods use the *same features*, attained without cheating. Dark grey is "ideal" accuracy, cheating while training the entire deep network. Surprisingly, cheating only for the linear classifier rivals cheating for the whole network. Cheating accuracy on pretrained features (light blue) makes clear that this effect is due to finetuning on the train domains, and not simply overparameterization (i.e., a very large number of features).

on top of these features would be unsurprising. However, we find that this cheating method does *not* ensure good performance on pretrained features, which implies that we are not yet in such a regime and that the effect we observe is indeed due to finetuning via ERM. Collectively, these results suggest that training modern deep architectures with established training and regularization practices may be "good enough" for learning features which generalize out-of-distribution and that the current bottleneck lies primarily in learning a simple, robust predictor.

Motivated by these findings, we propose a new objective, which we call Domain-Adjusted Regression (DARE). The DARE objective is convex and it learns a linear predictor on frozen features. Unlike invariant prediction [Peters et al., 2016], which projects out feature variation such that a single predictor performs acceptably on very different domains, DARE performs a domain-specific adjustment to unify the environmental features in a canonical latent space. Based on the presumption that standard ERM features are good enough (made formal in Section 7.4), DARE enjoys strong theoretical guarantees: under a new model of distribution shift which captures ideas from invariant/non-invariant latent variable models, we precisely characterize the adversarial risk of the DARE solution against a natural perturbation set, and we prove that this risk is *minimax*. The perturbation set consists of all test distributions where the adversary can introduce any change in directions in which the training domains already vary but is allowed only limited variation in new directions. We further provide the first *finite-environment* convergence guarantee to the minimax risk, improving over existing results which merely demonstrate a threshold in the number of observed environments at which the solution is discovered [Rosenfeld et al., 2021, Chen et al., 2022, Wang et al., 2022a]. Finally, we show how our objective can be modified to leverage access to unlabeled samples at test-time. We use this to derive a method for provably effective "just-in-time" unsupervised domain adaptation, for which we provide a finite-sample excess risk bound.

Evaluated on finetuned features, we find that DARE compares favorably to existing methods, consistently achieving equal or better performance. We also find that methods which previously underperformed on these benchmarks do much better in this frozen feature setting, often besting ERM. This suggests that these approaches *are* beneficial for linear prediction, and that when they do work, it is primarily due to learning a better linear classifier. We hope these results will encourage going "back to basics", with future work focusing on simpler, easier to understand methods for robust prediction.

## 7.2 ERM Learns Surprisingly Useful Features

Our experiments are motivated by a simple question: *is the observed failure of deep neural networks to generalize out-of-distribution more appropriately attributed to inadequate* feature learning *or inadequate* robust prediction*?* Both could undoubtedly be improved, but we are concerned with which currently serves as the primary bottleneck. It's typical to train the entire network end-to-end and then evaluate on a test distribution; in reality, this measures the quality of the interaction of the features and the classifier, not of either one individually.

Using datasets and methodology from DomainBed [Gulrajani and Lopez-Paz, 2021], we finetune a ResNet-50 [He et al., 2016] with ERM on the training domains to extract features. Next, we cheat while learning a linear classifier on top of these frozen features by optimizing on both the train and test domains. We compare this cheating linear classifier to a full cheating network trained on all domains end-to-end. If it is the case that ERM learns features which do not generalize, we should expect that the cheating linear classifier will not substantially improve over the current state of the art and will perform significantly worse than cheating end-to-end, since the latter method can adapt the features to better suit the test domain.

Instead, we find that simply giving the linear predictor access to all domains while training makes up the vast majority of the gap between current state of the art (ERM with heavy regularization) and the ideal setting where we train a network on all domains. In other words, ERM produces features which are informative enough that a linear classifier on top of these frozen features is—in principle—capable of generalizing *almost as well as if we had access to the test domain when training the entire network.*[1] These features can also substantially outperform current state of the art given a small amount of labeled test data for retraining the last layer. Figure F.1 in the Appendix depicts the evaluation methodology described above, along with a more detailed explanation.

We conjecture that this phenomenon occurs more broadly: our findings suggest that for many applications, existing features learned via ERM may be "good enough" for out-of-distribution generalization in deep learning. That is, ERM may already "undo" a large component of the non-linearities we observe in complex data, even for unseen distributions. This also implies that in settings where we have access to a small amount of data from the target distribution, simply retraining the last linear layer on this data will suffice for excellent performance.

Two immediate questions are in which other settings this holds and if, instead of using more elaborate approaches such as complex regularization, the remaining gap could be closed using simpler methods. Based on this idea, we posit that future work would benefit from modularity, working to improve representation learning and robust classification/regression separately.[2] There are several distinct advantages to this approach:

- **More robust comparisons and better reproducibility:** Current methods have myriad degrees of freedom which makes informative comparisons difficult; evaluating feature learning and robust regression separately eliminates many of these sources of experimental variation.

- **Less compute overhead:** Training large networks to learn both features and a classifier is expensive. Benchmarks could include files with the weights for ready-to-use deep feature embedders trained with various objectives—these models can be much larger and trained on much more data than would be feasible for many. Using these features, academic researchers could thus make faster progress on better classifiers with less compute.

- **Modular theoretical guarantees:** Conditioning on the frozen features, we can use more classical analyses to provide guarantees for the simpler parametric classifiers learned on top of these features. For example, Bansal et al.

[1]On a few domains the linear method sees a gap of $\sim$5% accuracy from the idealized setting. We emphasize that our use of a simple linear predictor serves as a *lower bound* on the achievable error using ERM features. The end-to-end result represents ideal performance that would a priori seem totally out of reach with current methods.

[2]We are not suggesting an abandonment of the end-to-end framework; we believe both methods have merit. For example, the fact that the representation suffices does not imply that it corresponds to meaningful semantic features, and encouraging such a correspondence may depend on learning a system end-to-end.

[2021] derive a generalization bound for simpler classifiers which is agnostic to the complexity of the features on which they are trained.

We conclude by emphasizing that while the predictor in our experiments is linear, future methods need not be. Rather, we are highlighting that **there may be no need for complex, expensive, highly variable regularization of a deep network when much simpler approaches suffice.**

## 7.3 The Domain-Adjusted Regression Objective

The goal of many prior methods in deep domain adaptation or generalization is to learn a single network which does well on all environments simultaneously, often by throwing away non-invariant components [Peng et al., 2019a, Arjovsky et al., 2019]. While invariance is a powerful framework, there are some clear drawbacks, such as the need to throw away possibly informative features. In settings where we expect the distribution shift to be less than worst-case, this would be unnecessarily conservative. Furthermore, Zhao et al. [2019] show that marginal feature invariance cannot ensure generalization, and Stojanov et al. [2021] develop a toy setting where no single feature embedder can be sufficient.

Instead, we reframe the problem by thinking of each training domain as a distinct transformation from a shared canonical representation space. In this framing, we can "adjust" each domain in order to undo these transformations, aligning the representations to learn a single robust predictor in this unified space. Specifically, we propose to whiten each domain's features to have zero mean and identity covariance, which can be thought of as a "domain-specific batchnorm" but using the full feature covariance. This idea is not totally new: some prior works align the moments between domains to improve generalization. The difference is that DARE does not learn a single featurizer which aligns the moments, but rather *aligns the moments of already learned features*; the latter approach maintains useful variation between domains which would be eliminated by the former.

DARE is closer to methods which learn separate batchnorm parameters per domain over a deep network, possibly adjusting at test-time [Seo et al., 2019, Chang et al., 2019, Segù et al., 2020]—these methods perform well, but they are entirely heuristic based, difficult to optimize, and come with no formal guarantees. Our theoretical analysis thus serves as preliminary justification for the observed benefits of such methods, which have so far lacked serious grounding.

To begin, define $\mathcal{E}$ as the set of observed training environments, each of which is defined by a distribution $p^e(x, y)$ over features $x \in \mathbb{R}^d$ and class labels $y \in [k]$. For each $e \in \mathcal{E}$, denote the mean and covariance of the features as $\mu_e, \Sigma_e$. Our first step is to adjust the features via the whitening transformation $\Sigma_e^{-1/2}(x - \mu_e)$. If we believe that ERM is sufficient to recover a linear transformation of a ground-truth representation, whitening the data will then "undo" each domain's unique transformation. Unfortunately, we cannot undo the test transformation because we have no way of knowing what the test mean will be. Interestingly, this is not a problem provided the predictor satisfies a simple constraint. Suppose that the predictor's output on the mean representation of each environment is a multiple of the all-ones vector. As softmax is invariant to a constant offset, this enforces that the environment mean has *no effect* on the final probability distribution, and thus there is no need to adjust for the test-time mean. We therefore enforce this constraint during training, with the hope that the same invariance will approximately hold at test-time. Formally, the DARE objective finds a matrix $\beta \in \mathbb{R}^{d \times k}$ which solves

$$\min_{\beta} \sum_{e \in \mathcal{E}} \mathbb{E}_{p^e}[\ell(\beta^\top \Sigma_e^{-1/2} x,\ y)] \ \text{ subject to softmax} \left(\beta^\top \Sigma_e^{-1/2} \mu_e\right) = \frac{1}{k}\mathbf{1}.\ \forall e \in \mathcal{E}, \tag{7.1}$$

where $\ell$ is the multinomial logistic loss and we omit the bias $\beta_0$ for brevity. For binary classification, $\beta$ is a vector and the softmax is replaced with the logistic function—the constraint is then equivalent to requiring the mean of the logits to be 0. Thus, the DARE objective explicitly regresses on the adjusted features, while the constraint enforces that each environment mean has no effect on the output distribution to encourage predictions to also be invariant to test-time transformations. The astute reader will point out that we also do not know the correct whitening matrix for the test data—instead, we adjust using our best guess for the test covariance: denoting this estimate as $\bar{\Sigma}$, our prediction on a new sample $x$ is $f(x; \beta) = \text{softmax}\left(\beta^\top \bar{\Sigma}^{-1/2} x\right)$. We prove that this prediction is minimax so long as our guess is "sufficiently close", and in practice we find that simply averaging the training domain adjustments performs well. Table F.1 in the Appendix shows this average is actually quite close to the sample covariance of the (unseen) test domain, explaining the good performance.

Note that unlike many prior methods, DARE does not enforce invariance of the features themselves. Rather, it *aligns the representations* such that different domains share similar optimal predictors. To support this claim, Table F.2 displays the cosine similarity between optimal linear classifiers for individual domains—we observe a large increase in average similarity as a result of the feature adjustment. Further, in Section 7.5 we demonstrate that the DARE objective is in fact minimizing the worst-case risk under a constrained set of possible distribution shifts, and it is therefore less conservative than methods which require complete feature invariance.

**Implementation in practice.** Due to its convexity, the DARE objective is extremely simple to optimize. In practice, we finetune a deep network over the training data with ERM and then extract the features. Next, treating the frozen features of the training data as direct observations $x$, we minimize the empirical Lagrangian form:

$$\widehat{\mathcal{L}}_{\text{cls}}^{\lambda}(\beta) := \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \left[ \frac{1}{n_e} \sum_{i=1}^{n_e} \ell(\beta^\top \widehat{\Sigma}_e^{-1/2} x_i,\ y_i) + \lambda \ell(\beta^\top \widehat{\Sigma}_e^{-1/2} \widehat{\mu}_e, k^{-1}\mathbf{1}) \right],$$

where $\widehat{\mu}_e, \widehat{\Sigma}_e$ are the usual sample estimates and $n_e$ is the number of samples in environment $e$. We find that the solution is incredibly robust to the choice of $\lambda$, but it is natural to wonder whether each of the components above is necessary for the performance gains we observe. We ablate both the whitening operation and the use of the constraint (Appendix F.5) and see performance drops in both cases. We also consider estimating the test-time mean rather than enforcing invariance, but this results in substantially worse accuracy—the environment feature means vary quite a bit, so the estimate is usually inaccurate.

For regression, we consider the same setup but minimize mean squared error on targets $y \in \mathbb{R}$. Here, the DARE solution is constrained such that the mean output has no effect on the prediction, meaning each domain's mean prediction should be zero. We discuss this in more detail in Appendix F.1.1, along with an interesting connection to anchor regression [Rothenhäusler et al., 2021].

**Leveraging unlabeled test samples.** Another benefit to our approach is that the adjustments for each domain do not depend on labels, so given unlabeled samples from the test domain we can often do even better. In this case, it is preferable to solve equation 7.1 without the constraint and use the empirical test covariance for $\bar{\Sigma}$. Such access occurs in the setting of unsupervised domain adaptation, but unlike methods which use those samples while training (or even for test-time training), this adjustment can be done *just-in-time*, without any retraining! We name this task *Just-in-Time Unsupervised Domain Adaptation* (JIT-UDA), and we provide finite-sample risk bounds for the DARE objective in this setting. We believe JIT-UDA presents a promising direction for future theoretical research: it is much weaker—and arguably more realistic—to assume access to unlabeled samples only at test-time. JIT-UDA is also amenable to analyses beyond worst-case, such as minimizing regret when observing test data sequentially [Rosenfeld et al., 2022c].

## 7.4 A New Model of Distribution Shift

The DARE objective is based on the intuition that all domains jointly share a representation space and that they arise as unique transformations from this space.

To capture this notion mathematically, we model the joint distribution $p^e(\epsilon, y)$ over latents $\epsilon \in \mathbb{R}^d$ and label $y \in \{0, 1\}$, along with an environment-specific transformation to observations $x \in \mathbb{R}^d$ for each domain. In the fully general case, this transformation can take an arbitrary form and can be written as $x = T_e(\epsilon, y)$.

Our primary assumption is that $p^e(y \mid \epsilon)$ is constant for all domains. Our goal is thus to invert each transformation $T_e$ such that we are learning an invariant conditional $p(y \mid T_e^{-1}(x))$ (throughout we assume $T_e$ is invertible). One can also view this model as a generalization of covariate shift: where the usual assumption is constancy of $p(y \mid x)$, the inverse transformation gives a richer model which can more realistically capture real-world variation across domains. It is important to note that this model generalizes (and has *strictly weaker* requirements than) both IRM and domain-invariant representation learning, which can be recovered by assuming $T_e$ is the same for all environments. For example, the $T_e$ could correspond to different image "styles"; we then would hope to learn to eliminate individual style variations.

For a typical deep learning analysis we would model $T_e$ as a non-linear generative map from latents to high-dimensional observations. However, our finding that ERM features are good enough suggests that modeling the learned features as a simple function of the "true latents" $\epsilon$ is not unreasonable. In other words, we now consider $x$ to represent the frozen features of the trained network, and we expect this network to have already "undone" the majority of the complexity, resulting in observations which are a simple function of the ground truth. Accordingly, we consider the following model:

$$\epsilon = \epsilon_0 + b_e, \; y = \mathbf{1}\{\beta^{*T}\epsilon + \eta \geq 0\}, \; x = A_e\epsilon. \tag{7.2}$$

Here, $\epsilon_0 \sim p^e(\epsilon_0)$, which we allow to be *any domain-specific distribution*; we assume only that its mean is zero (such that $\mathbb{E}[\epsilon] = b_e$) and that its covariance exists. We fix $\beta^* \in \mathbb{R}^d$ for all domains and model $\eta$ as logistic noise. Finally, $A_e \in \mathbb{R}^{d\times d}, b_e \in \mathbb{R}^d$ are domain-specific. For regression, we model the same generative process for $x, \epsilon$, while for the response, we have $y \in \mathbb{R}$ and $\eta$ is zero-mean independent noise: $y = \beta^{*T}\epsilon + \eta$. We remark that the reason for separate definitions of $p^e(\epsilon_0)$ and $b_e$ is that our robustness guarantees are agnostic to the distribution of latent residuals $p^e(\epsilon_0)$—the *only* aspect of the latent distribution $p(\epsilon)$ which affects our bounds is the environment mean $b_e$.

**Connection to Invariant Prediction.** Statistical models of varying and invariant (latent) features have recently become a popular tool for analyzing the behavior of deep representation learning algorithms [Rosenfeld et al., 2021, Chen et al., 2022, Wald et al., 2021]. We see that Equation (7.2) can model such a setting by assuming, e.g., that a subspace of the columnspan of $A_e$ is constant for all $e$, while the remaining subspace can vary. In such a case, the features $x$ are expressed as the sum of a varying and an invariant component, and any minimax representation must remove the varying component, throwing away potentially useful information. Instead, DARE *realigns* these components so that we can use them at test-time. We illustrate this with the following running example:

*Example* 7.4.1 (Invariant Latent Subspace Recovery). Consider the model (Equation (7.2)) with $\epsilon \sim \mathcal{N}(0, I_{d_1+d_2})$. Define $\Pi := \begin{bmatrix} I_{d_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{d_2} \end{bmatrix}$ and assume $A_e = \begin{bmatrix} \Sigma^{1/2} & \mathbf{0} \\ \mathbf{0} & \Sigma_e^{1/2} \end{bmatrix}$ for all $e$, where $\Sigma \in \mathbb{R}^{d_1 \times d_1}$ is constant for all domains but $\Sigma_e \in \mathbb{R}^{d_2 \times d_2}$ varies.

Here we have a simple latent variable model: the features have the decomposition $x = A_e \epsilon = \Sigma^{1/2}\Pi\epsilon + \Sigma_e^{1/2}(I - \Pi)\epsilon$, where the first component is constant across environments and the second varies. It will be instructive at this point to analyze what an invariant prediction algorithm such as IRM would do in this setting. Here, the IRM constraint would enforce learning a featurizer $\Phi$ such that $\Phi(x)$ has an invariant relationship with the target. Under the above model, the solution is $\Phi(x) = \Pi A_e^{-1} x = \Sigma^{1/2}\Pi\epsilon$, retainining only the component lying in span$(\Pi)$. Crucially, removing these features actually results in *worse* performance on the training environments—IRM only does so because using these features could harm test performance in the worst case. However, projecting out the varying component in this manner is unnecessarily conservative, as we may expect that for a future distribution, $\Sigma_{e'}$ will not be *too* different from what we have seen before. Instead of projecting out this component, DARE performs a more nuanced alignment of environment subspaces, and it is thus able to take advantage of this additional informaton. We will see shortly the resulting benefits. This would imply that the minimax-optimal prediction must throw away the varying subspace. Instead, DARE *realigns* the subspaces so that we can use them at test-time.

## 7.5 Theoretical Analysis

Before we can analyze the DARE objective, we observe that there is a possible degeneracy in Equation (7.2), since two identical observational distributions $p(x, y)$ can have different regression vectors $\beta^*$. We therefore begin with a simple assumption:

**Assumption 7.5.1.** *Write the SVD of $A_e$ as $U_e S_e V_e^\top$. We assume $V_e = V \ \forall e$.*

One special case where this holds is when $A_e$ is constant for all domains; this is very similar to the "additive intervention" setting of Rothenhäusler et al. [2021], since only $p^e, b_e$ can vary. We assume WLOG that $V = I$, as any other value can be subsumed by the other parameters. We further let $\mathbb{E}[\epsilon_0 \epsilon_0^\top] = I$ WLOG by the same reasoning. With Assumption 7.5.1, we can uniquely recover $A_e = U_e S_e$ via the eigendecomposition of the covariance $\Sigma_e = A_e A_e^\top = U_e S_e^2 U_e^\top$. We therefore use the notation $\Sigma_e^{1/2}$ to refer to $A_e$ recovered in this way. We allow covariances to have zero eigenvalues, in which case we write the matrix inverse to implicitly refer to the pseudoinverse. As is standard in domain generalization analysis, unless stated otherwise we assume full distribution access to the training domains, though standard concentration inequalities could easily be applied.

**A remark on our assumptions.** Assumption 7.5.1 and the constraint in Equation (7.1) are not trivial. Domain generalization is an exceptionally difficult problem, and showing anything meaningful requires *some* assumption of consistency between train and test. Our assumptions are only as strong as necessary to prove our results, but future work could relax them, resulting in weaker but possibly still reasonable performance guarantees. In Appendix F.1.2, we discuss in greater detail the necessity of Assumption 7.5.1, as well as the additional conditions we assume without loss of generality and their implications for the DARE objective. Additionally, experiments in Appendix F.5 demonstrate that our covariance estimation is indeed accurate, and the fact that our method exceeds state of the art even with this strong constraint (and does worse without the constraint, see Figure F.3) is further evidence that our assumptions are reasonable.

We begin by deriving the solution to Equations (Equation (7.1)) and (Equation (F.1)). Recall that the DARE constraint requires that the mean representation of each domain has no effect on our prediction. To enforce this, the DARE solution must project out the subspace in which the means vary. Given a set of $E$ training environments, define $\mathbb{B}$ as the $d \times E$ matrix whose columns are the environmental mean parameters $b_e$. Throughout this section, we make use of the matrix $\widehat{\Pi}$, which is defined as the orthogonal projection onto the nullspace of $\mathbb{B}^\top$: $\widehat{\Pi} := I - \mathbb{B}\mathbb{B}^\dagger = U_{\widehat{\Pi}} S_{\widehat{\Pi}} U_{\widehat{\Pi}}^\top \in \mathbb{R}^{d \times d}$. This matrix projects onto the DARE constraint set, and it turns out to be all that is necessary to state the solution:

**Theorem 7.5.2.** (Closed-form solution to the DARE population objective). *Under model (Equation (7.2)), the solution to the* DARE *population objective (Equation (F.1)) for linear regression is* $\widehat{\Pi}\beta^*$. *If* $\epsilon$ *is Gaussian, then the solution for logistic regression (Equation (7.1)) is* $\alpha\widehat{\Pi}\beta^*$ *for some* $\alpha \in (0, 1]$.

The intuition behind the proof is as follows: due to the constraint, the centered DARE objective is equivalent to the uncentered objective—for the latter, the excess risk of a vector $\widehat{\beta}$ is $\mathbb{E}[(\widehat{\beta}^\top \epsilon - \beta^{*T}\epsilon)^2] = \|\widehat{\beta} - \beta^*\|_2^2$. Therefore, the solution is the

$\ell_2$-projection of $\beta^*$ onto the constraint set, which is precisely $\widehat{\Pi}\beta^*$.

In Example 7.4.1, we saw how invariant prediction will discard a large subspace of the representation and why this is undesirable. Instead, DARE undoes the environment transformation and regresses directly on $\epsilon = T_e^{-1}(x)$. Because we are performing a separate transformation for each environment, we are aligning the varying subspaces rather than throwing them away, allowing us to make use of additional information. Though the DARE solution also recovers $\beta^*$ up to a projection, it is using the adjusted features; DARE therefore only removes what cannot be aligned. In particular, whereas $\Pi$ has rank $d_1$ in Example 7.4.1, $\widehat{\Pi} = I$ would have full rank—this retains strictly more information: if we expect worst-case distribution shift we can still project to the invariant subspace, but if not then we can often perform much better. Indeed, in the ideal setting where we have a good estimate of $\Sigma_{e'}$ (e.g., under mild distribution shift or when solving JIT-UDA), we can make the Bayes-optimal prediction as $\mathbb{E}[y \mid x] = \beta^{*T} A_{e'}^{-1} x$. Thus we see a clear advantage that DARE enjoys over invariant prediction.

The second result of Theorem 7.5.2 depends on a key lemma (Lemma F.2.1 in the Appendix) about the closed-form solution to a projection-constrained logistic regression problem. The result is somewhat general and we expect it could be of independent interest, yet we found surprisingly few related results in the literature. Though we prove this lemma only for Gaussian $z$, we found empirically that the result approximately holds whenever $z$ is dimension-wise independent and symmetric about the origin., likely as a consequence of the Central Limit Theorem (see discussion in the Appendix).

### 7.5.1 The Adversarial Risk of DARE

Moving forward, we denote the DARE solution $\beta^*_{\widehat{\Pi}} := \widehat{\Pi}\beta^*$, with $\beta^*_{I-\widehat{\Pi}}$ defined analogously. We next study the behavior of the DARE solution under worst-case distribution shift. We consider the setting where an adversary directly observes our choices of $\bar{\Sigma}, \widehat{\beta}$ and chooses new environmental parameters $A_{e'}, b_{e'}$ so as to cause the greatest possible loss. Specifically, we study the square loss, defining the *excess* test risk of a predictor as $\mathcal{R}_{e'}(\widehat{\beta}) := \mathbb{E}_{p_{e'}}[(\widehat{\beta}^\top \bar{\Sigma}^{-1/2} x - \beta^{*T}\epsilon)^2]$ (we leave the dependence on $\bar{\Sigma}$ implicit). For logistic regression we therefore analyze the squared error with respect to the log-odds. With some abuse of notation, we also reference excess risk when only using a particular subspace, i.e. $\mathcal{R}^{\Pi}_{e'}(\widehat{\beta}) := \mathbb{E}_{p_{e'}}[(\widehat{\beta}_{\Pi}^\top \bar{\Sigma}^{-1/2} x - \beta_{\Pi}^{*T}\epsilon)^2]$.

Because we guess $\bar{\Sigma}$ before observing any data from this new distribution, ensuring success is impossible in the general case. Instead, we consider a set of restrictions on the adversary which will make the problem tractable. Define the error in our test domain adjustment as $\Delta := \Sigma_{e'}^{1/2}\bar{\Sigma}^{-1/2} - I$; observe that if $\bar{\Sigma} = \Sigma_{e'}$, then $\Delta = \mathbf{0}$. Our first assumption[3] says that the effect of our adjustment error with respect to the *interaction between subspaces* $\widehat{\Pi}$ and $(I - \widehat{\Pi})$ is bounded:

**Assumption 7.5.3** (Approximate recovery of subspaces)**.** *For a fixed constant* $B \geq 0$*,* $\|(I - \widehat{\Pi})\Delta\widehat{\Pi}\widehat{\beta}\|_2 \leq B\|\widehat{\Pi}\beta^*\|_2$.

*Remark* 7.5.4. To see specific cases when this would hold, consider the decomposition of $\Delta$ according to its components in the subspaces $\widehat{\Pi}$ and $I - \widehat{\Pi}$: $U_{\widehat{\Pi}}^\top \Delta U_{\widehat{\Pi}} = \begin{bmatrix} \Delta_1 & \Delta_{12} \\ \Delta_{21} & \Delta_2 \end{bmatrix}$, where $\Delta_1 \in \mathbb{R}^{\mathrm{rank}(\widehat{\Pi})\times\mathrm{rank}(\widehat{\Pi})}$. A few settings automatically satisfy Assumption 7.5.3 with $B = 0$, due to the fact that $U_{\widehat{\Pi}}^\top \Delta U_{\widehat{\Pi}}$ will be block-diagonal. In particular, this will be the case if all domains share an invariant subspace—e.g., if $\Pi A_e \Pi$ is constant as in Example 7.4.1. Below, we show that in this setting, exact recovery of this subspace occurs once we observe $\mathrm{rank}(I - \Pi)$ environments—this matches (actually, it is one less than) the linear environment complexity of most invariant predictors [Rosenfeld et al., 2021], and therefore Assumption 7.5.3 with $B = 0$ is no stronger than assuming a linear number of environments. This will similarly occur if $U_e$ is shared across all environments (e.g., under fixed $A_e$ as described above) and we use any sort of "averaging" guess of $\bar{\Sigma}$.

Our second assumption concerns the magnitude of our error in the non-varying subspace:

**Assumption 7.5.5** (Bound on adjustment error in non-varying subspace)**.** *Using only covariates in the non-varying subspace, the risk of the ground truth regressor* $\beta^*$ *is less than that of the trivial zero predictor:* $\mathcal{R}_{p_{e'}}^{\widehat{\Pi}}(\beta^*) < \mathcal{R}_{p_{e'}}^{\widehat{\Pi}}(\mathbf{0})$.

The need for this restriction should be immediate—if our adjustment error were so large that this did not hold, even the oracle regression vector would do worse than simply always predicting $\widehat{y} = 0$. Assumption 7.5.5 is satisfied for example if $\|\Delta\widehat{\Pi}\| < 1$, which again is guaranteed if there is an invariant subspace. Note that we make *no restriction* on the risk in the subspace $I - \widehat{\Pi}$—the adversary is allowed any amount of variation in directions where we have *already seen* variation in the mean terms $b_e$, but introduction of *new* variation is assumed bounded. **This is a no-free-lunch necessity:** if we have never seen a particular type of variation, we cannot possibly know how to use it at test-time.

With these restrictions on the adversary, our main result derives the supremum of the excess test risk of the DARE solution under adversarial distribution shift. Furthermore, we prove that this risk is *minimax*: making no more restrictions on the adversary other than a global bound on the mean, the DARE solution achieves the

[3]Though we label these as assumptions, they are properly interpreted as *restrictions* on an adversary—we consider an "uncertainty set" comprising all possible domains subject to these requirements.

best performance we could possibly hope for at test-time:

**Theorem 7.5.6** (DARE risk and minimaxity)**.** *For any $\rho \geq 0$, denote the set of possible test environments $\mathcal{A}_\rho$ which contains all parameters $(A_{e'}, b_{e'})$ subject to Assumptions Assumption 7.5.3 and Assumption 7.5.5 and a bound on the mean: $\|b_{e'}\|_2 \leq \rho$. For logistic or linear regression, let $\widehat{\beta}$ be the minimizer of the corresponding* DARE *objective as in Theorem 7.5.2. Then,*

$$\sup_{(A_{e'}, b_{e'}) \in \mathcal{A}_\rho} \mathcal{R}_{e'}(\widehat{\beta}) = (1+\rho^2)(\|\beta^*\|_2^2 + 2B\|\beta^*_{\widehat{\Pi}}\|_2 \|\beta^*_{I-\widehat{\Pi}}\|_2).$$

*Furthermore, the* DARE *solution is* minimax*:*

$$\widehat{\beta} \in \arg\min_{\beta \in \mathbb{R}^d} \sup_{(A_{e'}, b_{e'}) \in \mathcal{A}_\rho} \mathcal{R}_{e'}(\beta)$$

*Proof sketch.* We decompose the excess risk into error on the the target mean $\beta^{*T} b_{e'}$ and error on the residuals $\beta^{*T}\epsilon_0$, bounding the two separately. To show $\widehat{\beta}$ is minimax, we construct a series of adversarial choices of $A_{e'}, b_{e'}$ which force larger risk for any predictor which does not satisfy certain properties. These properties progressively eliminate possible predictors until all those which remain have the same adversarial risk as $\widehat{\beta}$. □

A special case when our assumptions hold is when all domains share an invariant subspace and we only predict using that subspace, but this is often too conservative. There are settings where allowing for (limited) new variation can improve our predictions, and Theorem 7.5.6 shows that DARE should outperform invariant prediction in such settings.

### 7.5.2 The Environment Complexity of DARE

An important new measure of domain generalization algorithms is their *environment complexity*, which describes how the test risk behaves as a function of the number of (possibly random) domains we observe. In contrast to Example 7.4.1, for this analysis we assume an invariant subspace $\Pi$ outside of which *both* $A_e$ and $b_e$ can vary arbitrarily—we formalize this as a prior over the $b_e$ whose covariance has the same span as $I - \Pi$. Thus the minimax vector is $\Pi\beta^*$ even after adjustment, so we can directly compare DARE to existing invariant prediction methods. Our next result demonstrates that DARE achieves the same threshold as prior methods, but we also prove the first *finite-environment convergence guarantee*, quantifying how quickly the risk of the DARE predictor approaches that of the minimax-optimal predictor. We begin by defining two quantities which appear in our bound.

**Definition 7.5.7.** The *effective rank* of a matrix $\Sigma$ is defined as $r(\Sigma) := \frac{\mathrm{Tr}(\Sigma)}{\|\Sigma\|_2}$ and satisfies $1 \leq r(\Sigma) \leq \mathrm{rank}(\Sigma)$.

**Definition 7.5.8.** Denote the eigenvalues in descending order as $\lambda_1 \geq \ldots \geq \lambda_d$. We define the smallest gap between consecutive eigenvalues: $\xi(\Sigma) := \min_{i \in [d-1]} \lambda_i - \lambda_{i+1}$.

**Theorem 7.5.9** (Environment complexity of DARE)**.** *Fix test parameters $A_{e'}, b_{e'}$ and guess $\bar{\Sigma}$. Suppose we minimize the* DARE *regression objective (Equation (F.1)) on environments whose means $b_e$ are Gaussian vectors with covariance $\Sigma_b$, with span$(\Sigma_b)$ = span$(I - \Pi)$. After seeing $E$ training domains:*

1. *If $E \geq \mathrm{rank}(\Sigma_b)$ then* DARE *recovers the minimax-optimal predictor almost surely: $\widehat{\beta} = \beta^*_\Pi$.*
2. *Otherwise, if $E \geq r(\Sigma_b)$ then with probability $\geq 1 - \delta$,*

$$\mathcal{R}_{e'}(\widehat{\beta}) \leq \mathcal{R}_{e'}(\beta^*_\Pi) + \mathcal{O}\left(\frac{\|\Sigma_b\|_2}{\xi(\Sigma_b)}\left(\sqrt{\frac{r(\Sigma_b)}{E}} + \max\left\{\sqrt{\frac{\log 1/\delta}{E}}, \frac{\log 1/\delta}{E}\right\}\right)\right),$$

*where $\mathcal{O}(\cdot)$ hides dependence on $\|\Delta\|_2$.*

For coherence we present the first item as a probabilistic statement, but it holds deterministically so long as there are $\mathrm{rank}(\Sigma_b)$ linearly independent observations of $b_e$.

*Proof sketch.* The proof analyzes the error in our recovery of the correct subspace $\|\Pi - \widehat{\Pi}\|_2$. Item 1 is immediate, as under this condition we have $\|\Pi - \widehat{\Pi}\|_2 = 0$. For item 2, we show how to bound $\mathcal{R}_{e'}(\widehat{\beta}) - \mathcal{R}_{e'}(\beta^*_\Pi)$ as $\mathcal{O}(\|\Pi - \widehat{\Pi}\|_2)$. We then invoke a variant of the Davis-Kahan theorem plus a spectral concentration inequality to derive the result. □

*Remark* 7.5.10. Prior analyses of invariant prediction methods only show a discontinuous threshold where the minimax predictor is discovered after seeing a fixed number of environments—usually linear in the non-invariant latent dimension—and DARE achieves a slightly better threshold. But one should expect that if the variation of environmental parameters is not *too* large then we can do better, and indeed Theorem 7.5.9 shows that if the *effective rank* of $\Sigma_b$ is sufficiently small, the risk of the DARE predictor will approach that of the minimax predictor as $\mathcal{O}(E^{-1/2})$.

### 7.5.3 Applying DARE to JIT-UDA

So far, we have only considered a setting with no knowledge of the test domain. As discussed in Section 7.3, we'd expect that estimating the adjustment via unlabeled samples will improve performance. Prior works have extensively explored how to leverage access to unlabeled test samples for improved generalization—but while some suggest ways of using unlabeled samples at test-time, they are not truly "just-in time", in that they still require updating the network parameters using the unlabeled data, with cost scaling with the size of the network. Furthermore, the advantages of access to such data at test time has not yet been formally quantified. Our final theorem investigates the provable benefits of using the empirical moments instead of enforcing invariance, giving a finite-sample convergence guarantee for the unconstrained DARE objective in the JIT-UDA setting:

**Theorem 7.5.11** (JIT-UDA, shortened)**.** *Assume the data follows the model equation 7.2 and that we observe $n_s = \Omega(m(\Sigma_S)d^2)$ samples from source distribution $\mathcal{N}(\mu_S, \Sigma_S)$ and $n_T = \Omega(m(\Sigma_T)d^2)$ samples from target distribution $\mathcal{N}(\mu_T, \Sigma_T)$. Suppose we solve for $\widehat{\beta}$ which minimizes the unconstrained,* uncentered *objective $\widehat{\mathcal{L}}^0_{reg}$ over the source data and predict $\widehat{\beta}^\top \widehat{\Sigma}_T^{-1/2} x$ on the target data. Then with probability at least $1 - 3d^{-1}$, the excess squared risk of our predictor on the new environment is bounded as*

$$\mathcal{R}_T(\widehat{\beta}) = \mathcal{O}\left(d^2 \|\mu_T\|_2^2 \left(\frac{m(\Sigma_S)}{n_S} + \frac{m(\Sigma_T)}{n_T}\right)\right).$$

The full theorem statement can be found in the Appendix. Experimentally, we found that DARE does not outperform methods specifically intended for UDA, possibly due to the fact that $n \ll d^2$—but we believe this is a promising direction for future theoretical research, since it doesn't require unlabeled samples at train-time and it can incorporate new data in a streaming fashion.

## 7.6 Experiments

Most algorithms implemented in the DomainBed benchmark are only applicable to deep networks; many apply complex regularization to either the learned features or the network itself. We instead compare to three popular algorithms which work for linear classifiers: ERM [Vapnik, 1999], IRM [Arjovsky et al., 2019], and GroupDRO [Sagawa et al., 2020a]. We also discovered that simply rescaling ERM by downweighting each domain proportionally to its size serves as a strong baseline. Concurrent work makes a similar observation, exploring this baseline when training the whole network rather than just the last layer [Idrissi et al., 2021]. We denote this method "Reweighted ERM" and we include it in our reported results in Table 7.1. We evaluate all approaches on four datasets: Office-Home [Venkateswara et al., 2017a], PACS [Li et al., 2017], VLCS [Fang et al., 2013], and DomainNet [Peng et al., 2019a]. The features for each trial are the default hyperparameter sweep of

| **Dataset / Algorithm** | **Mean Accuracy by Domain ($\pm$ 90% CI)** | | | |
|---|---|---|---|---|
| Office-Home | A | C | P | R |
| ERM | $61.4 \pm 1.9$ | $52.1 \pm 1.6$ | $74.6 \pm 0.7$ | $75.8 \pm 1.9$ |
| IRM | $62.1 \pm 1.5$ | $53.2 \pm 1.7$ | $75.2 \pm 0.9$ | $77.3 \pm 0.8$ |
| GroupDRO | $62.5 \pm 0.5$ | $53.1 \pm 1.7$ | $\underline{75.7 \pm 0.5}$ | $\underline{77.7 \pm 1.1}$ |
| Reweighted ERM | $62.5 \pm 0.8$ | $52.5 \pm 2.3$ | $\underline{75.7 \pm 0.5}$ | $\underline{77.4 \pm 0.8}$ |
| DARE | $\mathbf{64.7} \pm \mathbf{0.8}$ | $\mathbf{55.4} \pm \mathbf{1.1}$ | $\mathbf{77.5} \pm \mathbf{0.3}$ | $\mathbf{79.2} \pm \mathbf{0.5}$ |
| PACS | A | C | P | S |
| ERM | $84.3 \pm 1.5$ | $79.5 \pm 1.9$ | $96.7 \pm 0.5$ | $74.9 \pm 3.7$ |
| IRM | $83.6 \pm 0.4$ | $79.2 \pm 0.5$ | $97.1 \pm 0.2$ | $76.4 \pm 2.3$ |
| GroupDRO | $83.6 \pm 1.0$ | $79.1 \pm 0.2$ | $96.9 \pm 0.3$ | $\underline{76.8 \pm 2.8}$ |
| Reweighted ERM | $83.6 \pm 1.0$ | $78.8 \pm 0.2$ | $97.1 \pm 0.3$ | $76.5 \pm 2.3$ |
| DARE | $\mathbf{85.6} \pm \mathbf{0.6}$ | $80.5 \pm 1.0$ | $96.6 \pm 0.4$ | $76.7 \pm 2.0$ |

| **Dataset / Algorithm** | **Mean Accuracy by Domain ($\pm$ 90% CI)** | | | | | |
|---|---|---|---|---|---|---|
| VLCS | C | L | S | V | | |
| ERM | $97.6 \pm 0.7$ | $\mathbf{66.1} \pm \mathbf{0.6}$ | $71.9 \pm 1.9$ | $76.1 \pm 2.3$ | | |
| IRM | $\underline{98.6 \pm 0.6}$ | $64.9 \pm 0.9$ | $72.9 \pm 0.7$ | $74.1 \pm 2.1$ | | |
| GroupDRO | $\underline{98.5 \pm 0.7}$ | $65.0 \pm 0.5$ | $73.9 \pm 1.0$ | $75.1 \pm 1.8$ | | |
| Reweighted ERM | $\underline{98.6 \pm 0.7}$ | $65.5 \pm 0.5$ | $73.4 \pm 0.3$ | $74.9 \pm 1.7$ | | |
| DARE | $\underline{98.5 \pm 0.2}$ | $62.5 \pm 1.6$ | $74.3 \pm 1.4$ | $75.5 \pm 1.4$ | | |
| DomainNet | c | i | p | q | r | s |
| ERM | $57.6 \pm 0.6$ | $17.9 \pm 0.6$ | $44.7 \pm 0.7$ | $12.6 \pm 0.9$ | $59.0 \pm 0.7$ | $48.6 \pm 0.3$ |
| IRM | $\underline{60.9 \pm 0.4}$ | $\underline{19.3 \pm 0.2}$ | $\underline{47.6 \pm 0.4}$ | $12.4 \pm 0.4$ | $\underline{61.9 \pm 1.0}$ | $\underline{49.8 \pm 0.6}$ |
| GroupDRO | $57.8 \pm 0.7$ | $\underline{18.8 \pm 0.4}$ | $\underline{45.3 \pm 0.4}$ | $11.9 \pm 0.8$ | $59.5 \pm 1.0$ | $47.9 \pm 0.9$ |
| Reweighted ERM | $\underline{61.0 \pm 0.4}$ | $19.1 \pm 0.4$ | $\underline{47.4 \pm 0.4}$ | $12.3 \pm 0.6$ | $\underline{61.8 \pm 1.1}$ | $\underline{49.9 \pm 0.5}$ |
| DARE | $\mathbf{61.7} \pm \mathbf{0.2}$ | $\underline{19.3 \pm 0.1}$ | $\underline{47.4 \pm 0.5}$ | $\mathbf{13.1} \pm \mathbf{0.7}$ | $\underline{61.2 \pm 1.3}$ | $\mathbf{51.4} \pm \mathbf{0.3}$ |

Table 7.1: Performance of *linear* predictors on top of fixed features learned via ERM. Each letter is a domain. Because all algorithms use the same set of features for each trial, results are not independent. Therefore, **bold** indicates highest mean according to one-sided paired t-tests at $p = 0.1$ significance. If not the overall highest, $\underline{\text{underline}}$ indicates higher mean than ERM under the same test.

DomainBed; for computational reasons, we used fewer random hyperparameter choices per trial, meaning the reported accuracies are not directly comparable. Nevertheless, our results are significant because they are consistently evaluated according to the same methodology. Additional comparisons to other end-to-end methods can be found in Appendix F.5.

We find that DARE consistently matches or surpasses prior methods. All algorithms use the same set of features for each trial, so their performances are highly dependent. To account for this, we perform a one-sided paired t-test between algorithms to determine the best performers. The fact that DARE consistently bests ERM is somewhat surprising: it is intended to guard against a worst-case distribution shift and depends on the quality of our guess $\bar{\Sigma}$, so we would in general expect worse performance on some datasets.

**Prior methods now consistently outperform ERM.** It is interesting that the previously observed gap between ERM and alternatives disappears in this setting with a linear predictor. For example, Gulrajani and Lopez-Paz [2021] report IRM and GroupDRO performing much worse on DomainNet—5-10% lower accuracy on some domains—but they surpass ERM when using frozen features. This could be because they learn worse features, or perhaps they are just more difficult to optimize over a deep network. This also implies that when these methods *do* work, it is most likely due to learning a better linear classifier. This further motivates work on methods for learning simpler robust predictors.

## 7.7 Related Work

A popular approach to domain generalization matches the domains in feature space, either by aligning moments [Sun et al., 2016a] or with an adversarial loss [Ganin et al., 2016], though these methods are known to be inadequate in general [Zhao et al., 2019]. DARE differs from these approaches in that the constraint requires only that the *feature mean projection onto the vector* $\beta$ be invariant. Domain-invariant projections [Baktashmotlagh et al., 2013] were recently analyzed by Chen and Bühlmann [2021], though notably under fully observed features and only for domain adaptation. They analyze other invariances as well, and we expect combining these methods with domain adjustment can serve as a direction for future study.

There has been intense recent focus on invariant prediction, based on ideas from causality [Peters et al., 2016] and catalyzed by IRM [Arjovsky et al., 2019]. Though the goal of such methods is minimax-optimality under major distribution shift, later work identifies critical failure modes of this approach [Rosenfeld et al., 2021, Kamath et al., 2021]. As discussed in Example 7.4.1, these methods eliminate features whose information is not invariant, which is often overly conservative. DARE instead allows for *limited* new variation by aligning the non-invariant subspaces, enabling stronger theoretical guarantees.

Some prior works "normalize" each domain by learning separate batchnorm parameters but sharing the rest of the network. This was initially suggested for UDA [Li et al., 2016, Bousmalis et al., 2016, Chang et al., 2019], which is not directly comparable to DARE since it requires unlabeled test data. This idea has also been applied to domain generalization [Seo et al., 2019, Segù et al., 2020] but in an ad-hoc manner. Because of the difficulty in training the network end-to-end, there is no consistent method for optimizing or validating the objective—in particular, all deep domain generalization methods were recently called into question when Gulrajani and Lopez-Paz [2021] gave convincing evidence that nothing beats ERM when evaluated fairly. Nevertheless, our analysis provides an initial justification for these methods, suggesting that this idea is worth exploring further.

# Chapter 8

# (Almost) Provable Error Bounds Under Distribution Shift via Disagreement Discrepancy

This chapter is based on Rosenfeld et al. [2022b]:
Rosenfeld, E. & Garg, S.
(Almost) Provable Error Bounds Under Distribution Shift via Disagreement Discrepancy.
In *Thirty-sixth Conference on Neural Information Processing Systems*, 2023.

## 8.1 Introduction

When deploying a model, it is important to be confident in how it will perform under inevitable distribution shift. Standard methods for achieving this include data dependent uniform convergence bounds [Mansour et al., 2009, Ben-David et al., 2006] (typically vacuous in practice) or assuming a precise model of how the distribution can shift [Rahimian and Mehrotra, 2019, Chen et al., 2022, Rosenfeld et al., 2021]. Unfortunately, it is difficult or impossible to determine how severely these assumptions are violated by real data ("all models are wrong"), so practitioners usually cannot trust such bounds with confidence.

To better estimate test performance in the wild, some recent work instead tries to directly predict accuracy of neural networks using unlabeled data from the test distribution of interest, [Garg et al., 2022, Baek et al., 2022, Lu et al., 2023]. While these methods predict the test performance surprisingly well, they lack pointwise trustworthiness and verifiability: their estimates are good on average over all distribution shifts, but they provide no guarantee or signal of the quality of any individual prediction (here, each point is a single test *distribution*, for which a method predicts a classifier's average accuracy). Because of the opaque conditions under which these methods work, it is also difficult to anticipate their failure cases—indeed, it is reasonably common for them to substantially overestimate test accuracy for a particular shift, which is problematic when optimistic deployment can be costly or catastrophic. Worse yet, we find that this gap *grows with test error* (Figure 8.1), making these predictions least reliable under large distribution shift, which is precisely when their reliability is most important. Although it is clearly impossible to guarantee upper bounds on test error for all shifts, there is still potential for error bounds that are intuitive and reasonably trustworthy.

In this work, we develop a method for (almost) provably bounding test error of classifiers under distribution shift using unlabeled test points. Our bound's only requirement is a simple, intuitive, condition which describes the ability of a hypothesis class to achieve small loss on a particular objective defined over the (unlabeled) train and test distributions. Inspired by $\mathcal{H}\Delta\mathcal{H}$-divergence [Ben-David et al., 2010b], our method requires training a critic to maximize agreement with the classifier of interest on the source distribution while simultaneously maximizing *dis*agreement on the target distribution; we refer to this joint objective as the *disagreement discrepancy*, and so we name the method $\textsc{Dis}^2$. We optimize this discrepancy over linear classifiers using deep features—or linear functions thereof—finetuned on only the training set. Recent evidence suggests that such representations are sufficient for highly expressive classifiers even under large distribution shift [Rosenfeld et al., 2022b]. Experimentally, we find that our bound is valid effectively 100% of the time,[1] consistently giving non-trivial lower bounds on test accuracy which are reasonably comparable to competitive baselines.

Additionally, we empirically show that it is even possible to approximately test this bound's likelihood of being satisfied with only unlabeled data: the optimization process itself provides useful information about the bound's validity, and we use this to construct a score which linearly correlates with the tightness of the bound. This score can then be used to relax the original bound into a sequence of successively tighter-yet-less-conservative estimates, interpolating between robustness and accuracy and allowing a user to make estimates according to their specific risk tolerance.

While maximizing agreement is statistically well understood, our method also

[1]The few violations are expected a priori, have an obvious explanation, and only occur for a specific type of learned representation. We defer a more detailed discussion of this until after we present the bound.

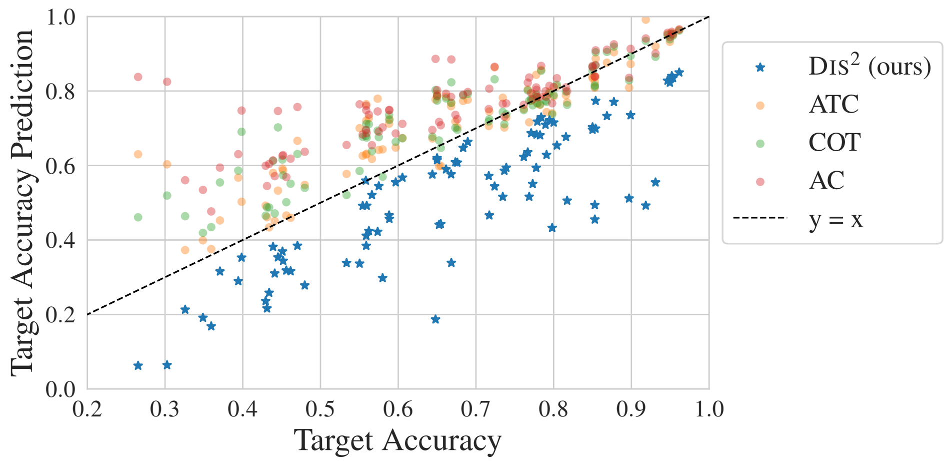


Figure 8.1: **Our bound vs. three prior methods for estimation across a wide variety of distribution shift benchmarks** (e.g., WILDs, BREEDs, DomainNet) and training methods (e.g., ERM, FixMatch, BN-adapt). Prior methods are accurate on average, but it is difficult or impossible to know when a given prediction is reliable and why. Worse yet, they usually overestimate accuracy, with the gap growing as test accuracy decreases—*this is precisely when a reliable, conservative estimate is most desirable.* Instead, Dɪs$^2$ maximizes the **dis**agreement **dis**crepancy to give a reliable error upper bound which holds effectively 100% of the time.

calls for maximizing *dis*agreement on the target distribution. This is not so straightforward in the multiclass setting, and we observe that prior works use unsuitable losses which do not correspond to minimizing the 0-1 loss of interest and are non-convex (or even *concave*) in the model logits [Chuang et al., 2020, Pagliardini et al., 2023]. To rectify this, we derive a new "disagreement loss" which serves as an effective proxy loss for maximizing multiclass disagreement. Experimentally, we find that minimizing this loss results in lower risk (that is, higher disagreement) compared to prior methods, and we believe it can serve as a useful drop-in replacement for any future methods which require maximizing multiclass disagreement.

Experiments across numerous vision datasets (BREEDs [Santurkar et al., 2020], FMoW-WILDs [Koh et al., 2021], Visda [Peng et al., 2017], Domainnet [Peng et al., 2019b], CIFAR10, CIFAR100 [Krizhevsky and Hinton, 2009] and OfficeHome [Venkateswara et al., 2017b]) demonstrate the effectiveness of our bound. Though Dɪs$^2$ is competitive with prior methods for error estimation, we emphasize that our focus is *not* on improving raw predictive accuracy—rather, we hope to obtain reliable (i.e., conservative), reasonably tight bounds on the test error of a given classifier under distribution shift. In particular, while existing methods tend to severely overestimate accuracy as the true accuracy drops, our bound maintains its validity while remaining non-vacuous, *even for drops in accuracy as large as 70%*. In addition to source-only training, we experiment with unsupervised domain adaptation methods that use unlabeled target data and show that our observations continue to hold.

## 8.2 Related Work

**Estimating test error with unlabeled data.** The generalization capabilities of overparameterized models on in-distribution data have been extensively studied using conventional machine learning tools [Neyshabur et al., 2015, 2017, Neyshabur, 2017, Neyshabur et al., 2018, Dziugaite and Roy, 2017, Bartlett et al., 2017, Zhou et al., 2018, Long and Sedghi, 2019, Nagarajan and Kolter, 2019a]. This research aims to bound the generalization gap by evaluating complexity measures of the trained model. However, these bounds tend to be numerically loose compared to actual generalization error [Zhang et al., 2016, Nagarajan and Kolter, 2019b]. Another line of work instead explores the use of unlabeled data for predicting in-distribution generalization [Platanios et al., 2016, 2017, Garg et al., 2021, Nakkiran and Bansal, 2019, Jiang et al., 2022b]. More relevant to our work, there are several methods that predict the error of a classifier under distribution shift with unlabeled test data: (i) methods that explicitly predict the correctness of the model on individual unlabeled points [Deng and Zheng, 2021, Deng et al., 2021, Chen et al., 2021a]; and (ii) methods that directly estimate the overall error without making a pointwise prediction [Chen et al., 2021b, Guillory et al., 2021, Chuang et al., 2020, Garg et al., 2022, Baek et al., 2022].

To achieve a consistent estimate of the target accuracy, several works require calibration on the target domain [Jiang et al., 2022b, Guillory et al., 2021]. However, these methods often yield poor estimates because deep models trained and calibrated on a source domain are not typically calibrated on previously unseen domains [Ovadia et al., 2019]. Additionally, Deng and Zheng [2021], Guillory et al. [2021] require a subset of labeled target domains to learn a regression function that predicts model performance—but thus requires significant a priori knowledge about the nature of shift that, in practice, might not be available before models are deployed in the wild.

Closest to our work is Chuang et al. [2020], where the authors use domain-invariant predictors as a proxy for unknown target labels. However, there are several crucial differences. First, like other works, their method only *estimates* the target accuracy—the error bounds they derive are not reasonably computable in practice. Second, their method relies on multiple approximations and the tuning of numerous hyperparameters, e.g. a threshold term and multiple lagrangian multipliers which trade off strengths of different regularizers. Their approach is also computationally demanding; as a result, proper tuning is difficult and the method does not scale to modern deep networks. Finally, they suggest minimizing the (concave) negative cross-entropy loss, but we show that this can be a poor proxy for maximizing disagreement, and we propose a more suitable replacement which performs much better in practice.

**Uniform convergence bounds.** Our bound is inspired by classic analyses using $\mathcal{H}$- and $\mathcal{H}\Delta\mathcal{H}$-divergence [Mansour et al., 2009, Ben-David et al., 2006, 2010b]. These provide error bounds via a complexity measure that is both data- and hypothesis-class-dependent. This motivated a long line of work on training classifiers with small corresponding complexity, such as restricting classifiers' discriminative power between source and target data [Ganin et al., 2016, Sun et al., 2016b, Long et al., 2018, Zhang et al., 2019]. Unfortunately, such bounds are often intractable to evaluate and are usually vacuous in real world settings. We provide a more detailed comparison between such bounds and our approach in Section 8.3.1.

## 8.3 Deriving an (Almost) Provable Error Bound

**Notation.** Let $\mathcal{S}, \mathcal{T}$ denote the source and target (train and test) distributions, respectively, over labeled inputs $(x, y) \in \mathcal{X} \times \mathcal{Y}$, and let $\widehat{\mathcal{S}}$, $\widehat{\mathcal{T}}$ denote sets of samples from them with cardinalities $n_S$ and $n_T$ (they also denote the corresponding empirical distributions). Recall that we observe only the covariates $x$ without the label $y$ when a sample is drawn from $\mathcal{T}$. We consider classifiers $h : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}$ which output a vector of logits, and we let $\widehat{h}$ denote the particular classifier whose error we aim to bound. Generally, we use $\mathcal{H}$ to denote a hypothesis class of such classifiers. Occasionally, where clear from context, we use $h(x)$ to refer to the argmax logit, i.e. the predicted class. We treat these classifiers as deterministic throughout, though our analysis can easily be extended to probabilistic classifiers and labels. For a distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, let $\epsilon_{\mathcal{D}}(h, h') := \mathbb{E}_{\mathcal{D}}[\mathbf{1}\{\arg\max_y h(x)_y \neq \arg\max_y h'(x)_y\}]$ denote the one-hot disagreement between classifiers $h$ and $h'$ on $\mathcal{D}$. Let $y^*$ represent the true labeling function such that $y^*(x) = y$ for all samples $(x, y)$; with some abuse of notation, we write $\epsilon_{\mathcal{D}}(h)$ to mean $\epsilon_{\mathcal{D}}(h, y^*)$, i.e. the 0-1 error of classifier $h$ on distribution $\mathcal{D}$.

The bound we derive in this work is extremely simple and relies on one new concept:

**Definition 8.3.1.** The *disagreement discrepancy* $\Delta(h, h')$ is the disagreement between $h$ and $h'$ on $\mathcal{T}$ minus their disagreement on $\mathcal{S}$:

$$\Delta(h, h') := \epsilon_{\mathcal{T}}(h, h') - \epsilon_{\mathcal{S}}(h, h').$$

We leave the dependence on $\mathcal{S}, \mathcal{T}$ implicit. Note that this term is symmetric and signed—it can be negative. With this definition, we now have the following lemma:

**Lemma 8.3.2.** *For any classifier $h$, $\epsilon_{\mathcal{T}}(h) = \epsilon_{\mathcal{S}}(h) + \Delta(h, y^*)$.*

*Proof.* By definition, $\epsilon_{\mathcal{T}}(h) = \epsilon_{\mathcal{S}}(h) + (\epsilon_{\mathcal{T}}(h) - \epsilon_{\mathcal{S}}(h)) = \epsilon_{\mathcal{S}}(h) + \Delta(h, y^*)$. $\square$

We cannot directly use Lemma 8.3.2 to estimate $\epsilon_{\mathcal{T}}(\widehat{h})$ because the second term is unknown. However, observe that $y^*$ is *fixed*. That is, while a learned $\widehat{h}$ will depend on $y^*$—and therefore $\Delta(\widehat{h}, y^*)$ may be large under large distribution shift—$y^*$ **is *not* chosen to maximize $\Delta(\widehat{h}, y^*)$ in response to the $\widehat{h}$ we have learned.** This means that for a sufficiently expressive hypothesis class $\mathcal{H}$, it should be possible to identify an alternative labeling function $h' \in \mathcal{H}$ for which $\Delta(\widehat{h}, h') \geq \Delta(\widehat{h}, y^*)$ (we refer to such $h'$ as the *critic*). In other words, we should be able to find an $h' \in \mathcal{H}$ for which its *implied* error gap $\epsilon_{\mathcal{T}}(\widehat{h}, h') - \epsilon_{\mathcal{S}}(\widehat{h}, h')$—i.e., the error gap if we assume $h'$ is the true labeling function—is at least as large as the *true* error gap $\epsilon_{\mathcal{T}}(\widehat{h}) - \epsilon_{\mathcal{S}}(\widehat{h})$. This key observation serves as the basis for our bound, and we discuss it in greater detail in Section 8.3.1.

In this work we consider the class $\mathcal{H}$ of linear critics, with $\mathcal{X}$ defined as source-finetuned deep neural representations or the resulting logits output by $\widehat{h}$. Prior work provides strong evidence that this class has surprising capacity under distribution shift, including the possibility that functions very similar to $y^*$ lie in $\mathcal{H}$ [Rosenfeld et al., 2022b, Kirichenko et al., 2022, Kang et al., 2020]. We formalize this intuition with the following assumption:

**Assumption 8.3.3.** *Define $h^* := \arg\max_{h' \in \mathcal{H}} \Delta(\widehat{h}, h')$. We assume*

$$\Delta(\widehat{h}, y^*) \leq \Delta(\widehat{h}, h^*).$$

Note that this statement is only meaningful when considering restricted $\mathcal{H}$ which may not contain $y^*$, as we do here. Note also that this assumption is made specifically for $\widehat{h}$, i.e. on a per-classifier basis. This is important because while the above may not hold for every classifier $\widehat{h}$, it need only hold for the classifiers whose error we would hope to bound, which is in practice a very small subset of classifiers (such as those which can be found by approximately minimizing the empirical training risk via SGD). From Lemma 8.3.2, we immediately have the following result:

**Proposition 8.3.4.** *Under Assumption 8.3.3, $\epsilon_{\mathcal{T}}(\widehat{h}) \leq \epsilon_{\mathcal{S}}(\widehat{h}) + \Delta(\widehat{h}, h^*)$.*

Unfortunately, identifying the optimal critic $h^*$ is intractable, meaning this bound is still not estimable—we present it as an intermediate result for clarity of presentation. To derive the practical bound we report in our experiments, we need one additional step. In Section 8.4, we derive a "disagreement loss" which we use to approximately maximize the empirical disagreement discrepancy $\widehat{\Delta}(\widehat{h}, \cdot) = \epsilon_{\widehat{\mathcal{T}}}(\widehat{h}, \cdot) - \epsilon_{\widehat{\mathcal{S}}}(\widehat{h}, \cdot)$. Relying on this loss, we instead make the assumption:

**Assumption 8.3.5.** *Suppose we identify the critic $h' \in \mathcal{H}$ which maximizes a concave surrogate to the empirical disagreement discrepancy. We assume $\Delta(\widehat{h}, y^*) \leq \Delta(\widehat{h}, h')$.*

This assumption is slightly stronger than Assumption 8.3.3—in particular, Assumption 8.3.3 implies with high probability a weaker version of Assumption 8.3.5 with additional terms that decrease with increasing sample size and a tighter proxy loss.[2] Thus, the difference in strength between these two assumptions shrinks as the number of available samples grows and as the quality of our surrogate objective improves. Ultimately, our bound holds without these terms, implying that the stronger assumption is reasonable in practice. We can now present our main bound:

**Theorem 8.3.6** (Main Bound)**.** *Under Assumption 8.3.5, with probability $\geq 1 - \delta$,*

$$\epsilon_{\mathcal{T}}(\widehat{h}) \leq \epsilon_{\widehat{\mathcal{S}}}(\widehat{h}) + \widehat{\Delta}(\widehat{h}, h') + \sqrt{\frac{(n_S + 4n_T)\log {}^1\!/\!_\delta}{2n_S n_T}}.$$

*Proof.* Assumption 8.3.5 implies $\epsilon_{\mathcal{T}}(\widehat{h}) \leq \epsilon_{\mathcal{S}}(\widehat{h}) + \Delta(\widehat{h}, h') = \epsilon_{\mathcal{S}}(\widehat{h}, y^*) + \epsilon_{\mathcal{T}}(\widehat{h}, h') - \epsilon_{\mathcal{S}}(\widehat{h}, h')$, so the problem reduces to upper bounding these three terms. We define the random variables

$$r_{\mathcal{S},i} = \begin{cases} {}^1\!/\!_{n_S}, & h'(x_i) = \widehat{h}(x_i) \neq y_i, \\ -{}^1\!/\!_{n_S}, & h'(x_i) \neq \widehat{h}(x_i) = y_i, \\ 0, & \text{otherwise} \end{cases} \qquad r_{\mathcal{T},i} = \frac{\mathbf{1}\{\widehat{h}(x_i) \neq h'(x_i)\}}{n_T}$$

for source and target samples, respectively. By construction, the sum of all of these variables is precisely $\epsilon_{\widehat{\mathcal{S}}}(\widehat{h}, y^*) + \epsilon_{\widehat{\mathcal{T}}}(\widehat{h}, h') - \epsilon_{\widehat{\mathcal{S}}}(\widehat{h}, h')$ (note these are the empirical terms). Further, observe that

$$\mathbb{E}\left[\sum_{\mathcal{S}} r_{\mathcal{S},i}\right] = \mathbb{E}_{\mathcal{S}}[\mathbf{1}\{\widehat{h}(x_i) \neq y_i\} - \mathbf{1}\{\widehat{h}(x_i) \neq h'(x_i)\}] = \epsilon_{\mathcal{S}}(\widehat{h}, y^*) - \epsilon_{\mathcal{S}}(\widehat{h}, h'),$$

$$\mathbb{E}\left[\sum_{\mathcal{T}} r_{\mathcal{T},i}\right] = \mathbb{E}_{\mathcal{T}}[\mathbf{1}\{\widehat{h}(x_i) \neq h'(x_i)\}] = \epsilon_{\mathcal{T}}(\widehat{h}, h'),$$

[2]Roughly, Assumption 8.3.3 implies $\Delta(\widehat{h}, y^*) \leq \Delta(\widehat{h}, h') + \mathcal{O}\left(\sqrt{\frac{\log {}^1\!/\!_\delta}{\min(n_S, n_T)}}\right) + \gamma$, where $\gamma$ is a data-dependent measure of how tightly the surrogate loss bounds the 0-1 loss in expectation.

and thus the expectation of their sum is $\epsilon_{\mathcal{S}}(\widehat{h}, y^*) + \epsilon_{\mathcal{T}}(\widehat{h}, h') - \epsilon_{\mathcal{S}}(\widehat{h}, h')$ (the population terms). Now we apply Hoeffding's inequality: the probability that the expectation exceeds their sum by $t$ is no more than $\exp\left(-\frac{2t^2}{n_{\mathcal{S}}(2/n_{\mathcal{S}})^2 + n_{\mathcal{T}}(1/n_{\mathcal{T}})^2}\right)$. Solving for $t$ completes the proof. $\square$

*Remark* 8.3.7. While we state Theorem 8.3.6 as an implication, **Assumption 8.3.5 is *equivalent* to the stated bound up to finite-sample terms**. Our empirical findings (and prior work) suggest that Assumption 8.3.5 is reasonable in general, but this equivalence allows us to actually *prove* that it holds in practice for some shifts. We elaborate on this in Appendix G.5.

The high-level statement of Theorem 8.3.6 is that if there is a simple (e.g., linear) critic $h' \in \mathcal{H}$ with large disagreement discrepancy, $\widehat{h}$ could have high error—likewise, if no critic achieves large discrepancy, we should expect low error. Here we gain a deeper understanding of the conceptual idea behind Assumption 8.3.5 and what it allows us to say *via* Theorem 8.3.6 about error under distribution shift. We distill this into the following core principle:

**Principle 8.3.8.** Consider the set of labeling functions which are approximately consistent with the labels on the source data. If all "reasonably simple" functions in this set would imply a classifier has low error on the target data, we should consider it *more likely* that the classifier indeed has low error than that the true labeling function is very complex.

In particular, if we accept the premise that existing representations are good enough that a simple function can achieve high accuracy under distribution shift [Rosenfeld et al., 2022b], it follows—up to some approximation—that the only labeling functions we need consider are the ones which are simple and agree with $y^*$ on $\mathcal{S}$.

*Remark* 8.3.9. Bounding error under distribution shift is fundamentally impossible without assumptions. Prior works which estimate accuracy using unlabeled data rely on experiments, suggesting that whatever condition allows their method to work holds in a variety of settings [Garg et al., 2022, Baek et al., 2022, Lu et al., 2023, Jiang et al., 2022b, Guillory et al., 2021]; using these methods is equivalent to *implicitly* assuming that it will hold for future shifts. Understanding these conditions is thus crucial for assessing in a given scenario whether they can be expected to be satisfied.[3] It is therefore of great practical value that Assumption 8.3.5 is a simple,

[3]Whether and when to trust a black-box estimate that is consistently accurate in all observed settings is a centuries-old philosophical problem [Hume, 2000] which we do not address here. Regardless,

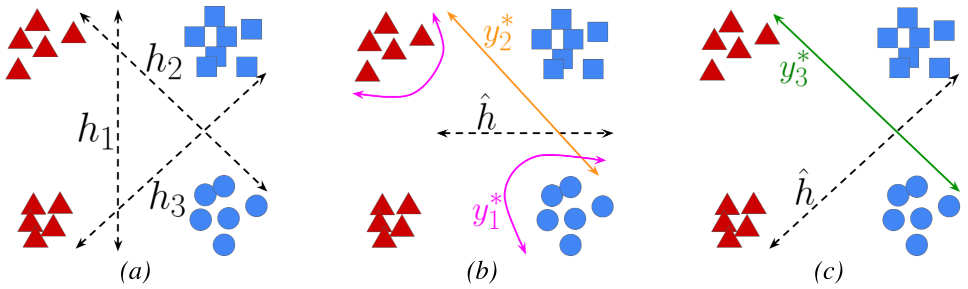


Figure 8.2: **The advantage of DIS$^2$ over bounds based on $\mathcal{H}$- and $\mathcal{H}\Delta\mathcal{H}$-divergence.** Consider the task of classifying circles and squares (triangles are unlabeled). **(a):** Because $h_1$ and $h_2 \oplus h_3$ perfectly discriminate between $\mathcal{S}$ (blue) and $\mathcal{T}$ (red), $\mathcal{H}$- and $\mathcal{H}\Delta\mathcal{H}$-divergence bounds are always vacuous. In contrast, DIS$^2$ is only vacuous when 0% accuracy is induced by a reasonably likely ground truth (such as $y_3^*$ in **(c)**, but not $y_1^*$ in **(b)**), and can often give non-vacuous bounds (such as $y_2^*$ in **(b)**).

intuitive requirement: below we demonstrate that this simplicity allows us to a identify a potential failure case *a priori*.

### 8.3.1 How Does DIS$^2$ Improve over $\mathcal{H}$- and $\mathcal{H}\Delta\mathcal{H}$-Divergence?

To verifiably bound a classifier's error under distribution shift, one must develop a meaningful notion of distance between distributions. One early attempt at this was *$\mathcal{H}$-divergence* [Ben-David et al., 2006, Mansour et al., 2009] which measures the ability of a binary hypothesis class $\mathcal{H}$ to discriminate between $\mathcal{S}$ and $\mathcal{T}$ in feature space. This was later refined to *$\mathcal{H}\Delta\mathcal{H}$-divergence* [Ben-David et al., 2010b], which is equal to $\mathcal{H}$-divergence where the discriminator class comprises all exclusive-ors between pairs of functions from the original class $\mathcal{H}$. Though these measures can in principle provide non-vacuous bounds, they usually do not, and evaluating them is often intractable (particularly $\mathcal{H}\Delta\mathcal{H}$, because it requires maximizing an objective over all *pairs* of hypotheses). Furthermore, these bounds are overly conservative even for simple function classes and distribution shifts because they rely on uniform convergence. In practice, *we do not care* about bounding the error of all classifiers in $\mathcal{H}$—we only care to bound the error of $\widehat{h}$. This is a clear advantage of DIS$^2$ over $\mathcal{H}\Delta\mathcal{H}$.

**The true labeling function is never worst-case.**[4] More importantly, as

Figure 8.1 shows that these estimates are *not* consistently accurate, making interpretability that much more important.

[4]If it were, we'd see exactly 0% test accuracy—and when does that ever happen?

Principle 8.3.8 exemplifies, one should not expect the distribution shift to be *truly* worst-case, because the test distribution $\mathcal{T}$ and ground truth $y^*$ are not chosen adversarially with respect to $\widehat{h}$ [Rosenfeld et al., 2022c]. Figure 8.2 gives a simple demonstration of this point. Consider the task of learning a linear classifier to discriminate between squares and circles on the source distribution $\mathcal{S}$ (blue) and then bounding the error of this classifier on the target distribution $\mathcal{T}$ (red), whose true labels are unknown and are therefore depicted as triangles. Figure 8.2a demonstrates that both $\mathcal{H}$- and $\mathcal{H}\Delta\mathcal{H}$-divergence achieve their maximal value of 1, because both $h_1$ and $h_2 \oplus h_3$ perfectly discriminate between $\mathcal{S}$ and $\mathcal{T}$. Thus both bounds would be vacuous.

Now, suppose we were to learn the max-margin $\widehat{h}$ on the source distribution (Figure 8.2b). It is *possible* that the true labels are given by the worst-case boundary as depicted by $y_1^*$ (pink), thus "flipping" the labels and causing $\widehat{h}$ to have 0 accuracy on $\mathcal{T}$. In this setting, a vacuous bound is correct. However, this seems rather unlikely to occur in practice—instead, recent experimental evidence [Rosenfeld et al., 2022b, Kirichenko et al., 2022, Kang et al., 2020] suggests that the true $y^*$ will be much simpler. The maximum disagreement discrepancy here would be approximately $0.5$, giving a test accuracy lower bound of $0.5$—this is consistent with plausible alternative labeling functions such as $y_2^*$ (orange). Even if $y^*$ is not linear, we may still expect that *some* linear function will induce larger discrepancy; this is precisely Assumption 8.3.3. Suppose instead we learn $\widehat{h}$ as depicted in Figure 8.2c. Then a simple ground truth such as $y_3^*$ (green) is plausible, which would mean $\widehat{h}$ has 0 accuracy on $\mathcal{T}$. In this case, $y_3^*$ is also a critic with disagreement discrepancy equal to 1, and so $\textsc{Dis}^2$ would correctly output an error upper bound of $1$.

**A setting where $\textsc{Dis}^2$ may be invalid.** There is one setting where it should be clear that Assumption 8.3.5 is less likely to be satisfied: when the representation we are using is explicitly regularized to keep $\max_{h' \in \mathcal{H}} \Delta(\widehat{h}, h')$ small. This occurs for domain-adversarial representation learning methods such as DANN [Ganin et al., 2016] and CDAN [Long et al., 2018], which penalize the ability to discriminate between $\mathcal{S}$ and $\mathcal{T}$ in feature space. Given a critic $h'$ with large disagreement discrepancy, the discriminator $D(x) = \mathbf{1}\{\arg\max_y \widehat{h}(x)_y = \arg\max_y h'(x)_y\}$ will achieve high accuracy on this task (precisely, $\frac{1+\Delta(\widehat{h},h')}{2}$). By contrapositive, enforcing low discriminatory power means that the max discrepancy must also be small. It follows that for these methods $\textsc{Dis}^2$ should not be expected to hold universally, and in practice we see that this is the case (Figure 8.3). Nevertheless, when $\textsc{Dis}^2$ does overestimate accuracy, it does so by significantly less than prior methods.

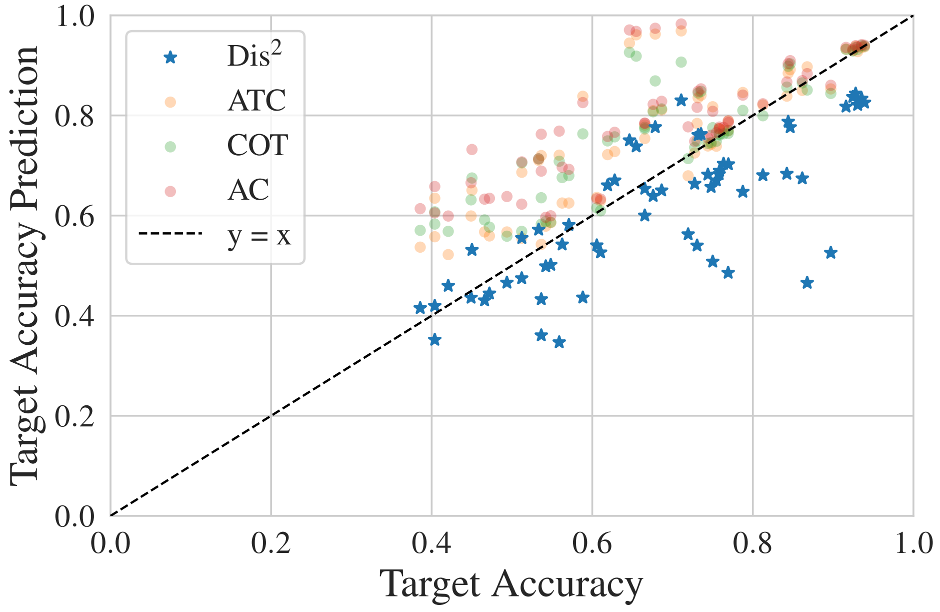


Figure 8.3: **DIS$^2$ may be invalid when the features are explicitly learned to violate Assumption 8.3.5.** Domain-adversarial representation learning algorithms such as DANN and CDAN indirectly minimize $\max_{h' \in \mathcal{H}} \Delta(\widehat{h}, h')$, meaning the necessary condition is less likely to be satisfied. Nevertheless, when DIS$^2$ does overestimate accuracy, it almost always does so by less than prior methods.

## 8.4 Efficiently Maximizing the Disagreement Discrepancy

For a classifier $\widehat{h}$, Theorem 8.3.6 clearly prescribes how to bound its test error: first, train a critic $h'$ on the chosen $\mathcal{X}$ to approximately maximize $\Delta(\widehat{h}, h')$, then evaluate $\epsilon_{\widehat{\mathcal{S}}}(\widehat{h})$ and $\widehat{\Delta}(\widehat{h}, h')$ using a holdout set. The remaining difficulty is in identifying the maximizing $h' \in \mathcal{H}$—that is, the one which minimizes $\epsilon_{\mathcal{S}}(\widehat{h}, h')$ and maximizes $\epsilon_{\mathcal{T}}(\widehat{h}, h')$. We can approximately minimize $\epsilon_{\mathcal{S}}(\widehat{h}, h')$ by minimizing the sample average of the convex surrogate $\ell_{\text{logistic}} := -\frac{1}{\log |\mathcal{Y}|} \log \operatorname{softmax}(h(x))_y$ as justified by statistical learning theory. However, it is less clear how to maximize $\epsilon_{\mathcal{T}}(\widehat{h}, h')$.

A few prior works suggest proxy losses for multiclass disagreement [Chuang et al., 2020, Pagliardini et al., 2023]. We observe that these losses are not theoretically justified, as they do not upper bound the 0-1 disagreement loss we hope to minimize and are non-convex (or even concave) in the model logits. Indeed, it is easy to identify simple settings in which minimizing these losses will result in a degenerate classifier with arbitrarily small loss but high agreement. Instead, we derive a new loss which satisfies the above desiderata and thus serves as a more principled approach to maximizing disagreement.

**Definition 8.4.1.** The *disagreement logistic loss* of a classifier $h$ on a labeled sample $(x, y)$ is defined as

$$\ell_{\text{dis}}(h, x, y) := \frac{1}{\log 2} \log\left(1 + \exp\left(h(x)_y - \frac{1}{|\mathcal{Y}| - 1} \sum_{\widehat{y} \neq y} h(x)_{\widehat{y}}\right)\right).$$

**Fact 8.4.2.** The disagreement logistic loss is convex in $h(x)$ and upper bounds the 0-1 disagreement loss (i.e., $\mathbf{1}\{\arg\max_{\widehat{y}} h(x)_{\widehat{y}} = y\}$). For binary classification, the disagreement logistic loss is equivalent to the logistic loss with the label flipped.

We expect that $\ell_{\text{dis}}$ can serve as a useful drop-in replacement for any future algorithm which requires maximizing disagreement in a principled manner. We combine $\ell_{\text{logistic}}$ and $\ell_{\text{dis}}$ to arrive at the empirical disagreement discrepancy objective:

$$\widehat{\mathcal{L}}_\Delta(h') := \frac{1}{|\widehat{\mathcal{S}}|} \sum_{x \in \widehat{\mathcal{S}}} \ell_{\text{logistic}}(h', x, \widehat{h}(x)) + \frac{1}{|\widehat{\mathcal{T}}|} \sum_{x \in \widehat{\mathcal{T}}} \ell_{\text{dis}}(h', x, \widehat{h}(x)).$$

By construction, $1 - \widehat{\mathcal{L}}_\Delta(h')$ is concave and bounds $\widehat{\Delta}(\widehat{h}, h')$ from below. However, note that the representations are already optimized for accuracy on $\mathcal{S}$, which suggests that predictions will have low entropy and that the $1/\log|\mathcal{Y}|$ scaling is unnecessary for balancing the two terms. We therefore drop the scaling factor, simply using standard cross-entropy; this often leads to higher discrepancy. In practice we optimize this objective with multiple initializations and hyperparameters and select the solution with the largest empirical discrepancy on a holdout set to ensure a conservative bound. Experimentally, we find that replacing $\ell_{\text{dis}}$ with either of the surrogate losses from Chuang et al. [2020], Pagliardini et al. [2023] results in smaller discrepancy; we present these results in Appendix G.2.

**Tightening the bound by optimizing over the logits.** Looking at Theorem 8.3.6, it is clear that the value of the bound will decrease as the capacity of the hypothesis class is restricted. Since the number of features is large, one may expect that Assumption 8.3.5 holds even for a reduced feature set. In particular, it is well documented that deep networks optimized with stochastic gradient descent learn representations with small effective rank, often not much more than the number of classes [Arora et al., 2018, 2019, Pezeshki et al., 2021, Huh et al., 2022]. This suggests that the logits themselves should contain most of the features' information about $\mathcal{S}$ and $\mathcal{T}$ and that using the full feature space is unnecessarily conservative.

To test this, we evaluate $\text{DIS}^2$ on the full features, the logits output by $\widehat{h}$, and various fractions of the top principal components (PCs) of the features. We observe that using logits indeed results in tighter error bounds *while still remaining valid*—in contrast, using fewer top PCs also results in smaller error bounds, but at some point they become invalid (Figure G.2). The bounds we report in this work are thus evaluated on the logits of $\widehat{h}$, except where we provide explicit comparisons in Section 8.5.

**Identifying the ideal number of PCs via a "validity score".** Even though reducing the feature dimensionality eventually results in an invalid bound, it is tempting to consider how we may identify approximately when this occurs, which could give a more accurate (though less conservative) prediction. We find that *the optimization trajectory itself* provides meaningful signal about this change. Specifically, Figure G.3 shows that for feature sets which are not overly restrictive, the critic very rapidly ascends to the maximum source agreement, then slowly begins overfitting. For much more restrictive feature sets (i.e., fewer PCs), the critic optimizes much more slowly, suggesting that we have reached the point where we are artificially restricting $\mathcal{H}$ and therefore underestimating the disagreement discrepancy. We design a "validity score" which captures this phenomenon, and we observe that it is roughly linearly correlated with the tightness of the eventual bound (Figure G.4). Though the score is by no means perfect, we can evaluate $\text{DIS}^2$ with successively fewer PCs and only retain those above a certain score threshold, reducing the average prediction error while remaining reasonably conservative (Figure G.5). For further details, see Appendix G.3.

## 8.5 Experiments

**Datasets.** We conduct experiments across 11 vision benchmark datasets for distribution shift on datasets that span applications in object classification, satellite imagery, and medicine. We use four BREEDs datasets: [Santurkar et al., 2020] Entity13, Entity30, Nonliving26, and Living17; FMoW [Christie et al., 2018] and Camelyon [Bandi et al., 2018] from WILDS [Koh et al., 2021]; Officehome [Venkateswara et al., 2017b]; Visda [Peng et al., 2018, 2017]; CIFAR10, CIFAR100 [Krizhevsky and Hinton, 2009]; and Domainet [Peng et al., 2019b]. Each of these datasets consists of multiple domains with different types of natural and synthetic shifts. We consider subpopulation shift and natural shifts induced due to differences in the data collection process of ImageNet, i.e., ImageNetv2 [Recht et al., 2019] and a combination of both. For CIFAR10 and CIFAR100 we evaluate natural shifts due to variations in replication studies [Recht et al., 2018b] and common corruptions [Hendrycks and Dietterich, 2019]. For all datasets, we use the same source and target domains commonly used in previous studies [Garg et al., 2023, Sagawa et al., 2021]. We provide precise details about the distribution shifts considered in Appendix G.1. Because distribution shifts vary widely in scope, prior evaluations which focus on only one specific type of shift (e.g., corruptions) often do not convey the full story. **We therefore emphasize the need for more comprehensive evaluations across many different types of shifts and training methods**, as we present here.

**Experimental setup and protocols.** Along with source-only training with ERM, we experiment with Unsupervised Domain Adaptation (UDA) methods that aim to improve target performance with unlabeled target data (FixMatch [Sohn et al., 2020], DANN [Ganin et al., 2016], CDAN [Long et al., 2018], and BN-adapt [Li et al., 2016]). We experiment with Densenet121 [Huang et al., 2017a] and Resnet18/Resnet50 [He et al., 2016] pretrained on ImageNet. For source-only ERM, as with other methods, we default to using strong augmentations: random horizontal flips, random crops, as well as Cutout [DeVries and Taylor, 2017] and RandAugment [Cubuk et al., 2020]. Unless otherwise specified, we default to full finetuning for source-only ERM and UDA methods. We use source hold-out performance to pick the best hyperparameters for the UDA methods, since we lack labeled validation data from the target distribution. For all of these methods, we fix the algorithm-specific hyperparameters to their original recommendations following the experimental protocol in [Garg et al., 2023]. For more details, see Appendix G.1.

**Methods evaluated.** We compare $\textsc{Dis}^2$ to four competitive baselines: *Average Confidence* (AC; [Guo et al., 2017]), *Difference of Confidences* (DoC; [Guillory et al., 2021]), *Average Thresholded Confidence* (ATC; [Garg et al., 2022]), and *Confidence Optimal Transport* (COT; [Lu et al., 2023]). We give detailed descriptions of these methods in Appendix G.1. For all methods, we implement post-hoc calibration on validation source data with temperature scaling [Guo et al., 2017], which has been shown to improve performance. For $\textsc{Dis}^2$, we report bounds evaluated both on the full features and on the logits of $\widehat{h}$ as described in Section 8.4. Unless specified otherwise, we set $\delta = .01$ everywhere. We also experiment with dropping the lower order concentration term in Theorem 8.3.6, using only the sample average. Though this is of course no longer a conservative bound, we find it is an excellent predictor of test error.

**Metrics for evaluation.** We report the standard prediction metric, *mean absolute error* (MAE). As our emphasis is on conservative error bounds, we also report the

| | MAE (↓) | | Coverage (↑) | | Overest. (↓) | |
|---|---|---|---|---|---|---|
| **DA?** | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ |
| Prediction Method | | | | | | |
| AC [Guo et al., 2017] | 0.1055 | 0.1077 | 0.1222 | 0.0167 | 0.1178 | 0.1089 |
| DoC [Guillory et al., 2021] | 0.1046 | 0.1091 | 0.1667 | 0.0167 | 0.1224 | 0.1104 |
| ATC NE [Garg et al., 2022] | 0.0670 | 0.0838 | 0.3000 | 0.1833 | 0.0842 | 0.0999 |
| COT [Lu et al., 2023] | 0.0689 | 0.0812 | 0.2556 | 0.1833 | 0.0851 | 0.0973 |
| DIS$^2$ (Features) | 0.2807 | 0.1918 | 1.0000 | 1.0000 | 0.0000 | 0.0000 |
| DIS$^2$ (Logits) | 0.1504 | 0.0935 | 0.9889 | 0.7500 | 0.0011 | 0.0534 |
| DIS$^2$ (Logits w/o $\delta$) | 0.0829 | 0.0639 | 0.7556 | 0.4167 | 0.0724 | 0.0888 |

Table 8.1: **Comparing the DIS$^2$ bound to prior methods for predicting accuracy.** DA denotes if the representations were learned via a domain-adversarial algorithm. In addition to mean absolute error (MAE), we report what fraction of predictions correctly bound the true error (Coverage), and the average prediction error among shifts whose accuracy is overestimated (Overest.). DIS$^2$ has reasonably competitive MAE but substantially higher coverage. By dropping the concentration term in Theorem 8.3.6 we can do even better, at some cost to coverage.

*coverage*, i.e. the fraction of predictions for which the true error does not exceed the predicted error. Finally, we measure the conditional average overestimation: this is the MAE among predictions which overestimate the accuracy.

**Results.** Reported metrics for all methods can be found in Table 8.1. We aggregate results over all datasets, shifts, and training methods—we stratify only by whether the training method is domain-adversarial, as this affects the validity of Assumption 8.3.5. We find that DIS$^2$ achieves competitive MAE while maintaining substantially higher coverage, even for domain-adversarial features. When it does overestimate accuracy, it does so by much less, implying that it is ideal for conservative estimation even when any given error bound is not technically satisfied. Dropping the concentration term performs even better (sometimes beating the baselines), at the cost of some coverage. This suggests that efforts to better estimate the true maximum discrepancy may yield even better predictors. We also show scatter plots to visualize performance on individual distribution shifts, plotting each source-target pair as a single point. For these too we report separately the results for domain-adversarial (Figure 8.3) and non-domain-adversarial methods

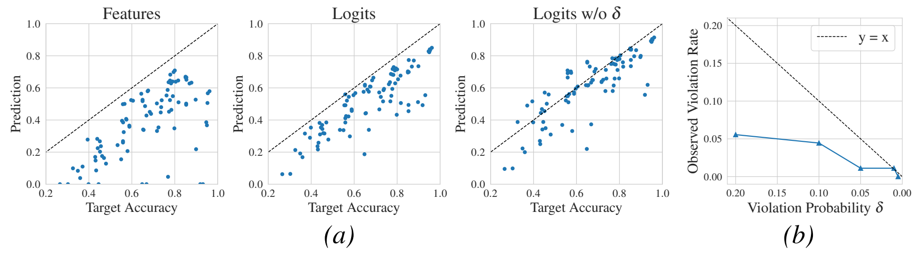


Figure 8.4: **(a):** Scatter plots depicting DIS$^2$ estimated bound vs. true error for a variety of shifts. "w/o $\delta$" indicates that the lower-order term of Theorem 8.3.6 has been dropped. **(b):** Observed bound violation rate vs. desired probability $\delta$. Observe that the true rate lies at or below $y = x$ across a range of values.

(Figure 8.1). To avoid clutter, these two plots do not include DoC, as it performed comparably to AC. Some of the BREEDS shifts contain as few as 68 test samples, which explains why accuracy is heavily underestimated for some shifts, as the lower order term in the bound dominates at this sample size.

Figure 8.4a displays additional scatter plots which allow for a direct comparison of the variants of DIS$^2$. Finally, Figure 8.4b plots the observed violation rate (i.e. $1-$coverage) of DIS$^2$ on non-domain-adversarial methods for varying $\delta$. We observe that it lies at or below the line $y = x$, meaning the probabilistic bound provided by Theorem 8.3.6 holds across a range of failure probabilities. Thus we see that our probabilistic bound is empirically valid all of the time—not in the sense that each individual shift's error is upper bounded, but rather that the desired violation rate is always satisfied.

**Strengthening the baselines to improve coverage.** Since the baselines we consider in this work prioritize predictive accuracy over conservative estimates, their coverage can possibly be improved without too much increase in error. We explore this option using LOOCV: for a desired coverage, we learn a parameter to either scale or shift a method's prediction to achieve that level of coverage on all but one of the datasets. We then evaluate the method on all shifts of the remaining dataset, and we repeat this for each dataset. Appendix G.4 reports the results for varying coverage levels. We find that (i) the baselines do not achieve the desired coverage on the held out data, though they get somewhat close; and (ii) the adjustment causes them to suffer higher MAE than DIS$^2$. Thus DIS$^2$ is on the Pareto frontier of MAE and coverage, and is preferable when conservative bounds are desirable. We believe identifying alternative methods of post-hoc prediction adjustment is a promising future direction.

## 8.6 Conclusion

The ability to evaluate *trustworthy*, non-vacuous error bounds for deep neural networks under distribution shift remains an extremely important open problem. Due to the wide variety of real-world shifts and the complexity of modern data, restrictive a priori assumptions on the distribution (i.e., before observing any data from the shift of interest) seem unlikely to be fruitful. On the other hand, prior methods which estimate accuracy using extra information—such as unlabeled test samples—often rely on opaque conditions whose likelihood of being satisfied is difficult to predict, and so they sometimes provide large overestimates of test accuracy with no warning signs.

This work attempts to bridge this gap with a simple, intuitive condition and a new disagreement loss which together result in competitive error *prediction*, while simultaneously providing an (almost) provable probabilistic error *bound*. We also study how the process of evaluating the bound (e.g., the optimization landscape) can provide even more useful signal, enabling better predictive accuracy. We expect there is potential to push further in each of these directions, hopefully extending the current accuracy-reliability Pareto frontier for test error bounds under distribution shift.

# Chapter 9

# One-Shot Strategic Classification Under Unknown Costs

This chapter is based on Rosenfeld and Rosenfeld [2024]:
Rosenfeld, E. & Rosenfeld, N.
One-Shot Strategic Classification Under Unknown Costs.
*(In Submission) arXiv preprint arXiv:2311.02761*, 2023.

## 9.1 Introduction

Across a multitude of domains, machine learning is increasingly being used to inform decisions about human users. But when users stand to gain from certain predictive outcomes, they may act to obtain favorable predictions by modifying their features. Since this can harm predictive performance, learning becomes susceptible to *Goodhart's law*, which states that "when a measure becomes a target, it ceases to be a good measure" [Goodhart, 1975]. This natural tension between learning systems and their users applies broadly: loan approvals, admissions, hiring, insurance, and welfare benefits are all examples in which the system seeks predictions that are accurate, whereas users—irrespective of their true label—wish to be classified as positive.

The field of *strategic classification* [Brückner et al., 2012, Hardt et al., 2016] studies learning in such settings, with the principal aim of learning classifiers that are robust to strategic user behavior. However, most works in this field rely on the key assumption that the learner knows precisely how users would respond to any given classifier. This is typically modeled as knowledge of the underlying *cost function* $c(x, x')$ which defines the cost users incur for modifying features $x$ to become $x'$. This assumption enables tractable learning, but it is unrealistic and it effectively takes the "sting" out of Goodhart's law: in a statistical sense, the power to anticipate user responses completely nullifies the effect of users' gaming behavior on predictive performance (for radial basis cost functions; see Appendix H.7).

Indeed, if we survey some well-known examples of Goodhart's law [e.g. Chrystal et al., 2003, Elton, 2004, Fire and Guestrin, 2019, Teney et al., 2020], it becomes apparent that much of the policy challenge arises precisely from *not* knowing how users would respond. This motivates us to instead focus on learning strategically robust classifiers under an *unknown* cost function. To cope with this uncertainty, we take a conservative approach and model the system as aiming to learn a classifier that is robust to all cost functions $c$ in some uncertainty set $\mathcal{C}$, which we think of as including all cost functions which are believed to be plausible (alternatively, it is a set which we can be confident will contain the true, unknown $c$). Our approach is therefore doubly robust, providing guarantees under both the manipulation of inputs by strategic behavior and an adversarial choice of cost function. We argue this is necessary: we prove that if one optimizes for strategic classification under a single, fixed cost, then any discrepancy between that cost and the true cost can result in dramatically reduced accuracy.

While some works have studied strategic learning under unknown responses [Dong et al., 2018, Ahmadi et al., 2021, Shao et al., 2023, Lechner et al., 2023, Harris et al., 2023], their focus is entirely on sequential learning settings. This allows them to cope with response uncertainty via exploration: deploying a series of models over time, observing how users respond, and adapting accordingly. Algorithms for such online settings are designed to ensure that regret decays sufficiently fast with the number of rounds—but they provide no guarantees on worst-case outcomes for any single deployment. We observe that there are many realistic settings in which multiple deployments are too costly or technically impossible (e.g. financial regulation); in which arbitrary exploration is unrealistic or unethical (e.g. testing in education); in which there is need for immediately beneficial short-term outcomes (e.g. epidemic vaccination); or in which even a single bad round could be very harmful (e.g. environmental conservation).

Motivated by such examples, this work studies robust strategic learning in a "one-shot" setting where the learner must commit to one single model at deployment. In this setting, we recast our objective as optimizing for a classifier that minimizes the worst-case strategic risk over an uncertainty set $\mathcal{C}$. We devise two efficient algorithms, for the full-batch and stochastic settings, which provably converge to the minimax solution at the rate $\tilde{\mathcal{O}}(T^{-1/2})$. Notably, this rate is dimension independent for the typical cost functions—including $\ell_1$ and $\ell_2$ norms—and it is achieved despite the inner maximization being non-concave. A key step in our approach is to adapt the *strategic hinge loss* [Levanon and Rosenfeld, 2022] to properly handle a large set of possible costs beyond the $\ell_2$ norm. As the strategic hinge loss is non-convex, we apply a regularization term that accounts for the unique structure of strategic response: our analysis uncovers that the “correct” form of regularization is in fact the *dual norm of $\beta$ with respect to the cost*. We complement this with an updated generalization bound for this loss which corrects a previous error.

More broadly, our approach relies on the observation that strategic responses induce a shift in the data distribution [Perdomo et al., 2020]. Because different costs induce different shifts, uncertainty over cost functions translates to uncertainty over test distributions. This allows us to formulate our problem as one of *distributionally robust optimization* (DRO) [Namkoong and Duchi, 2016, Duchi and Namkoong, 2021], where the goal is to predict well on the worst-case distribution within some given set. Whereas the typical DRO formulation defines the uncertainty set with respect to a more traditional probability divergence, in our case the set of distributions is inherited from the structure of strategic responses and includes all shifts that can result from strategic behavior under any $c \in \mathcal{C}$. This means that the entire *set* of possible distributions becomes dependent on the classifier, which is one of the primary challenges our work addresses.

### 9.1.1 Related Work

**Strategic classification.** Introduced in Hardt et al. [2016], and based on earlier works [Brückner and Scheffer, 2009, Brückner et al., 2012, Großhans et al., 2013], the literature on strategic classification has since been growing steadily. Focusing on supervised classification in the batch (offline) setting, here we list a relevant subset. Advances have been made on both statistical [Zhang and Conitzer, 2021, Sundaram et al., 2021] and algorithmic aspects of learning [Levanon and Rosenfeld, 2021], but the latter lacks guarantees. In contrast, our work provides efficient algorithms that are provably correct. Efforts have also been made to extend learning beyond the basic setting of Hardt et al. [2016]. These include: accounting for users with noisy estimates [Jagadeesan et al., 2021], missing information [Ghalme et al., 2021, Bechavod et al., 2022], more general preferences [Sundaram et al., 2021, Levanon and Rosenfeld, 2022, Eilat et al., 2023]; incorporating causal elements into learning [Miller et al., 2020b, Chen et al., 2023b, Horowitz and Rosenfeld, 2023, Mendler-Dünner et al., 2022], and considering societal implications [Milli et al., 2019, Levanon and Rosenfeld, 2021, Lechner and Urner, 2022].

**(Initially) unknown user responses.** Several works have considered the case of inferring unknown user responses under different forms of strategic learning, but in online or sequential settings. Dong et al. [2018] bound the Stackelberg regret of online learning when both costs and features are adversarial, but when only negatively-labeled users respond; Harris et al. [2023] bound a stronger form of regret when responses only come from positively-labeled users. Ahmadi et al. [2021] propose a strategic variant of the perceptron and provide mistake bounds for $\ell_1$ norm with unknown diagonal scaling or $\ell_2$ norm multiplied by an unknown constant. Shao et al. [2023] study learning under unknown ball manipulations with personalized radii, and give mistake bounds and (interactive) sample complexity bounds for different informational structures. Lechner et al. [2023] bound the sample and iteration complexity of learning under general, non-best response manipulation sets via repeated model deployments to infer the manipulation graph. Lin and Zrnic [2023] learn under a misspecified response model, inferred in a (nonadaptive) exploration phase from some class of models. The analyses in these works better reflect reality in that the exact response cannot be known. But online deployment and long-term regret minimization are not appropriate for many natural use cases of strategic classification, which motivates our investigation of the one-shot setting.

## 9.2 Preliminaries

**Notation.** We study the problem of linear strategic classification on inputs $x \in \mathcal{X} \subseteq \mathbb{R}^d$ and labels $y \in \mathcal{Y} := \{\pm 1\}$ where the exact response of the test population is unknown. We consider linear classifiers $\operatorname{sign}(\beta^\top x)$, optimized over some bounded set $\mathcal{B} \subset \mathbb{R}^{d+1}$ (this includes a bias term which we leave implicit and which is not included in the vector norm). To model users' responses, we consider the typical setup of a *cost function* $c(x, x')$ which defines the cost for an agent to change their features from $x$ to $x'$. Together with a utility $u \geq 0$ gained from a positive classification, this cost determines the strategic response of a rational agent to a classifier $\beta$ via

$$x(\beta) := \arg\max_{x'} \left[ \mathbf{1}\{\beta^\top x' \geq 0\} \cdot u - c(x, x') \right]. \tag{9.1}$$

Thus, an agent will move only if it would change their classification from negative to positive, and only if $c(x, x') < u$. We handle non-uniqueness of the argmax by breaking ties arbitrarily, but we follow convention by assuming no strategic response if the net utility is exactly 0. We let $\delta(x; \beta) := x(\beta) - x$ denote the movement of an agent, and we suppress dependence of $\delta$ on $x, \beta$ where clear from context.

**Form of the cost function.** We study cost functions that can be written as $c(x, x') = \phi(\|x' - x\|)$ for a norm $\|\cdot\|$ and non-decreasing function $\phi : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. This includes the $\ell_2$-norm (by far the most common cost, sometimes squared), but it is also much more general: it allows for different non-linear transformations which may better reflect real-world costs, such as $\phi(r) = \ln(1 + r)$, and we allow $\|\cdot\|$ to denote *any* differentiable and monotonic norm, which includes all $p$-norms with $p \in [1, \infty]$. This significantly expands upon the costs discussed in the literature thus far.

**Parameterizing the space of costs.** Recall that we are interested in robust prediction when we cannot know the exact strategic response. Keeping the assumption of rational behavior, this naturally points to an unknown cost function as a primary source of this uncertainty. We parameterize this uncertainty via an unknown positive definite (PD) matrix $\Sigma \succ \mathbf{0}$ in $\mathbb{R}^{d \times d}$ which scales the relative costs per input dimension, denoting the induced norm as $\| \cdot \|_\Sigma$. For the 2-norm, this is the standard PD norm given by $\|x\|_\Sigma := \sqrt{x^\top \Sigma x}$, but we define it for general norms as $\|x\|_\Sigma := \|\Sigma^{1/2} x\|$. Hence, for our purposes, any cost function $c$ is uniquely determined by its parameterization $\Sigma$; we will use these two variables interchangeably. The other two factors in strategic response are the positive utility $u$ and the monotone transform $\phi$: for reasons that will soon be made clear, we only require knowledge of the maximum value $u_*$ such that $\phi(u_*) \leq u$ (this is clearly satisfied for known $\phi$ and $u$, as is typically assumed). Note this value need not be shared among users, and it suffices to know an interval in which it lies—but for simplicity we treat it as fixed.

**Encoding uncertainty.** Since the true cost is not known, and since we cannot estimate it in an online fashion, we instead assume a system-specified *uncertainty set* $\mathcal{C}$, defined as a compact, convex set of possible costs $c$ which is expected to contain the true cost. The goal of our analysis will be to derive strategies for efficiently identifying a classifier which ensures optimal (and boundable) worst-case test performance over *all* costs $c \in \mathcal{C}$, and therefore also bounded error under the true cost. Notably, this also means that *even if the true cost changes over time*, our error bound will hold so long as the cost remains within $\mathcal{C}$. In practice we want $\mathcal{C}$ to be broad enough that we can be confident it contains the test-time cost—but we will also see that our convergence guarantees scale inversely with the diameter of this set, so it should be selected to be no larger than necessary.

### 9.2.1 Strategic Learning Under a Single, Known Cost

As a first step towards learning robustly under all costs in $\mathcal{C}$, it will be useful to first consider learning under a single fixed cost. For a *known* cost $c$, the typical goal would be to minimize the 0-1 loss under strategic response with this cost: $\ell^c_{0-1}(\beta^\top x, y) := \mathbf{1}\{\text{sign}(\beta^\top (x + \delta)) \neq y\}$. It is common to instead consider a more easily optimized surrogate such as the hinge loss $\ell_{\text{hinge}}(\beta^\top x, y) := \max\{0, 1 - y\beta^\top (x + \delta)\}$, but the discontinuous nature of $\delta(x; \beta)$ w.r.t. $\beta$ means that optimization is still intractable. Instead, we make use of the recently proposed *strategic hinge loss* [Levanon and Rosenfeld, 2022] which augments the standard hinge loss with an additional term to account for this discontinuity: $\ell_{\text{s-hinge}}(\beta^\top x, y) := \max\{0, 1 - y(\beta^\top x + 2\|\beta\|_2)\}$ (note this does not explicitly include $\delta$).

Unfortunately, even this relaxation poses difficulties. Firstly, the objective is non-convex—though Levanon and Rosenfeld [2022] show that for known costs it often learns reasonable classifiers in practice, once we transition to unknown costs it will become clear that having a guaranteed sub-optimality bound is important (such a bound is impossible in general for non-convex objectives). Second, the additional term $2\|\beta\|_2$ captures the effect of strategic response only under the standard $\ell_2$-norm cost.[1]

**Cost-aware strategic hinge.** For our setting, we derive a more general strategic hinge loss that applies to the broader class of costs. This loss admits a natural form which relies on the *dual norm* of $\beta$ with respect to the cost function:

**Definition 9.2.1.** The $\Sigma$*-transformed dual norm* of $\beta$ is denoted $\|\beta\|_{*,\Sigma^{-1}} := \sup_{\|v\|_\Sigma = 1} \beta^\top v$ and is equal to $\|\Sigma^{-1/2}\beta\|_*$. We may leave dependence on $\Sigma$ implicit, writing simply $\|\beta\|_*$.

**Definition 9.2.2.** The *cost-dependent strategic hinge loss* is:

$$\ell^c_{\text{s-hinge}}(\beta^\top x, y) := \max\{0, 1 - y(\beta^\top x + u_* \|\beta\|_*)\}. \tag{9.2}$$

Note our proposed $\ell^c_{\text{s-hinge}}$ generalizes the previous $\ell_{\text{s-hinge}}$ since the $\ell_2$-norm is its own dual. The appearance of the dual norm here is not by chance. This quantity captures the dimension-wise sensitivity of our decision rule to changes in $x$, scaled *inversely proportionally* to the cost an agent incurs for moving in that dimension. This should be intuitive: for any given direction, the less it costs a user to modify their features, the more they can afford to move, and thus the greater the importance of reducing our classifier's sensitivity to it. We make this formal with the following lemma which bounds the maximal strategic change to inputs.

**Lemma 9.2.3.** *Fix $\beta$ and let $c(x, x') = \phi(\|x - x'\|_\Sigma)$. The maximum possible change to a user's score due to strategic behavior is $u_* \|\beta\|_*$.*

Proofs of all lemmas can be found in Appendix H.3. Thus we see how augmenting the hinge loss with the dual norm serves as a natural approach to robust strategic classification, and we use this loss throughout. More generally, it will also be useful to define the *k-shifted strategic hinge loss* as $\ell_{\text{s-hinge}}(\beta; k) := \max\{0, 1 - y(\beta^\top x + k)\}$.

Though $\ell^c_{\text{s-hinge}}$ allows for more general costs, it remains non-convex—we will return to this point in Section 9.4.2. Also note that the "correct" transformation to use

[1]While Levanon and Rosenfeld [2022] do discuss more general costs, they give only a generic description.

depends on the true cost. In our case this is unknown, but we show that it suffices for the dual norm to be bounded by a constant $B$: $\forall c \in \mathcal{C},\ \|\beta\|_* \leq B$. We also let $X \in \mathbb{R}$ denote the maximum of $\|x\|, \|x\|_2$ over the training examples. Finally, we denote by $L$ the Lipschitz constant of the loss gradient, which appears in the convergence rates of the algorithms we derive. This can be generically bounded as $L \leq X + u_* L_*$, where $L_*$ is the Lipschitz constant of the dual norm. For the common setting where $\|\cdot\|$ is a $p$-norm, we have $L_* = \max\left(1, d^{1/2 - 1/p}\right)$. Notably, this quantity is *independent of the dimension* $d$ for $p \leq 2$ and scales no worse than $\sqrt{d}$ otherwise.

**Risk and generalization.** Denote the overall strategic hinge risk by $R^c_{\text{s-hinge}}(\beta) := \mathbb{E}[\ell^c_{\text{s-hinge}}(\beta^\top x, y)]$, with the strategic 0-1 risk defined analogously as $R^c_{0-1}$. By design, the former upper bounds the latter:

**Lemma 9.2.4.** *For any cost $c \in \mathcal{C}$, $R^c_{0-1}(\beta) \leq R^c_{\textit{s-hinge}}(\beta)$.*

Thus, our cost-dependent strategic hinge loss is an effective proxy for the 0-1 loss. We next establish its generalization.

**Theorem 9.2.5.** *With probability $\geq 1 - \delta$, for all $\beta \in \mathcal{B}$ and all cost functions $c \in \mathcal{C}$,*

$$R^c_{0-1}(\beta) \leq \widehat{R}^c_{\textit{s-hinge}}(\beta) + \frac{B(4X + u_*) + 3\sqrt{\ln 1/\delta}}{\sqrt{n}},$$

*where $\widehat{R}$ is the empirical risk over a training set of size $n$.*

This result extends (and fixes an error in) the bound for $\ell_2$-norm cost from Levanon and Rosenfeld [2022]. The proof, found in Appendix H.4, applies standard Rademacher bounds by decomposing the strategic hinge loss and bounding the terms separately while accounting for general strategic responses. The fact that this bound holds uniformly for *all* cost functions is critical, as it allows us to apply it to the worst case cost even when that cost is unknown.

## 9.3 The Perils of Using a Wrong Cost Function

As the agents' movement depends intricately on the precise cost function, correctly anticipating strategic response requires a good estimate of that cost. In the one-shot setting, this is further complicated by the fact that we have no access to a mechanism by which to *infer* the cost (e.g., via online interaction): we must pick a single classifier and commit to it, without knowing the true cost function a priori. A natural approach would be to make use of existing machinery for single-cost strategic learning, as described above, using some reasonable choice for the cost. For example, one idea would be to simply pick a cost which we believe is reasonably "close" to the true cost, in the hope that predictive performance degrades gracefully with our error. Certainly, this is better than blindly proceeding with the default $\ell_2$-norm. Unfortunately we find that the task of cost-robust strategic learning is much more difficult (learning theoretically) than is first apparent—it turns out that without more explicit assumptions on our guess's distance to the true cost *and* on the data distribution itself, providing any sort of robustness guarantee is impossible.

**Hardness results.** Our first result proves that if we must to commit to a single fixed cost, unless that cost is exactly correct, minimizing the empirical risk can never provide a non-trivial data-independent guarantee:

**Theorem 9.3.1.** *Consider the task of learning a norm-bounded binary linear classifier. Fix any two costs $c_1, c_2$ with non-equal cost matrices, and let $0 \leq \epsilon \leq \frac{1}{2}$. There exists a distribution $q$ over $\mathcal{X} \times \mathcal{Y}$ such that:*

1. *For each of $c_1$ and $c_2$, there is a (different) classifier which achieves 0 error on $q$ when facing strategic response under that cost; and*
2. *Any classifier which achieves 0 error on $q$ under $c_1$ suffers error $\epsilon$ under $c_2$, and any classifier which achieves 0 error on $q$ under $c_2$ suffers error $1 - \epsilon$ under $c_2$.*

Theorem 9.3.1 says that if there is *any error at all* in estimating $\Sigma$, then any classifier which achieves perfect accuracy under our assumed cost might do no better than random (or even worse than random) when deployed. The proof (see Appendix H.1) constructs a worst-case distribution that is chosen adversarially with respect to the error in cost estimation. This construction is quite robust in that there exists an entire space of such solutions and the lower bound decays smoothly for distributions which are close to the one we define (especially for larger error in the estimate of $\Sigma$). However, there is hope that perhaps for non-adversarial distributions, estimating a fixed cost may be reasonable. To investigate this possibility, we also study the more natural setting of isotropic Gaussian $q(x \mid y)$. We find once again that, even in the limit of infinite data, guessing the wrong cost can be quite harmful:

**Theorem 9.3.2.** *Define the distribution $q$ over $\mathcal{X} \times \mathcal{Y}$ as $q(y) \overset{d}{=} \text{Unif}(\{\pm 1\})$, $q(x \mid y) \overset{d}{=} \mathcal{N}(y \cdot \mu_0, \sigma^2 I)$. Denote by $\Phi$ the standard Normal CDF. Let the true cost be defined as $\|x - x'\|_\Sigma$ with unknown cost matrix $\Sigma$, and let $\beta^*$ be the classifier which minimizes the strategic 0-1 risk under this cost.*

*Suppose one instead learns a classifier $\widehat{\beta}$ by assuming an incorrect cost $\widehat{\Sigma}$ and minimizing the population strategic 0-1 risk under that cost: $\widehat{\beta} := \arg\min_\beta \mathbb{E}_q[\ell_{0-1}^{\widehat{c}}(\beta)]$. Then the excess 0-1 risk suffered by $\widehat{\beta}$ is*

$$\Phi\left(\frac{\|\mu_0\|}{\sigma}\right) - \frac{1}{2}\left(\Phi\left(\frac{\|\mu_0\| - \epsilon}{\sigma}\right) + \Phi\left(\frac{\|\mu_0\| + \epsilon}{\sigma}\right)\right),$$

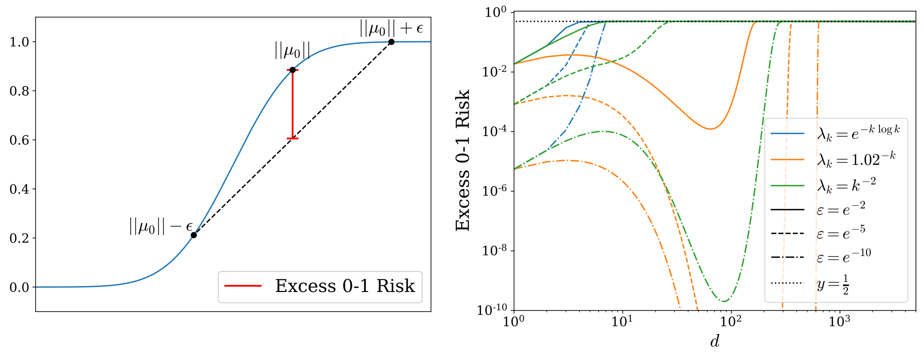


Figure 9.1: **Visualizing the excess 0-1 risk from Theorem 9.3.2. Left:** A toy illustration of the excess risk as a function of $\|\mu_0\|$ and $\epsilon$ and where they lie on the Gaussian CDF. **Right:** Excess risk curves as a function of the dimension $d$, where $\mu_0 = \frac{1}{\sqrt{d}}\mathbf{1}$ and $\sigma^2 = {}^1\!/_d$. Color indicates the function which generates the spectrum of $\Sigma$, where $\lambda_k$ is the $k$-largest eigenvalue in the series. Line style indicates the error $\varepsilon$ in estimating the eigenvalue in each dimension—precisely, we define $\widehat{\Sigma} = (1-\varepsilon)\Sigma$. Even with all eigenvalues estimated to error $1 - e^{-10}$, excess risk grows rapidly with $d$ towards $\frac{1}{2}$.

*where* $\epsilon = \epsilon(\Sigma, \widehat{\Sigma}, \mu_0) := \frac{u_* | \|\mu_0\|_{*,\widehat{\Sigma}^{-1}} - \|\mu_0\|_{*,\Sigma^{-1}} |}{\|\mu_0\|}$ *is the normalized estimation error of the* $\Sigma$*-transformed dual norm of* $\mu_0$.

Theorem 9.3.2 demonstrates that even for *much* more benign distributions, choosing a classifier based on a slightly incorrect cost can be very non-robust. Roughly, we should expect $\epsilon = \Theta(\text{Trace}(\widehat{\Sigma}^{-1} - \Sigma^{-1}))$, scaling as our error grows along directions where $\mu_0$ is large. Crucially, this is with respect to the *inverse* cost matrix $\Sigma^{-1}$, which means that a tiny estimation error in the lower end of the eigen-spectrum can have a large effect. For example, imagine that the classes are sufficiently separated so that the optimal classifier *absent* strategic behavior can get accuracy close to 1 (which is trivial in high dimensions). Then under strategic behavior, the error in guessing the smallest eigenvalue of $\Sigma$ need only be $\Omega(\|\mu_0\| \lambda_{\min}(\Sigma) \lambda_{\min}(\widehat{\Sigma}))$ to induce excess risk of approximately ${}^1\!/_4$, and it will rapidly approach ${}^1\!/_2$ as $d$ grows. To give a bit more intuition for these factors, Figure 9.1 visualizes a few simple examples of how the excess risk behaves as dimension increases, depending on the distribution of eigenvalues of $\Sigma$ and our error in estimating them. More abstractly, we can see that if we only slightly err in estimating the cost of movement in a direction, it could cause a very large error in estimating how much agents will move in that direction.

**Intentionally conservative guesses can still fail.** It is not immediately obvious why guessing a single cost should always have poor worst-case performance. For example, it might appear that we could just select a cost function which *under*estimates the cost to move in any given direction, and therefore overestimates the movement of any given point. In being intentionally conservative, this choice seems like it should be reasonably robust to misspecification, even if not optimally so. However, this ignores the key fact that in strategic classification there is also the potential for *beneficial* strategic response: a point with true label $y = 1$ but incorrect prediction $\widehat{y} = -1$ has the potential to correct this error—to the learner's benefit—by changing its features.

The above results show that it is not enough in strategic classification to be aware of the fact that users will move in response to the classifier, nor to have an educated guess for how they will move: it is essential to account for *uncertainty* in how they will move by anticipating the ways in which the true, unknown cost function may differ from what we expect. Our lower bounds thus motivate robustly learning a classifier based not on a single cost, but on a *set* in which the true cost is expected to lie—this reduces potential misspecification of the response model [Lin and Zrnic, 2023] while remaining applicable in the one-shot setting.

## 9.4 Maximizing Risk for a Fixed Classifier and Minimizing Risk for a Fixed Cost

To identify a robust classifier when the true cost is unknown, we will optimize the worst-case strategic risk with respect to the learner-specified uncertainty set $\mathcal{C}$. Our objective is therefore to solve the min-max problem

$$\min_{\beta} \max_{c \in \mathcal{C}} R^{c}_{\text{s-hinge}}(\beta). \tag{9.3}$$

As we noted in the introduction, if we consider the cost function to be inducing a classifier-specific distribution, then Equation (9.3) becomes an instance of *distributionally robust optimization*. A typical approach to this type of problem would be to use a gradient-based method which converges to a Nash equilibrium between the learner and an adversary (who "chooses" the cost). To apply this in our setting, we must recast Equation (9.3) as two separate objectives: one for the learner minimizing $\beta$ and the other for the adversary maximizing $c$. However, the idiosyncrasies of the strategic learning setting give rise to unique challenges which must be addressed for this approach to be successful.

**Algorithm 6** Pseudocode for MaxLossCost

```
input: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, Classifier $\beta$, Cost uncertainty set $\mathcal{C}$, Upper bound $u_*$.
define: $\|\beta\|_*^{\min} := \min_{\Sigma \in \mathcal{C}} \|\beta\|_*$, $\|\beta\|_*^{\max} := \max_{\Sigma \in \mathcal{C}} \|\beta\|_*$
initialize: $k \leftarrow \|\beta\|_*^{\min}$, $r \leftarrow \widehat{R}_{\text{s-hinge}}(\beta; u_* k)$
  Sort training points in increasing order by $v_i := y_i(1 - y_i \beta^\top x_i)$.
  Set $j$ as first index where $v_j - u_* k > 0$.
  for entry $v_i$ in sorted list, starting from $j$ do
    Set $k = {}^{v_i}/_{u_*}$.
    If $k > \|\beta\|_*^{\max}$, break.
    Update new risk $r = \widehat{R}_{\text{s-hinge}}(\beta; u_* k)$. Maintain maximum $r_{\max}$ and $k_{\max}$ which induced it.
  end for
  If $\widehat{R}_{\text{s-hinge}}(\beta; u_* \|\beta\|_*^{\max}) > r_{\max}$, return $\arg\max_{\Sigma \in \mathcal{C}} \|\beta\|_*$
  Otherwise, return $\Sigma \in \mathcal{C}$ such that $\|\beta\|_{*, \Sigma^{-1}} = k_{\max}$.
```

### 9.4.1 Solving the Inner Max for a Fixed Classifier

The first step to solving Equation (9.3) is determining a way to solve the inner maximization problem, which will serve as an important subroutine for our eventual algorithm which solves the full min-max objective. However, the max objective is non-concave. We thus begin with an efficient algorithm to address this. Algorithm 6 gives pseudocode[2] for MaxLossCost which, for a given classifier $\beta$, maximizes the strategic hinge loss over costs $c \in \mathcal{C}$. Due to Equation (9.2), this amounts to finding the maximizing $k$-shift for $k = u_* \|\beta\|_*$ where $\|\beta\|_*$ is achievable by some $c \in \mathcal{C}$. We next prove correctness and runtime:

**Lemma 9.4.1.** *For any classifier $\beta$, dataset $(\mathcal{X} \times \mathcal{Y})^n$, and uncertainty set $\mathcal{C}$, MaxLossCost runs in $\mathcal{O}(nd + n \ln n)$ time and returns a cost function $c \in \mathcal{C}$ which maximizes the cost-dependent strategic hinge risk $\widehat{R}^c_{\textit{s-hinge}}(\beta)$.*

MaxLossCost takes advantage of the fact that the loss is piecewise linear in the *scalar value* $\|\beta\|_*$, carefully constructing a list of the training samples which are sorted according to a particular intermediate value and traversing this list while updating the risk at the boundary of each linear component. For a full description of the algorithm see Appendix H.5.

[2]The code calls for identifying $\|\beta\|_*^{\min}$ and $\|\beta\|_*^{\max}$. The specific way we parameterize $\mathcal{C}$ makes this simple (see Section 9.5.1).

**Bounding the adversarial risk of any classifier.** Independent of our main objective, MaxLossCost already allows us to derive adversarial strategic risk bounds for any classifier: simply evaluate its worst-case empirical strategic hinge loss and then apply Theorem 9.2.5. In principle, this means we could *try* minimizing the strategic hinge loss for a single cost and we would technically be able to derive an error bound for the solution. However, this approach is not ideal for two reasons: first, as noted earlier, the strategic hinge loss is non-convex in $\beta$, so direct optimization may be unsuccessful. Second, as shown in Section 9.3 there is little reason to believe that optimizing an objective which does not account for the min-max structure will achieve good adversarial risk, making the generalization bound correct but usually unhelpful (e.g., trivial). Still, it is useful to be able to give a valid bound on the adversarial risk of whatever classifier we may hope to evaluate.

### 9.4.2 Solving the Outer Min for a Fixed Cost

We now switch to the task of finding a classifier which minimizes worst-case risk. Note Equation (9.3) poses two key challenges: (i) the strategic hinge loss is non-convex, and (ii) optimization is further complicated by the inner max operator. Hence, we begin with the case of a single fixed cost, which will serve us as an intermediary step towards optimizing over the set of all costs. Even for a single cost, the non-convexity of the strategic hinge means that we would only be able to guarantee convergence to a stationary point. We address this via regularization.

**Convexification via regularization.** Regularization is a standard means to introduce (strong) convexity: for the hinge loss, applying (squared) $\ell_2$ regularization makes the objective strongly convex, improving optimization while simultaneously preventing overfitting. Since we would want to regularize our objective anyways, one possible solution would be to add just enough $\ell_2$ regularization to convexify it. While sound in principle, this is inappropriate for our setting because it does not account for the cost-specific nature of strategic response, so the amount of regularization needed would be extreme. Specifically, we prove a lower bound on the required $\ell_2$ regularization to ensure convexity of the strategic hinge loss:

**Theorem 9.4.2.** *Let $\|\cdot\|$ be a $p$-norm and fix a cost matrix $\Sigma$. For any distribution $q$ on $\mathcal{X} \times \mathcal{Y}$, let $\tau^+ := q(y = 1)$. Then the $\ell_2$-regularized loss $R^c_{\text{s-hinge}}(\beta) + \lambda u_* \|\beta\|_2$ is non-convex unless $\lambda \geq \tau^+ \|\Sigma^{-1/2}\|_2$.*

Thus, the necessary $\ell_2$ regularization would be cost-dependent and quite large, scaling with the inverse of the smallest eigenvalue. Moreover, since we want to optimize this risk over all possible costs, this means that it would need to scale with the largest spectral norm among *all* inverse cost matrices in $\mathcal{C}$.

**Dual norm regularization.** Luckily, we observe that for a given cost $c$ we can instead apply a small amount of *dual norm* regularization to $\beta$. This naturally accounts for the problem's structure and eliminates the need for excessive penalization. Our dual-regularized objective is $R^c_{\text{s-hinge}}(\beta) + \lambda u_* \|\beta\|_*$, where $\lambda$ determines the regularization strength. Since the dual norm $\|\beta\|_*$ is implicitly defined via the cost matrix, the following is straightforward:

**Proposition 9.4.3.** *For any norm $\|\cdot\|$, cost matrix $\Sigma$, and distribution $q$, the dual-regularized loss $R^c_{\textit{s-hinge}}(\beta) + \lambda u_* \|\beta\|_*$ is guaranteed to be convex for all $\lambda \geq \tau^+$.*

Proofs are in Appendix H.2. Proposition 9.4.3 shows that a small, cost-independent amount of dual norm regularization ensures convexity of the outer min problem for *any* cost function, making it the natural choice in this setting.

## 9.5 Efficiently Identifying the Minimax-Optimal Classifier

We have shown how to efficiently solve for the maximizing cost $c \in \mathcal{C}$ for any classifier $\beta$, allowing us to apply the corrected generalization bound in Theorem 9.2.5 and get an upper bound on the adversarial 0-1 strategic risk. We have also seen that with dual norm regularization, optimizing the strategic hinge loss becomes tractable and also accounts for the unique structure of strategic response. The remaining challenge is to combine these two methods to find the overall solution to the min-max objective.

**Solving a different loss for each cost simultaneously.** An interesting consequence of using the dual norm is that for any fixed cost, the "correct" regularization is a function of that cost. Since we want to minimize this objective with respect to the entirety of $\mathcal{C}$, we absorb a *separate* dual norm regularization term into the min-max objective for each possible cost, defining our new objective as

$$\min_{\beta} \max_{c \in \mathcal{C}} \left[ R^c_{\text{s-hinge}}(\beta) + \lambda u_* \|\beta\|_* \right]. \tag{9.4}$$

Contrast this with typical regularization, which (e.g. for $\ell_2$) would instead attempt to solve $\min_\beta \left[ \max_{c \in \mathcal{C}} \left[ R^c_{\text{s-hinge}}(\beta) \right] + \lambda \|\beta\|_2 \right]$, where regularization does not depend on the inner maximization. For brevity, in the remainder of this work we leave the regularization implicit, writing simply $R^c_{\text{s-hinge}}(\beta)$. Proposition 9.4.3 shows that a single choice of $\lambda$ always suffices: whereas the structure-aware regularizer depends on the cost, the ideal coefficient does not. By setting $\lambda$ appropriately and optimizing this new objective we account for the cost uncertainty "for free", effectively solving the problem across all costs in $\mathcal{C}$ *simultaneously* with a separate, appropriate regularization term for each. Importantly, this means our solution will be optimal with respect to the regularized objective. But once a solution is found, we can evaluate the adversarial risk without the regularization term and Theorem 9.2.5 will still apply.

### 9.5.1 Solving the Objective in the Full-Batch Setting with the Subgradient Method

Given a train set $\{(x_i, y_i)\}_{i=1}^n$, perhaps the simplest idea for solving Equation (9.4) is to follow the gradient of the empirical adversarial risk. The validity of this approach is not trivial because of the min-max formulation, but we show that it is indeed correct by invoking the following result:

**Theorem 9.5.1** (Danskin's Theorem [Bertsekas, 1997])**.** *Suppose $f : \mathbb{R}^n \times \mathcal{Z} \to \mathbb{R}$ is a continuous function, where $\mathcal{Z} \subset \mathbb{R}^m$ is a compact set. Define $g(x) := \max_{z \in \mathcal{Z}} f(x, z)$. Then $g(x)$ is convex in $x$ if $f(x, z)$ is convex in $x$ for every $z \in \mathcal{Z}$. Furthermore, $\partial_x g(x) = \mathrm{Conv}\,\{\partial_x f(x, z) \; : \; z \in \arg\max_z f(x, z)\}$, where $\partial$ is the subdifferential and "*Conv*" indicates the convex hull.*

Note that this requires $\mathcal{Z} \subset \mathbb{R}^m$, whereas we introduced $\mathcal{C}$ as a set of PD matrices in $\mathbb{R}^{d \times d}$. Using the assumption that $\mathcal{C}$ is compact and convex, we resolve this by associating to each cost matrix $\Sigma$ a vector of $d$ eigenvalues $\sigma_1^2 \ldots \sigma_d^2$ under a fixed basis, with each eigenvalue $\sigma_i^2$ constrained to lie in the set $[\sigma_{i\ell}^2, \sigma_{iu}^2]$. These lower and upper bounds allow us to re-parameterize the cost uncertainty set so that it is now a subset of $\mathbb{R}^d$. One remaining technicality is that this requires a fixed basis which may not be shared with the true cost. However, as a consequence of Lemma 9.2.3, our results will still be meaningful so long as there exists a $c \in \mathcal{C}$ which induces the same dual norm as the true cost. Without loss of generality, we suppose this basis is the identity.

Thus, we let $\mathcal{Z}$ denote the reparameterized uncertainty set $\mathcal{C}$ and plug in $\widehat{R}^{c=z}_{\text{s-hinge}}$ for $f$. We conclude that the subderivative of the worst-case strategic hinge loss is given by the subderivative of the loss evaluated at any cost in $\mathcal{C}$ which maximizes it. We can optimize the objective via the subgradient method (Algorithm 9 in the Appendix), where each subgradient evaluation involves a call to the maximization subroutine MAXLOSSCOST. We then combine this with well-known convergence results for the subgradient method and bound the generalization error via Theorem 9.2.5, giving the complete result:

**Theorem 9.5.2.** *Suppose we run Algorithm 9 for $T$ iterations and get classifier $\widehat{\beta}$. With probability $\geq 1 - \delta$, the worst-case 0-1 strategic loss over costs in $\mathcal{C}$ can be bounded by*

$$\max_{c \in \mathcal{C}} R^c_{0-1}(\widehat{\beta}) \leq \min_\beta \max_{c \in \mathcal{C}} R^c_{\text{s-hinge}}(\beta) + \mathcal{O}\left( \frac{LB}{\sqrt{T}} + (X + u_*) \frac{B + \sqrt{\ln 1/\delta}}{\sqrt{n}} \right).$$

The proof appears in Appendix H.5. Despite the difficulty posed by strategic response *and* an unknown cost (as made apparent in Section 9.3), Theorem 9.5.2 shows that robust one-shot learning is possible with sufficient consideration of the underlying structure.

### 9.5.2 Solving the Objective in the Minibatch Setting with Stochastic Mirror Descent-Ascent

Occasionally the number of training points $n$ may be sufficiently large that gradient descent is intractable, or the full dataset may not be available all at once. We would still like to be able to solve Equation (9.3), even when we cannot evaluate the full gradient. Unfortunately, the stochastic subgradient method is not an option here: evaluating the subderivative requires that we identify the cost that maximizes the *population* objective, which cannot be done on the basis of a subsample. As an alternative, we turn to a method from convex-concave optimization known as *Stochastic Mirror Descent-Ascent* (SMDA) [Nemirovski et al., 2009] which iteratively optimizes the min and max players to converge to a Nash equilibrium.

**Modifying mirror ascent to apply to our setting.** Since the objective is convex in $\beta$, minimization is straightforward. However, recall that the maximization over costs is non-concave. Previously we got around this by solving for the maximizing cost in a non-differentiable manner and then invoking Danskin's theorem—but the guarantees given by Nemirovski et al. [2009] require that we take iterative steps of gradient ascent on the adversary's (assumed concave) objective. We address this via an *algorithmic $\epsilon$-net*. Specifically, we can relax Equation (9.4) to a maximization over a *finite* set of costs $\mathcal{S} \subset \mathcal{C}$ such that the solution to this new objective is provably close to that of the original problem. Given such a set, the new objective becomes $\min_\beta \max_{c \in \mathcal{S}} R^c_{\text{s-hinge}}(\beta)$. Observe that the solution to this objective is equivalent to the solution to

$$\min_{\beta} \max_{q \in \Delta(|\mathcal{S}|)} \sum_{c_i \in \mathcal{S}} q_i R^{c_i}_{\text{s-hinge}}(\beta), \tag{9.5}$$

where $\Delta(|\mathcal{S}|)$ is the $|\mathcal{S}|$-simplex. In other words, we can solve this problem over all convex combinations of the risks under the different costs in $\mathcal{S}$ and arrive at the same solution [Rosenfeld et al., 2022c]. Crucially, unlike our original objective, Equation (9.5) is convex-concave, so it can be efficiently optimized via SMDA! The correctness of this relaxation relies on the property proven in Lemma 9.2.3: strictly speaking, the strategic loss depends only the *scalar* dual norm, not on the full cost matrix.

**Algorithm 7** Stochastic Mirror Descent-Ascent on the regularized strategic hinge loss

```
Input: Batch size $n$, Iterations $T$, Costs $\mathcal{C}$, Upper bound $u_*$, Regularization $\lambda$,
Discretization $\epsilon$, Step sizes $\eta_q, \eta_\beta$.
define: $\beta^{(0)} \leftarrow \mathbf{0}$
        $v \leftarrow [\epsilon(\sigma_{1\ell}^{-2} - \sigma_{1u}^{-2}), \ldots, \epsilon(\sigma_{d\ell}^{-2} - \sigma_{du}^{-2})]^\top$
        $q^{(0)} \leftarrow \epsilon\mathbf{1}$
for $t = 1, \ldots, T$ do
    Draw samples $\{(x_i, y_i)\}_{i=1}^n$.
    $c_k \leftarrow \vec{\sigma_u}^{-2} + k \cdot v, \quad k \in \{1, \ldots, \lceil 1/\epsilon \rceil\}$
    $q' \leftarrow q^{(t-1)}$
    $q'_k \leftarrow q'_k \exp(\eta_q \widehat{R}^{c_k}_{\text{s-hinge}}(\beta^{(t-1)}))$
    $q^{(t)} \leftarrow q' / \sum_k q'_k$
    $\beta^{(t)} \leftarrow \beta^{(t-1)} - \eta_\beta \left(\sum_k q_k^{(t)} \nabla \widehat{R}^{c_k}_{\text{s-hinge}}(\beta^{(t-1)})\right)$
end for
return $\widehat{\beta} := \frac{1}{T}\sum_{t=1}^T \beta^{(t)}$
```

The gap between the solution to the original objective and Equation (9.5) scales inversely with the fineness of discretization, so we want the maximum distance between any cost $c \in \mathcal{C}$ and its closest neighbor in $\mathcal{S}$ to be as small as possible. However, the memory and compute requirements of SMDA scale linearly with $|\mathcal{S}|$—and error scales as $\sqrt{\ln |\mathcal{S}|}$—so we also cannot let it grow too large. To balance these two considerations, we carefully construct a discretization over the "diagonal" of the space of eigenvalue intervals in $\mathcal{C}$, leading to a set $\mathcal{S}$ with cardinality $\Theta\left(\frac{TD}{\ln T}\right)$, where $T$ is the number of iterations and $D := \max_i 1/\sigma_{i\ell}^2 - 1/\sigma_{iu}^2$ is the diameter of $\mathcal{C}$. The exact construction for $\mathcal{S}$ appears in Algorithm 7. With this careful discretization, the computational cost to achieve a fixed sub-optimality gap is *dimension independent* for $p$-norms with $p \leq 2$, and the worst-case scaling (when $p = \infty$) is $\mathcal{O}\left(d^2/\ln d\right)$. In contrast, a typical $\epsilon$-net would have $|\mathcal{S}| = \Theta(\exp(d))$, with memory and compute scaling commensurately.

**Theorem 9.5.3.** *Suppose we run Algorithm 7 for $T$ rounds with discretization $\epsilon = \Theta\left(\frac{\ln T}{TD}\right)$ and get classifier $\widehat{\beta}$. Then over the randomness of the stochastic gradients, its expected sub-optimality*[3] *is bounded by*

$$\mathbb{E}[\max_{c\in\mathcal{C}} R^c_{\textit{s-hinge}}(\widehat{\beta})] \lesssim \min_\beta \max_{c\in\mathcal{C}} R^c_{\textit{s-hinge}}(\beta) + \mathcal{O}\left(\frac{LB}{\sqrt{T}} + B(X + u_*)\sqrt{\frac{\ln TD}{T}}\right).$$

[3]Exponential concentration of the adversarial risk to its expectation follows from Nemirovski et al. [2009], Proposition 3.2.

The proof can be found in Appendix H.6. Remarkably, convergence to the population minimax solution occurs at the same rate as the full-batch setting up to logarithmic factors—the cost of stochasticity is surprisingly small. Once again, we can use MAXLOSSCOST to evaluate the adversarial strategic risk and then apply Theorem 9.2.5 to get a generalization bound on the worst-case 0-1 error.

## 9.6 Conclusion

This paper studies robust strategic learning under unknown user costs in the challenging one-shot setting. Our results suggest that uncertainty in how users respond should be considered an integral aspect of strategic learning: motivated by realistic problem domains which permit only a single action (which must be immediately effective), we provide a learning framework based on distributionally robust optimization for modeling—and robustly handling—this uncertainty. We begin by showing that even a miniscule error in estimating the true cost can cause substantial error in deployment, motivating the use of an uncertainty set of possible costs. Next, we propose a natural proxy to the intractable min-max objective over this set, and we design two efficient algorithms for different settings that converge to the empirical solution, for which we also provide generalization guarantees.

Our approach highlights the value of dual norm regularization, which ensures good performance while accounting for the structure of strategic behavior coupled with cost uncertainty. Such structure can both harm accuracy (on negative examples) and help it (on positive examples). One important implication is that it often does not suffice to simply be more conservative, as this can fail to take advantage of settings where the learner's and agents' incentives are aligned. Rather, maintaining robustness without substantially sacrificing accuracy requires more careful consideration of and accounting for the interests and actions of future users.

# Chapter 10

# Outliers with Opposing Signals Have an Outsized Effect on Neural Network Optimization

This chapter is based on Rosenfeld and Risteski [2024]:
Rosenfeld, E. & Risteski, A.
Outliers with Opposing Signals have an Outsized Effect on Neural Network Optimization.
In *International Conference on Learning Representations*, 2024.

## 10.1 Introduction

There is a steadily growing list of intriguing properties of neural network (NN) optimization which are not readily explained by classical tools from optimization. Likewise, we have varying degrees of understanding of the mechanistic causes for each. Extensive efforts have led to possible explanations for the effectiveness of Adam [Kingma and Ba, 2014], Batch Normalization [Ioffe and Szegedy, 2015] and other tools for successful training—but the evidence is not always entirely convincing, and there is certainly little theoretical understanding. Other findings, such as grokking [Power et al., 2022] or the edge of stability [Cohen et al., 2021], do not have immediate practical implications but provide new ways to study what sets NN optimization apart. These phenomena are typically considered in isolation—though they are not completely disparate, it is unknown what specific underlying causes they may share. Clearly, a better understanding of NN training dynamics in a specific context can lead to algorithmic improvements [Chen et al., 2021c]; this suggests that any commonality will be a valuable tool for further investigation.

In this work, we identify a phenomenon in NN optimization which offers a new perspective on many of these prior observations and which we hope will contribute to a deeper understanding of how they may be connected. While we do not (and do not claim to) give a complete explanation, we present strong qualitative and quantitative evidence for a single high-level idea—one which naturally fits into several existing narratives and suggests a more coherent picture of their origin. Specifically, we demonstrate the prevalence of paired groups of outliers in natural data which have a significant influence on a network's optimization dynamics. These groups are characterized by the inclusion of one or more (relatively) large magnitude features that dominate the network's output at initialization and throughout most of training. In addition to their magnitude, the other distinctive property of these features is that they provide large, consistent, and *opposing* gradients, in that following one group's gradient to decrease its loss will increase the other's by a similar amount. Because of this structure, we refer to them as *Opposing Signals*. These features share a non-trivial correlation with the target task, but they are often not the "correct" (e.g., human-aligned) signal. In fact, in many cases these features perfectly encapsulate the classic statistical conundrum of "correlation vs. causation"—for example, a bright blue sky background does not determine the label of a CIFAR image, but it does most often occur in images of planes. Other features *are* very relevant, such as the presence of wheels and headlights in images of trucks and cars, or the fact that a colon often precedes either "the" or a newline token in written text.

Opposing signals are most easily understood with an example, which we will give along with a brief outline of their effect on training dynamics; a more detailed description is presented in Section 10.3. Figure 10.1 depicts the training loss of a ResNet-18 [He et al., 2016] trained with full-batch gradient descent (GD) on CIFAR-10 [Krizhevsky and Hinton, 2009], along with a few dominant outlier groups and their respective losses. In the early stages of training, the network enters a narrow valley in weight space which carefully balances the pairs' opposing gradients; subsequent sharpening of the loss landscape [Jastrzębski et al., 2020, Cohen et al., 2021] causes the network to oscillate with growing magnitude along particular axes, upsetting this balance. Returning to our example of a sky background, one step results in the class `plane` being assigned greater probability for all images with sky, and the next will reverse that effect. In essence, the "sky $=$ `plane`" subnetwork grows and shrinks.[1] The direct result of this oscillation is that the

[1]It would be more precise to say "strengthening connections between regions of the network's output and neurons which have large activations for sky-colored inputs". Though we prefer to avoid informal terminology, this example makes clear that the more relaxed phrasing is usually much cleaner. We therefore employ it when the intended meaning is clear.

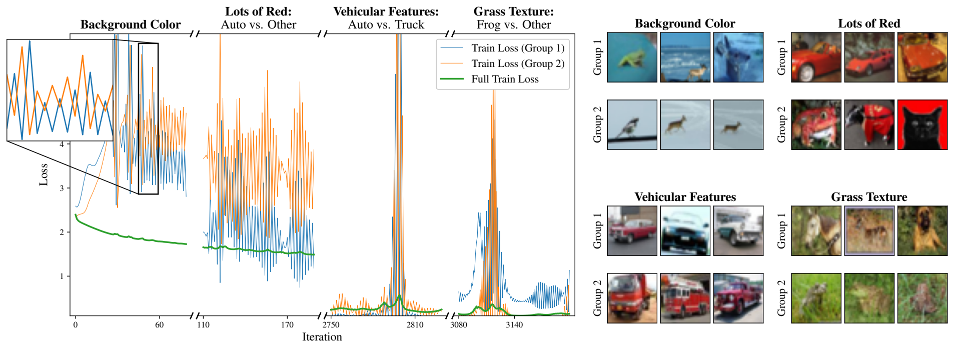


Figure 10.1: **Training dynamics of neural networks are heavily influenced by outliers with opposing signals.** We plot the overall loss of a ResNet-18 trained with GD on CIFAR-10, plus the losses of a small but representative set of outlier groups. These groups have consistent *opposing signals* (e.g., wheels and headlights can mean either `car` or `truck`). Throughout training, losses on these groups oscillate with growing and shrinking amplitude—this has an obvious correspondence to the intermittent spikes in overall loss and appears to be a direct cause of the edge of stability phenomenon.

network's loss on images of planes with a sky background will alternate between sharply increasing and decreasing with growing amplitude, with the exact opposite occurring for images of *non*-planes with sky. Consequently, the gradients of these groups will alternate directions while growing in magnitude as well. As these pairs represent a small fraction of the data, this behavior is not immediately apparent from the overall training loss—but eventually, it progresses far enough that the overall loss spikes. As there is an obvious direct correspondence between these two events throughout, we conjecture that opposing signals are a direct cause of the *edge of stability* phenomenon [Cohen et al., 2021]. We also note that the most influential signals appear to increase in complexity over time [Nakkiran et al., 2019].

We repeat this experiment across a range of vision architectures and training hyperparameters: though the precise groups and their order of appearance change, the pattern occurs consistently. We also verify this behavior for transformers on next-token prediction of natural text and small ReLU MLPs on simple 1D functions; we give some examples of opposing signals in text in Appendix I.2. However, we rely on images for exposition because it offers the clearest intuition. To isolate this effect, most of our experiments use GD, but we observe similar patterns during SGD which we present in Section 10.4.

**Summary of contributions.** The primary contribution of this paper is demonstrating the existence, pervasiveness, and large influence of opposing signals during NN optimization. We further present our current best understanding, with supporting experiments, of how these signals *cause* the observed training dynamics—in particular, we provide evidence that it is a consequence of depth and steepest descent methods. We complement this discussion with a toy example and an analysis of a two-layer linear net on a simple model. Notably, though rudimentary, our explanation enables concrete qualitative predictions of NN behavior during training, which we confirm experimentally. It also provides a new lens through which to study modern stochastic optimization methods, which we highlight via a case study of SGD vs. Adam. We see possible connections between opposing signals and a wide variety of phenomena in NN optimization and generalization, including *grokking* [Power et al., 2022], *catapulting/slingshotting* [Lewkowycz et al., 2020, Thilak et al., 2022], *simplicity bias* [Valle-Perez et al., 2019], *double descent* [Belkin et al., 2019, Nakkiran et al., 2020], and Sharpness-Aware Minimization [Foret et al., 2021]. We discuss these and other connections in Section 10.5.

## 10.2 Characterizing and Identifying Opposing Signals

Though their influence on aggregate metrics is non-obvious, identifying outliers with opposing signals is straightforward. Our methodology is as follows: when training a network with GD, we track its loss on each individual training point. For a given iteration, we select the training points whose loss exhibited the most positive and most negative change in the preceding step (there is large overlap between these sets in successive steps). This set will sometimes contain multiple opposing signals, which we distinguish via visual inspection. This last detail means that the images we depict are not random, but we emphasize that it would not be correct to describe this process as cherry-picking: though precise quantification is difficult, these signals consistently obey the maxim “I know it when I see it”. This is particularly true for images, such as the groups in Figure 10.1 which have easily recognizable patterns. To demonstrate this fact more generally, Appendix I.8 contains the pre-inspection samples for a ResNet-18, VGG-11 [Simonyan and Zisserman, 2014], and a small Vision Transformer [Dosovitskiy et al., 2020] at several training steps and for multiple seeds; we believe the implied groupings are immediate, even if not totally objective. We see algorithmic approaches to automatically clustering these samples as a direction for future study—for example, one could select samples by correlation in their loss time-series, or by gradient alignment.

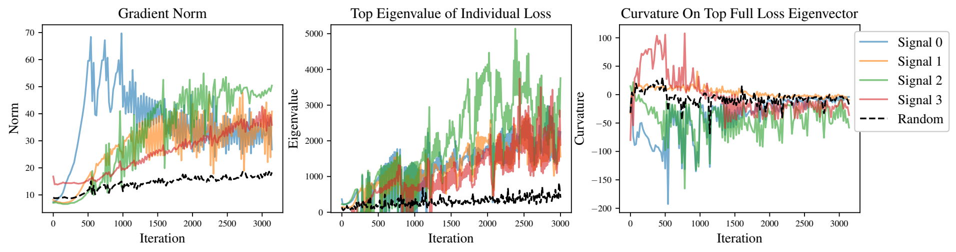


Figure 10.2: **Tracking other metrics which characterize outliers with opposing signals.** Maximal per-step change in loss relates to other useful metrics, such as per-sample gradient norm and curvature. We combine each pair of groups in Figure 10.1 to create training subsets which each exemplify one "signal": we see that these samples are also significant outliers according to the other metrics. (For a point $x$, "Curvature on Top Full Loss Eigenvector" is defined as $v^\top H(x)v$, where $v$ is the top eigenvector of the full loss Hessian and $H(x)$ is the Hessian of the loss on $x$ alone.)

**Measuring alternative metrics.** Given how these samples are selected, several other characterizations seem appropriate. For instance, one-step loss change is often a reasonable proxy for gradient norm; we could also consider the largest eigenvalue of the loss of the *individual point*, or how much curvature it has in the direction of the overall loss's top eigenvector. For large networks these options are far more compute-intensive than our chosen method, but we can evaluate them on specific small subsets. In Figure 10.2 we track these metrics for several opposing group pairs and we find that they are consistently much larger than that of random samples from the training set.

### 10.2.1 On the Possibility of a Formal Definition

Though the features and their exemplar samples are immediately recognizable, **we do not attempt to *exactly* define a "feature", nor an "outlier" with respect to that feature.** The presence of a particular feature is often ambiguous, and it is difficult to define a clear threshold for what makes a given point an outlier.[2] Thus, instead of trying to exactly partition the data, we simply note that these heavy tails *exist* and we use the most obvious outliers as representatives for visualization. In Figures 10.1 and 10.2 we choose an arbitrary cutoff of twenty samples per group.

We also note that what qualifies as an opposing signal or outlier may vary over

[2]In the case of language—where tokenization is discrete and more interpretable—a precise definition is sometimes possible. For example, one opposing pair in Appendix I.2 consists of sequences whose penultimate token is a colon and whose last token is either "the" or a newline.

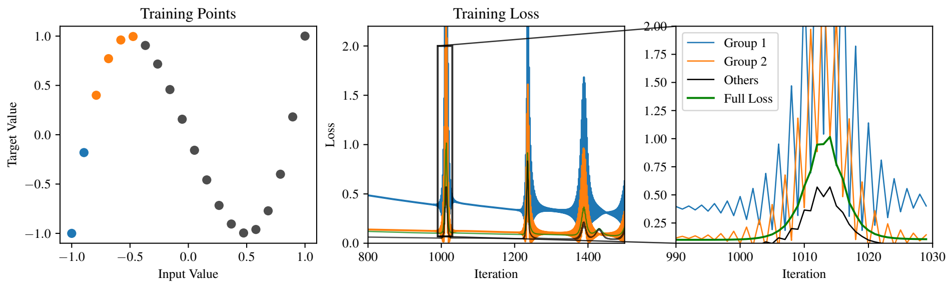


Figure 10.3: **Opposing signals when fitting a Chebyshev polynomial with a small MLP.** Though the data lacks traditional "outliers", it is apparent that the network has some features which are most influential only on the more negative inputs (or whose effect is otherwise cancelled out by other features). Since the correct use of this feature is opposite for these two groups, they provide opposing signals.

time. For visual clarity, Figure 10.1 depicts the loss on only the most dominant group pair in its respective training phase, but this pattern occurs simultaneously for many different signals and at multiple scales throughout training. Further, the opposing signals are with respect to the model's internal representations (and the label), not the input space itself; this means that the definition is also a property of the architecture. For example, following Cohen et al. [2021] we train a small MLP to fit a Chebyshev polynomial on evenly spaced points in the interval $[-1, 1]$ (Figure 10.3). This data has no "outliers" in the traditional sense, and it is not immediately clear what opposing signals are present. Nevertheless, we observe the same alternating behavior: we find a pair where one group is a small interval of $x$-values and the opposing group contains its neighbors, all in the range $[-1, -0.5]$. This suggests that the network has internal activations which are heavily influential only for more negative $x$-values. In this context, these two groups are the outliers.

## 10.3 Understanding the Effect of Opposing Signals

Beyond noting their existence, our eventual goal will be to derive actionable insights from this finding. To do this, it is necessary to gain a better understanding of *how* these outliers cause the observed behavior. In this section we give a simplified "mental picture" which serves as our current understanding of this process. We begin with an informal discussion of the outsized influence of opposing signals and how they lead to progressive sharpening; this subsection collates and expands on prior work to give important context for how these signals differ from typically imagined "noise". Next, we give a mechanistic description of the specific effect of opposing signals with a toy example. This explanation is intentionally high-level, but we will eventually see how it gives concrete predictions of specific behaviors, which we then verify on real networks. Finally, we prove that this behavior occurs on a two-layer linear network under a simple model.

### 10.3.1 Progressive Sharpening, and Intuition for Why these Features are so Influential

At a high level, most variation in the input is unneeded when training a network to minimize predictive error—particularly with depth and high dimension, only a small fraction of information will be propagated to the last linear layer [Huh et al., 2021]. Starting from random initialization, training a network aligns adjacent layers' singular values [Saxe et al., 2013, Mulayoff and Michaeli, 2020] to amplify meaningful signal while downweighting noise,[3] growing *sensitivity* to the important signal. This sensitivity can be measured, for example, by the spectral norm of the input-output Jacobian, which grows during training [Ma and Ying, 2021]; it has also been connected to growth in the norm of the output layer [Wang et al., 2022b].

Observe that with this growth, small changes to *how the network processes inputs* become more influential. Hypothetically, a small weight perturbation could massively increase loss by redirecting unhelpful noise to the subspace to which the network is most sensitive, or by changing how the last layer uses it. The increase of this sensitivity thus represents precisely the growth of loss Hessian spectrum, with the strength of this effect increasing with depth [Wang et al., 2016, Du et al., 2018, Mulayoff and Michaeli, 2020].[4]

Crucially, this sharpening also depends on the structure of the input. If the noise is independent of the target, it will be downweighted throughout training. In contrast, *genuine signals which oppose each other* will be retained and perhaps even further amplified by gradient descent; this is because the "correct" feature may be much smaller in magnitude (or not yet learned), so using the large, "incorrect" feature is often the most immediate way of minimizing loss. As a concrete example, observe that a randomly initialized network will lack the features required for the subtle task of distinguishing birds from planes. But it *will* capture the presence of sky, which is very useful for reducing loss on such images by predicting the conditional $p(\text{class} \mid \text{sky})$ (this is akin to the "linear/shallow-first" behavior described by Nakkiran et al. [2019], Mangalam and Prabhu [2019]). Thus, any method attempting to

[3]In this discussion we use the term "noise" informally. We refer not necessarily to pure randomness, but more generally to input variation which is least useful in predicting the target.

[4]The coincident growth of these two measures was previously noted by Ma and Ying [2021], Gamba et al. [2023], MacDonald et al. [2023], though they did not make explicit this connection to how the network processes different types of input variance.

minimize loss as fast as possible (e.g., steepest descent) may actually upweight these features. Furthermore, amplified opposing signals will cause greater sharpening than random noise, because using a signal to the benefit of one group is maximally harmful for the other—e.g., confidently predicting `plane` whenever there is sky will cause enormous loss on images of other classes with sky. Since random noise is more diffuse, this effect is less pronounced.

This description is somewhat abstract. To gain a more precise understanding, we illustrate the dynamics more explicitly on a toy example.

### 10.3.2 Illustrating with a Hypothetical Example of Gradient Descent

Consider the global loss landscape of a neural network: this is the function which describes how the loss changes as we move through parameter space. Suppose we identify a direction in this space which corresponds to the network's use of the "sky" feature to predict `plane` versus some other class. That is, we will imagine that whenever the input image includes a bright blue background, moving the parameters in one direction increases the logit of the `plane` class and decreases the others, and vice-versa. We will also decompose this loss—**among images with a sky background, we consider *separately* the loss on those labeled `plane` versus those with any other label.** Because the sky feature has large magnitude, a small change in weight space will induce a large change in the network outputs— i.e., a small movement in the direction "sky = `plane`" will greatly increase loss on these non-`plane` images.

Figure 10.4 depicts this heavily simplified scenario. Early in training, optimizing this network with GD will rapidly move towards the minimum along this direction. In particular, until better features are learned, the direction of steepest descent will lead to a network which upweights the sky feature and predicts $p(\text{class} \mid \text{sky})$ whenever it occurs. Once sufficiently close to the minimum, the gradient will point "through the valley" towards amplifying the more relevant signal [Xing et al., 2018]. However, this will also cause the sky feature to grow in magnitude—as well as its *potential* influence were the weights to be selectively perturbed, as described above. Both these factors contribute to progressive sharpening.

Here we emphasize the distinction between the loss on the *outliers* and the full train loss. As images without sky are not nearly as sensitive to movement along this axis, their gradient and curvature is much smaller—and since they comprise the majority of the dataset, the global loss landscape may not at first be significantly affected. Continued optimization will oscillate across the minimum with growing magnitude, but this growth may not be immediately apparent. Furthermore, *progress orthogonal to these oscillations need not be affected*—we find some evidence that these two processes occur somewhat independently, which we present in

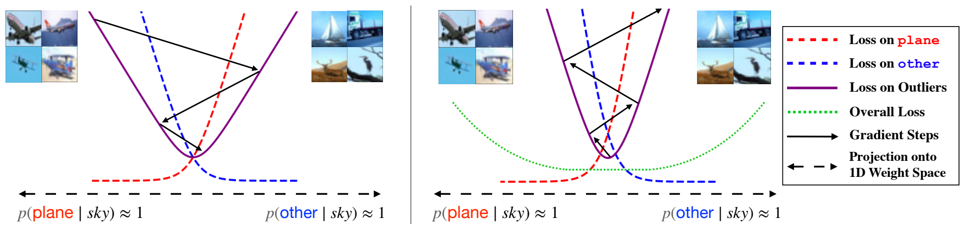


Figure 10.4: **A toy illustration of the effect of opposing signals.** Images with many blue pixels cause large activations, with high loss sensitivity. We project the loss to the hypothetical weight-space dimension "sky = `plane`". **Left:** Early optimization approaches the minimum, balancing the opposing gradients for `plane` and `other` (these are losses for *separate* training subsets: those labeled `plane` vs. those with any other label—the purple curve is their average). Progress continues through this valley, further growing the feature magnitude. **Right**: The valley sharpens and the iterates diverge, alternating between high and low loss for each group. Because most training points are insensitive to this axis, the overall loss may not be noticeably affected at first. Eventually either (a) the loss growth forces the network to downweight "sky", flattening the valley; or (b) the weights "catapult" to a different basin.

Section 10.4. Returning to the loss decomposition, we see that these oscillations will cause the losses to grow and *alternate*, with one group having high loss and then the other. Eventually the outliers' loss increases sufficiently and the overall loss spikes, either flattening the valley and returning to the first phase, or "catapulting" to a different basin [Wu et al., 2018, Lewkowycz et al., 2020, Thilak et al., 2022]. This phenomenon is depicted in Figure 10.1. Finally, we note that if one visualizes the dynamics in Figure 10.4 from above—so the left/right direction on the page becomes up/down—it gives exactly the pattern of a network's weights projected onto the top eigenvector of the Hessian (e.g., Figure 10.6b later in this work).

**Verifying our toy examples's predictions.** Though this explanation lacks precise details, it does enable concrete predictions of network behavior during training. Figure 10.5 tracks the predictions of a ResNet-18 on an image of sky—to eliminate possible confounders, we create a synthetic image as a single color block. Though the "`plane` vs. `other`" example seems almost *too* simple, we see exactly the described behavior—initial convergence to the minimum along with rapid growth in feature norm, followed by oscillation in class probabilities. Over time, the network learns to use other signal and downweights the sky feature, as evidenced by the slow decay in feature norm. We reproduce this figure for many other inputs and for a

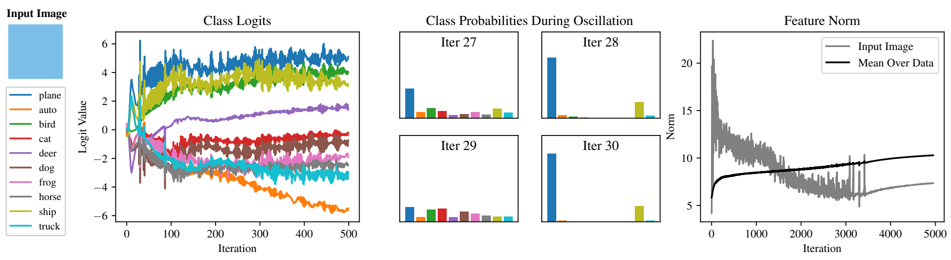


Figure 10.5: **Passing a sky-colored block through a ResNet during GD precisely tracks the predictions of our toy example. Left:** In the first phase, the network rapidly learns to use the sky feature to minimize loss. As signal is amplified, so too is the sky-colored input, and oscillation begins as depicted in Figure 10.4. **Middle:** During oscillation, gradient steps alternate along the axis "sky = `plane`" (and a bit `ship`). **Right:** The initial phase amplifies the sky input, causing rapid growth in feature norm. The network then oscillates, slowly learning to downweight this feature and rely on other signal (average feature norm provided for comparison).

VGG-11-BN in Appendix I.3, with similar findings.

Our example also suggests that **oscillation serves as a valuable regularizer that reduces reliance on easily learned opposing signals which may not generalize.** When a signal is used to the benefit of one group and the detriment of another, the advantaged group's loss goes down while the other's goes up, meaning the latter's gradient grows in magnitude while the former's shrinks. As the now gradient-dominating group is also the one disadvantaged by the use of this signal, the network will be encouraged to downweight this feature. In Appendix I.3.3 we reproduce Figure 10.5 with a VGG-11-BN trained with a very small learning rate to closely approximate gradient flow. We see that gradient flow and GD are very similar until reaching the edge of stability. After this point, the feature norm under GD begins to slowly decay while oscillating; in contrast, in the absence of oscillation, the feature norms of opposing signals under gradient flow grow continuously. If it is the case that opposing signals represent "simple" features which generalize worse, this could help to explain the poor generalization of gradient flow. A similar effect was observed by Jastrzębski et al. [2020], who noted that large initial learning rate leads to a better-conditioned loss landscape later.

### 10.3.3 Theoretical Analysis of Opposing Signals in a Simple Model

To demonstrate this effect formally, we study misspecified linear regression on inputs $x \in \mathbb{R}^d$ with a two-layer linear network. Though this model is simplified, it enables preliminary insight into the factors we think are most important for these dynamics to occur. Since we are analyzing the dynamics from initialization until the stability threshold, it will be sufficient to study the trajectory of *gradient flow*—for reasonable step sizes $\eta$, a similar result then follows for gradient descent. Our analysis reproduces the initial phase of quickly reducing loss on the outliers, followed by the subsequent growth in sensitivity to the *way* the opposing signal is used—i.e., progressive sharpening. We also verify this pattern (and the subsequent oscillation, which we do not formally prove) in experiments on real and synthetic data in Appendix I.5.

We remark that one relevant factor which our model lacks is the concept of a "partially useful" signal as described at the end of Section 10.3.1. This seems to require a somewhat more complex model to properly capture (e.g., multinomial logistic regression) so we view this analysis as an early investigation, capturing only part of relevant aspects of the phenomena we observe.

**Model.** We model the observed features as a distribution over $x \in \mathbb{R}^{d_1}$, assuming only that its covariance $\Sigma$ exists—for clarity we treat $\Sigma = I$ in the main text. We further model an additional vector $x_o \in \mathbb{R}^{d_2}$ representing the opposing signal, with $d_2 > d_1$. We will suppose that on some small fraction of outliers $p \ll 1$, $x_o \sim \text{Unif}\left(\left\{\pm\sqrt{\frac{\alpha}{pd_2}}\mathbf{1}\right\}\right)$ ($\mathbf{1}$ is the all-ones vector) for some $\alpha$ which governs the feature magnitude, and we let it be $\mathbf{0}$ on the remainder of the dataset. We model the target as the linear function $y = \beta^\top x + \frac{1}{\sqrt{d_2}}\mathbf{1}^\top |x_o|$; this captures the idea that the signal $x_o$ correlates strongly with the target, but in opposing directions of equal strength. Finally, we parameterize the network with vectors $b \in \mathbb{R}^{d_1}, b_o \in \mathbb{R}^{d_2}$ and scalar $c$ in one single vector $\theta$, as $f_\theta(x) = c \cdot (b^\top x + b_o^\top x_o)$. Note the specific distribution of $x_o$ is unimportant—furthermore, in our simulations we observed the exact same pattern with cross-entropy loss. From our experiments and this analysis, it seems that depth and a small signal-to-noise ratio are the only elements needed for this behavior to arise.

**Setup.** A standard initialization would be to sample $[b, b_o]^\top \sim \mathcal{N}(0, \frac{1}{d_1+d_2}I)$, which would then imply highly concentrated distributions for the quantities of interest. As tracking the precise concentration terms would not meaningfully contribute to the analysis, we simplify by directly assuming that at initialization these quantities are equal to their expected order of magnitude: $\|b\|_2^2 = \mathbf{1}^\top b = \frac{d_1}{d_1+d_2}$, $\|b_o\|_2^2 = \mathbf{1}^\top b_o = \frac{d_2}{d_1+d_2}$, and $b^\top \beta = \frac{\|\beta\|}{\sqrt{d_1+d_2}}$. Likewise, we let $c = 1$, ensuring that both layers have the same norm. We perform standard linear regression by minimizing the population loss $L(\theta) := \frac{1}{2}\mathbb{E}[(f_\theta(x) - y)^2]$. We see that the minimizer of this objective has $b_o = \mathbf{0}$ and $cb = \beta$. However, an analysis of gradient flow will elucidate how depth and strong opposing signals lead to sharpening as this minimum is approached.

**Results.** In exploring progressive sharpening, Cohen et al. [2021] found that sometimes the model would have a brief *decrease* in sharpness, particularly for square loss. In fact, this is consistent with our above explanation: for large $\alpha$ and a sharper loss (e.g. the square loss), the network will initially prioritize minimizing loss on the outliers, thus heavily reducing sharpness. Our first result proves that this occurs in the presence of large magnitude opposing signals:

**Theorem 10.3.1** (Initial *decrease* in sharpness)**.** *Let $k := \frac{d_2}{d_1} > 1$, and assume $\|\beta\| > \max\left(\frac{d_1}{\sqrt{d_1+d_2}}, \frac{24}{5}\right)$. At initialization, the sharpness $\|\nabla^2_\theta L(\theta)\|_2$ lies in $(\alpha, 3\alpha)$. Further, if $\sqrt{\alpha} = \Omega\left(\|\beta\| k \ln k\right)$, then both $b_o$ and the overall sharpness will* decrease *as $\tilde{O}(e^{-\alpha t})$ from $t = 0$ until some time $t_1 \leq \frac{\ln \|\beta\|/2}{2\|\beta\|}$.*

Proofs can be found in Appendix I.7. After this decrease, signal amplification can proceed—but this also means that the sharpness with respect to *how the network uses the feature $x_o$* will grow, so a small perturbation to the parameters $b_o$ will induce a large increase in loss.

**Theorem 10.3.2** (Progressive sharpening)**.** *If $\sqrt{\alpha} = \Omega\left(1 + \|\beta\|^2 k \ln k\right)$, then at starting at time $t_1$ the sharpness will increase linearly in $\|\beta\|$ until some time $t_2 \geq \frac{1}{2\|\beta\|_2^2}$, reaching at least $\frac{5}{8}\|\beta\|\alpha$. This lower bound on sharpness applies to each dimension of $b_o$.*

Oscillation will not occur during gradient flow—but for gradient descent with step size $\eta > \frac{16}{5\|\beta\|\alpha}$, $b_o$ will start to increase in magnitude while oscillating across the origin. If this growth continues, it will rapidly *reintroduce* the feature, causing the loss on the outliers to grow and alternate. Such reintroduction (an example of which occurs around iteration 3000 in Figure 10.5) seems potentially helpful for exploration. In Figure I.32 in the Appendix we simulate our model and verify exactly this sequence of events. We also show that an MLP trained on CIFAR-10 displays the same characteristic behavior.

### 10.3.4 Additional Findings

**Sharpness often occurs overwhelmingly in the first few layers.** Theorem 10.3.2 shows that progressive sharpening occurs specifically in $b_o$. Generally, our model suggests that sharpness will begin in the last layer but that that early in training it will shift to the earlier layers since they have more capacity to redirect the signal. In Appendix I.4 we track what fraction of curvature[5] of the top eigenvector lies in each

[5]The "fraction of curvature" is with respect to the top eigenvector of the loss Hessian. We partition this vector by network layer, so each sub-vector's squared norm represents that layer's contribution to

layer of various networks during training with GD. In a ResNet-18, sharpness occurs almost exclusively in the first convolutional layer after the first few training steps; the same pattern appears more slowly while training a VGG-11. In a Vision Transformer curvature occurs overwhelmingly in the embedding layer and very slightly in the earlier MLP projection heads. The text transformer (NanoGPT) follows the same pattern, though with less extreme concentration in the embedding. Thus it does seem to be the case that earlier layers have the most significant sharpness—especially if they perform dimensionality reduction or have particular influence over how the signal is propagated to later layers. This seems the likely cause of large gradients in the early layers of vision models [Chen et al., 2021c, Kumar et al., 2022], suggesting that this effect is equally influential during finetuning and pretraining and that further study can improve optimization.

**Batchnorm may smooth training, even if not the loss itself.** Cohen et al. [2021] noted that batchnorm (BN) [Ioffe and Szegedy, 2015] does not prevent networks from reaching the edge of stability and concluded, contrary to Santurkar et al. [2018], that BN does not smooth the loss landscape. We conjecture that the benefit of BN may be in downweighting the influence of opposing signals and mitigating this oscillation. In other words, BN may smooth the *optimization trajectory* of neural networks, rather than the loss itself (this is consistent with the distinction made by Cohen et al. [2021] between regularity and smoothness). In Section 10.4 we demonstrate that Adam *also* smooths the optimization trajectory and that minor changes to emulate this effect can aid stochastic optimization. We imagine that the effect of BN could also depend on the use of GD vs. SGD. Specifically, our findings hint at a possible benefit of BN which applies only to SGD: reducing the variance of imbalanced opposing signals across random minibatches.

**For both GD and SGD, approximately half of training points go up in loss on each step.** Though only the outliers are wildly oscillating, many more images contain some small component of the features they exemplify. Figure I.30 in the Appendix shows that the fraction of points which increase in loss hovers around 50% for every step—to some extent, a small degree of oscillation appears to be happening to the entire dataset.

**Different losses have different effects on sharpening.** Our model would predict that adding label smoothing to the cross-entropy loss should reduce sharpening, because smoothing reduces loss curvature under extreme overconfidence. Indeed, MacDonald et al. [2023] show this to be the case. This also hints at why logistic loss

the overall curvature.

may be more suitable for NN optimization, because it only has substantial curvature around $x = 0$ ($x$ being the logit, i.e. when prediction entropy is high), so unlike square or exponential loss, large magnitude features will not massively increase sharpness. We expect a similar property could contribute to the relative behavior of different activations (e.g. ReLU or tanh).

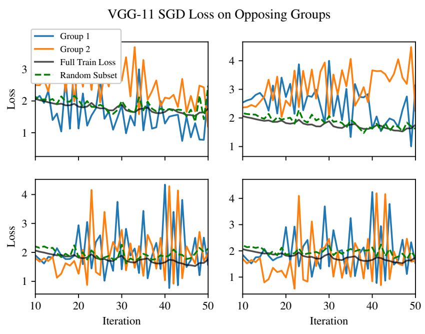
## 10.4 The Interplay of Opposing Signals and Stochasticity

Full-batch GD is not used in practice when training NNs. It is therefore pertinent to ask what these findings imply about stochastic optimization. We begin by verifying that this pattern persists during SGD. Figure 10.6a displays the losses for four opposing group pairs of a VGG-11-BN trained on CIFAR-10 with SGD batch size 128. We observe that the paired groups do exhibit clear opposing oscillatory patterns, but they do not alternate with every step, nor do they always move in opposite directions. This should not be surprising: we expect that not every batch will have a given signal in one direction or the other. For comparison, we include the *full* train loss in each figure—that is, including the points not in the training batch. We see that the loss on the outliers has substantially larger variance; to confirm that this is not just because the groups have many fewer samples, we also plot the loss on a random subset of training points of the same size. We reproduce this plot with a VGG-11 without BN in Figure I.31 in the Appendix.

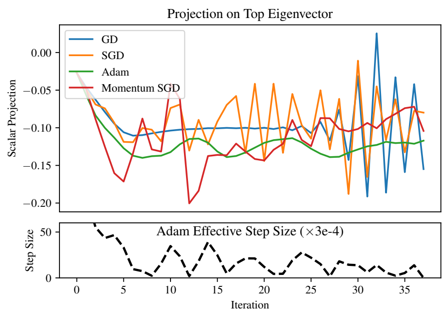
Having verified that this behavior occurs in the stochastic setting, we conjecture that current best practices for neural network optimization owe much of their success to how they handle opposing signals. As a proof of concept, we will make this more precise with a preliminary investigation of the Adam optimizer [Kingma and Ba, 2014].

### 10.4.1 How Adam Handles Gradients with Opposing Signals

To better understand their differences, Figure 10.6b visualizes the parameter iterates of Adam and SGD with momentum on a ReLU MLP trained on a 5k subset of CIFAR-10, alongside those of GD and SGD (all methods use the same initialization and sequence of training batches). The top figure is the projection of these parameters onto the top eigenvector of the loss Hessian of the network trained with GD, evaluated at the first step where the sharpness crosses $2/\eta$. We observe that SGD tracks a similar path to GD, though adding momentum mitigates the oscillation somewhat. In contrast, the network optimized with Adam markedly departs from this pattern, smoothly oscillating along one side. We identify three components of Adam which potentially contribute to this effect:



*(a)*



*(b)*

Figure 10.6: **Outliers with opposing signals have a significant influence even during SGD. Left:** We plot the losses of paired outlier groups on a VGG-11-BN trained on CIFAR-10, along with the full train loss for comparison. Modulo batch randomness, the outliers' loss follow the same oscillatory pattern with large magnitude. See appendix for the same without batchnorm. **Right (top):** We train a small MLP on a 5k subset of CIFAR-10 with various optimizers and project the iterates onto the top Hessian eigenvector. SGD closely tracks GD, bouncing across the valley; momentum somewhat mitigates the sharp jumps. Adam smoothly oscillates along one side. **Right (bottom):** Adam's effective step size drops sharply when moving too close or far from the valley floor.

**Advantage 1: Smaller steps along high curvature directions.** Adam's normalization causes smaller steps along the top eigenvector, especially near the minimum. The lower plot in Figure 10.6b shows that the effective step size in this direction—i.e., the absolute inner product of the parameter-wise step sizes and the top eigenvector—rapidly drops to zero as the iterates approach the valley floor (in the opposite direction, the gradient negates the momentum for the same effect). We conjecture that general normalization may not be essential to Adam's performance; we even expect it could be somewhat harmful by limiting exploration. On the other hand, normalizing steps by curvature *parameter-wise* does seem important; Pan and Li [2023] argue the same and show that parameter-wise gradient clipping improves SGD substantially. We highlight why this may be useful in the next point.

**Advantage 2: Managing heavy-tailed gradients and avoiding steepest descent.** Zhang et al. [2020] identified the "trust region" as an important contributor to Adam's success in attention models, pointing to heavy-tailed noise in the stochastic gradients. More recently, Kunstner et al. [2023] argued that Adam's superiority does not come from better handling noise, which they supported by experimenting with large batch sizes. Our result reconciles these contradictory claims by showing that the difficulty is not heavy-tailed *noise*, but strong, directed (and perhaps imbalanced) opposing signals. Unlike traditional "gradient noise", larger batch sizes may not reduce the effect of these signals—that is, the gradient is heavy-tailed (across parameters) even without being stochastic. Also of note: Adam's largest steps emulate Sign SGD, which is notably *not* a descent method. Figure 10.6b shows that Adam's steps are more parallel to the valley floor than those of steepest descent. Thus it seems advantageous to *intentionally* avoid steps along the gradient which point towards the local minimum, which might lead to over-reliance on these features. Indeed, Benzing [2022] observe that true second order methods perform worse than SGD on NNs, and Kunstner et al. [2023] show that Adam shares some behavior with Sign SGD with momentum. Here we see the value in taking small steps along high-curvature directions, particularly when we are close to the minimum. This point is also consistent with the observed generalization benefits of a large learning rate for SGD on NNs [Jastrzębski et al., 2020]; in fact, opposing signals naturally fit the concept of "easy-to-fit" features as modeled by Li et al. [2019]—who prove the benefit of *intentionally failing* to learn them.

**Advantage 3: Dampening.** Lastly, Adam's third important factor: downweighting the most recent gradient. Traditional SGD with momentum $\beta < 1$ takes a step which weights the current gradient by $\frac{1}{1+\beta} > \frac{1}{2}$. Though this makes intuitive sense, our results imply that heavily weighting the most recent gradient can be problematic. Instead, we expect an important addition is *dampening*, which multiplies the stochastic gradient at each step by some $(1 - \tau) < 1$. We observe that Adam's (unnormalized) gradient is equivalent to SGD with momentum and dampening both equal to $\beta_1$, plus a debiasing step. Recently proposed alternatives also include dampening in their momentum update but do not explicitly identify the distinction [Zhang et al., 2020, Pan and Li, 2023, Chen et al., 2023a].

### 10.4.2 Proof of Concept: Using These Insights to Aid Stochastic Optimization

To test whether our findings translate to practical gains, we design a variant of SGD which incorporates these insights. First, we use dampening $\tau = 0.9$ in addition to momentum. Second, we choose a global threshold: if the gradient magnitude for a parameter is above this threshold, we take a fixed step size (à la Sign SGD); otherwise, we take a gradient step as normal. The exact method appears in Appendix I.6.

Results in Appendix I.6 show that this approach matches Adam when training ResNet-56/110 on CIFAR-10 with learning rates for the unthresholded parameters across several orders of magnitude ranging from $10^{-4}$ to $10^{3}$. Notably, the fraction of parameters above the threshold (whose step size is fixed) is only around 10-25% per step. This implies that the trajectory and behavior of the network is dominated by this small fraction of parameters; the remainder can be optimized much more robustly, but their effect on the network's behavior is obscured. We therefore see the influence of opposing signals as a possible explanation for the "hidden" progress in grokking [Barak et al., 2022, Nanda et al., 2023]. We also compare this method to Adam for the early phase of training GPT-2 [Radford et al., 2019] on the OpenWebText dataset [Gokaslan et al., 2019]—not only do they perform the same, their loss similarity suggests that their exact trajectory may be very similar (Appendix I.6.2). Here the fraction of parameters above the threshold hovers around 50% initially and then gradually decays. The fact that many more parameters in the attention model are above the threshold suggests that the attention mechanism is more sensitive to opposing signals and that further investigation of how to mitigate this instability may be fruitful.

## 10.5 Discussion and Future Work

Many of the observations we make in this paper are not new, having been described in various prior works. Rather, this work identifies a possible *higher-order cause* which neatly ties these findings together. There are also many works which pursue a more theoretical understanding of each of these phenomena independently. Such analyses begin with a set of assumptions (on the data, in particular) and prove that the given behavior follows. In contrast, this work *begins* by identifying a condition—the presence of opposing signals—which we argue is likely a major cause of these behaviors. These two are not at odds: we believe in many cases our result serves as direct evidence for the validity of these modeling assumptions and that it may enable even more fine-grained analyses. This work provides an initial investigation which we hope will inspire future efforts towards a more complete understanding.

We now highlight some connections to these earlier findings. More general related work can be found in Appendix I.1.

**Heavy-tailed loss spectrum.** Earlier studies of the loss landscape noted a small group of very large outlier Hessian eigenvalues or Jacobian singular values (e.g. Sagun et al. 2016, 2017, Papyan 2018, see Appendix I.1 for more). Our method of identifying these paired groups, along with the metrics tracked in Figure 10.2, indicate that these outlier directions in the spectrum are precisely the directions with opposing signals in the gradient and that this pattern may be key to better understanding the generalization ability of NNs trained with SGD.

**Progressive sharpening and the edge of stability.** More recent focus has shifted to the top Hessian eigenvalue(s), where it was empirically observed that their magnitude (the loss "sharpness") grows during training [Jastrzębski et al., 2019, 2020, Cohen et al., 2021] (so-called *progressive sharpening*), leading to rapid oscillation in weight space [Xing et al., 2018, Jastrzębski et al., 2019]. Cohen et al. [2021] also found that for GD this coincides with a consistent yet non-monotonic decrease in training loss over long timescales, which they named the *edge of stability*. We observe that prior analyses have proven the *occurrence* of progressive sharpening and the edge of stability under various assumptions [Arora et al., 2022, Wang et al., 2022b], but the underlying *cause* has not been made clear. Our discussion, experiments, and theoretical analysis in Section 10.3 provide strong evidence for a genuine cause which aligns with several of these existing modeling assumptions. Roughly, our results seem to imply that progressive sharpening occurs when the network learns to rely on (or *not* rely on) opposing signals in a very specific way, while simultaneously amplifying overall sensitivity. This growth in sensitivity means a small parameter change modifying how opposing signals are used can massively increase loss. This leads to intermittent instability orthogonal to the "valley floor", accompanied by gradual training loss decay and occasional spikes as described by the toy example in Figure 10.4 and depicted on real data in Figure 10.1. Empirically, this oscillation seems somewhat independent of movement parallel to the floor (see Appendix I.6), but further study of the precise dynamics is needed.

**Spurious correlations, grokking, and slingshotting.** In images, the features corresponding to opposing signals match the traditional picture of "spurious correlations" surprisingly closely—it could be that a network maintaining balance or diverging along a direction also determines whether it continues to use a "spurious" feature or is forced to find an alternative way to minimize loss. Indeed, the exact phenomenon of a network "slingshotting" to a new region with improved generalization has been directly observed [Wu et al., 2018, Lewkowycz et al., 2020, Jastrzębski et al., 2021, Thilak et al., 2022]. *Grokking* [Power et al., 2022], whereby a network learns to generalize long after memorizing the training set, is closely related. Several works have shown that grokking is a "hidden" phenomenon, with gradual amplification of generalizing subnetworks [Barak et al., 2022, Nanda et al., 2023, Merrill et al., 2023]; it has even been noted to co-occur with weight oscillation [Notsawo Jr et al., 2023]. Our experiments in Section 10.4 and Appendix I.6 show that the influence of opposing signals obscures the behavior of the rest of the network, offering one possible explanation.

**Simplicity bias and double descent.** Nakkiran et al. [2019] observed that NNs learn functions of increasing complexity throughout training. Our experiments—particularly the slow decay in the norm of the feature embedding of opposing signals—lead us to believe it would be more accurate to say that they *unlearn* simple functions, which enables more complex subnetworks with smaller magnitude and better performance to take over. At first this seems at odds with the notion of *simplicity bias* [Valle-Perez et al., 2019, Shah et al., 2020], defined broadly as a tendency of networks to rely on simple functions of their inputs. However, it does seem to be the case that the network will use the simplest (e.g., largest norm) features that it can, so long as such features allow it to approach zero training loss; otherwise it may eventually diverge. This tendency also suggests a possible explanation for *double descent* [Belkin et al., 2019, Nakkiran et al., 2020]: even after interpolation, the network pushes towards greater confidence and the weight layers continue to balance [Saxe et al., 2013, Du et al., 2018], increasing sharpness. This could lead to oscillation, pushing the network to learn new features which generalize better [Wu et al., 2018, Rosenfeld et al., 2022b, Thilak et al., 2022]. This behavior would also be more pronounced for larger networks because they exhibit greater sharpening. Note that the true explanation is not quite so straightforward: generalization is sometimes improved via methods that *reduce* oscillation (like loss smoothing), implying that this behavior is not always advantageous. A better understanding of these nuances is an important subject for future study.

**Sharpness-Aware Minimization** Another connection we think merits further inquiry is Sharpness-Aware Minimization (SAM) [Foret et al., 2021], which is known to improve generalization of neural networks for reasons still not fully understood [Wen et al., 2023]. In particular, the better-performing variant is 1-SAM, which takes positive gradient steps on each training point in the batch individually. It it evident that several of these updates will point along directions of steepest descent/ascent orthogonal to the valley floor (and, if not normalized, the updates may be *very* large). Thus it may be that 1-SAM is in some sense “simulating” oscillation and divergence out of this valley in both directions, enabling exploration in a manner that would not normally be possible until the sharpness grows large enough—these intermediate steps would also encourage the network to downweight these features sooner and faster. Standard SAM would only take this step in one of the two directions, or perhaps not at all if the opposing signals are equally balanced. Furthermore, unlike 1-SAM the intermediate step would blend together all opposing signals in the minibatch. These possibilities seem a promising direction for further exploration.

## 10.6 Conclusion

The existence of small, paired groups of training data with such a significant yet non-obvious influence on neural network training raises as many questions as it answers. This work presents an initial investigation into the effect of opposing signals on various aspects of optimization, but there is still much more to understand. Though it is clear they have a large influence on training, less obvious is whether reducing their influence is *necessary* for improved optimization or simply coincides with it. At the same time, there is evidence that the behavior these signals induce may serve as an important method of exploration and/or regularization. If so, another key question is whether these two effects can be decoupled—or if the incredible generalization ability of neural networks is somehow inherently tied to their optimization instability.

# Appendices

# Appendix A

# Appendix for Chapter 2

## A.1 Proofs of Theorems 2.3.1 and 2.3.2

Here we provide the complete proofs for Theorems 2.3.1 and 2.3.2. We fist prove the following lemma, which is essentially a restatement of the Neyman-Pearson lemma [Neyman and Pearson, 1933] from statistical hypothesis testing.

**Lemma A.1.1 (Neyman-Pearson).** *Let $X$ and $Y$ be random variables in $\mathbb{R}^d$ with densities $\mu_X$ and $\mu_Y$. Let $h : \mathbb{R}^d \to \{0, 1\}$ be a random or deterministic function. Then:*

1. *If $S = \left\{ z \in \mathbb{R}^d : \frac{\mu_Y(z)}{\mu_X(z)} \le t \right\}$ for some $t > 0$ and $\mathbb{P}(h(X) = 1) \ge \mathbb{P}(X \in S)$, then $\mathbb{P}(h(Y) = 1) \ge \mathbb{P}(Y \in S)$.*
2. *If $S = \left\{ z \in \mathbb{R}^d : \frac{\mu_Y(z)}{\mu_X(z)} \ge t \right\}$ for some $t > 0$ and $\mathbb{P}(h(X) = 1) \le \mathbb{P}(X \in S)$, then $\mathbb{P}(h(Y) = 1) \le \mathbb{P}(Y \in S)$.*

*Proof.* Without loss of generality, we assume that $h$ is random and write $h(1|x)$ for the probability that $h(x) = 1$.

First we prove part 1. We denote the complement of $S$ as $S^c$.

$$\mathbb{P}(h(Y)=1) - \mathbb{P}(Y \in S) \tag{A.1}$$
$$= \int_{\mathbb{R}^d} h(1|z)\,\mu_Y(z)dz - \int_S \mu_Y(z)dz \tag{A.2}$$
$$= \left[\int_{S^c} h(1|z)\mu_Y(z)dz + \int_S h(1|z)\mu_Y(z)dz\right] - \left[\int_S h(1|z)\mu_Y(z)dz + \int_S h(0|z)\mu_Y(z)dz\right] \tag{A.3}$$
$$= \int_{S^c} h(1|z)\mu_Y(z)dz - \int_S h(0|z)\mu_Y(z)dz \tag{A.4}$$
$$\geq t\left[\int_{S^c} h(1|z)\mu_X(z)dz - \int_S h(0|z)\mu_X(z)\right] \tag{A.5}$$
$$= t\left[\int_{S^c} h(1|z)\mu_X(z)dz + \int_S h(1|z)\mu_X(z)dz - \int_S h(1|z)\mu_X(z)dz - \int_S h(0|z)\mu_X(z)\right] \tag{A.6}$$
$$= t\left[\int_{\mathbb{R}^d} h(1|z)\mu_X(z)dz - \int_S \mu_X(z)dz\right] \tag{A.7}$$
$$= t\left[\mathbb{P}(h(X)=1) - \mathbb{P}(X \in S)\right] \tag{A.8}$$
$$\geq 0 \tag{A.9}$$

The inequality in the middle is due to the fact that $\mu_Y(z) \leq t\,\mu_X(z)\ \forall z \in S$ and $\mu_Y(z) > t\,\mu_X(z)\ \forall z \in S^c$. The inequality at the end is because both terms in the product are non-negative by assumption.

The proof for part 2 is virtually identical, except both "$\geq$" become "$\leq$." □

**Remark: connection to statistical hypothesis testing.** Part 2 of Lemma Lemma A.1.1 is known in the field of statistical hypothesis testing as the Neyman-Pearson Lemma [Neyman and Pearson, 1933]. The hypothesis testing problem is this: we are given a sample that comes from one of two distributions over $\mathbb{R}^d$: either the null distribution $X$ or the alternative distribution $Y$. We would like to identify which distribution the sample came from. It is worse to say "$Y$" when the true answer is "$X$" than to say "$X$" when the true answer is "$Y$." Therefore we seek a (potentially randomized) procedure $h : \mathbb{R}^d \to \{0, 1\}$ which returns "$Y$" when the sample really came from $X$ with probability no greater than some failure rate $\alpha$. In particular, out of all such rules $h$, we would like the *uniformly most powerful* one $h^*$, i.e. the rule which is most likely to correctly say "$Y$" when the sample really came from $Y$. Neyman and Pearson [1933] showed that $h^*$ is the rule which returns "$Y$" deterministically on the set $S^* = \{z \in \mathbb{R}^d : \frac{\mu_Y(z)}{\mu_X(z)} \geq t\}$ for whichever $t$ makes $\mathbb{P}(X \in S^*) = \alpha$. In other words, to state this in a form that looks like Part 2 of Lemma Lemma A.1.1: if $h$ is a different rule with $\mathbb{P}(h(X) = 1) \leq \alpha$, then $h^*$ is more powerful than $h$, i.e. $\mathbb{P}(h(Y) = 1) \leq \mathbb{P}(Y \in S^*)$.

Now we state the special case of Lemma Lemma A.1.1 for when $X$ and $Y$ are isotropic Gaussians.

**Lemma A.1.2 (Neyman-Pearson for Gaussians with different means).** *Let $X \sim \mathcal{N}(x, \sigma^2 I)$ and $Y \sim \mathcal{N}(x + \delta, \sigma^2 I)$. Let $h : \mathbb{R}^d \to \{0, 1\}$ be any deterministic or random function. Then:*

1. If $S = \left\{z \in \mathbb{R}^d : \delta^T z \leq \beta\right\}$ for some $\beta$ and $\mathbb{P}(h(X) = 1) \geq \mathbb{P}(X \in S)$, then $\mathbb{P}(h(Y) = 1) \geq \mathbb{P}(Y \in S)$
2. If $S = \left\{z \in \mathbb{R}^d : \delta^T z \geq \beta\right\}$ for some $\beta$ and $\mathbb{P}(h(X) = 1) \leq \mathbb{P}(X \in S)$, then $\mathbb{P}(h(Y) = 1) \leq \mathbb{P}(Y \in S)$

*Proof.* This lemma is the special case of Lemma Lemma A.1.1 when $X$ and $Y$ are isotropic Gaussians with means $x$ and $x + \delta$.

By Lemma Lemma A.1.1 it suffices to simply show that for any $\beta$, there is some $t > 0$ for which:

$$\{z : \delta^T z \leq \beta\} = \left\{z : \frac{\mu_Y(z)}{\mu_X(z)} \leq t\right\} \quad \text{and} \quad \{z : \delta^T z \geq \beta\} = \left\{z : \frac{\mu_Y(z)}{\mu_X(z)} \geq t\right\} \tag{A.10}$$

The likelihood ratio for this choice of $X$ and $Y$ turns out to be:

$$\frac{\mu_Y(z)}{\mu_X(z)} = \frac{\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^d (z_i - (x_i + \delta_i))^2\right)}{\exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^d (z_i - x_i)^2\right)} \tag{A.11}$$

$$= \exp\left(\frac{1}{2\sigma^2}\sum_{i=1}^d 2z_i\delta_i - \delta_i^2 - 2x_i\delta_i\right) \tag{A.12}$$

$$= \exp(a\delta^T z + b) \tag{A.13}$$

where $a > 0$ and $b$ are constants w.r.t $z$, specifically $a = \frac{1}{\sigma^2}$ and $b = \frac{-(2\delta^T x + \|\delta\|^2)}{2\sigma^2}$.

Therefore, given any $\beta$ we may take $t = \exp(a\beta + b)$, noticing that

$$\delta^T z \leq \beta \iff \exp(a\delta^T z + b) \leq t \tag{A.14}$$

$$\delta^T z \geq \beta \iff \exp(a\delta^T z + b) \geq t \tag{A.15}$$

□

Finally, we prove Theorems 2.3.1 and 2.3.2.

**Theorem 2.3.1 (restated).** *Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function. Let $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$. Let $g(x) = \arg\max_c \mathbb{P}(f(x+\varepsilon) = c)$. Suppose that for a specific $x \in \mathbb{R}^d$, there exist $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0,1]$ such that:*

$$\mathbb{P}(f(x+\varepsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}(f(x+\varepsilon) = c) \tag{A.16}$$

*Then $g(x+\delta) = c_A$ for all $\|\delta\|_2 < R$, where*

$$R = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})) \tag{A.17}$$

*Proof.* To show that $g(x+\delta) = c_A$, it follows from the definition of $g$ that we need to show that

$$\mathbb{P}(f(x+\delta+\varepsilon) = c_A) > \max_{c_B \neq c_A} \mathbb{P}(f(x+\delta+\varepsilon) = c_B) \tag{A.18}$$

We will prove that $\mathbb{P}(f(x+\delta+\varepsilon) = c_A) > \mathbb{P}(f(x+\delta+\varepsilon) = c_B)$ for every class $c_B \neq c_A$. Fix one such class $c_B$ without loss of generality.

For brevity, define the random variables

$$X := x + \varepsilon = \mathcal{N}(x, \sigma^2 I) \tag{A.19}$$

$$Y := x + \delta + \varepsilon = \mathcal{N}(x+\delta, \sigma^2 I) \tag{A.20}$$

In this notation, we know from (Equation (A.16)) that

$$\mathbb{P}(f(X) = c_A) \geq \underline{p_A} \quad \text{and} \quad \mathbb{P}(f(X) = c_B) \leq \overline{p_B} \tag{A.21}$$

and our goal is to show that

$$\mathbb{P}(f(Y) = c_A) > \mathbb{P}(f(Y) = c_B) \tag{A.22}$$

Define the half-spaces:

$$A := \{z : \delta^T(z-x) \leq \sigma\|\delta\|\Phi^{-1}(\underline{p_A})\} \tag{A.23}$$

$$B := \{z : \delta^T(z-x) \geq \sigma\|\delta\|\Phi^{-1}(1-\overline{p_B})\} \tag{A.24}$$

Algebra (deferred to the end) shows that $\mathbb{P}(X \in A) = \underline{p_A}$. Therefore, by (Equation (A.21)) we know that $\mathbb{P}(f(X) = c_A) \geq \mathbb{P}(X \in A)$. Hence we may apply Lemma Lemma A.1.2 with $h(z) := \mathbf{1}[f(z) = c_A]$ to conclude:

$$\mathbb{P}(f(Y) = c_A) \geq \mathbb{P}(Y \in A) \tag{A.25}$$

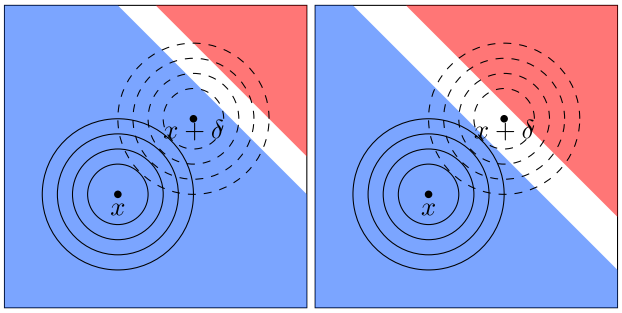


Figure A.1: Illustration of the proof of Theorem 2.3.1. The solid line concentric circles are the density level sets of $X := x + \varepsilon$; the dashed line concentric circles are the level sets of $Y := x + \delta + \varepsilon$. The set $A$ is in blue and the set $B$ is in red. The figure on the left depicts a situation where $\mathbb{P}(Y \in A) > \mathbb{P}(Y \in B)$, and hence $g(x + \delta)$ may equal $c_A$. The figure on the right depicts a situation where $\mathbb{P}(Y \in A) < \mathbb{P}(Y \in B)$ and hence $g(x + \delta) \neq c_A$.

Similarly, algebra shows that $\mathbb{P}(X \in B) = \overline{p_B}$. Therefore, by (Equation (A.21)) we know that $\mathbb{P}(f(X) = c_B) \leq \mathbb{P}(X \in B)$. Hence we may apply Lemma Lemma A.1.2 with $h(z) := \mathbf{1}[f(z) = c_B]$ to conclude:

$$\mathbb{P}(f(Y) = c_B) \leq \mathbb{P}(Y \in B) \tag{A.26}$$

To guarantee (Equation (A.22)), we see from (Equation (A.25), Equation (A.26)) that it suffices to show that $\mathbb{P}(Y \in A) > \mathbb{P}(Y \in B)$, as this step completes the chain of inequalities

$$\mathbb{P}(f(Y) = c_A) \geq \mathbb{P}(Y \in A) > \mathbb{P}(Y \in B) \geq \mathbb{P}(f(Y) = c_B) \tag{A.27}$$

We can compute the following:

$$\mathbb{P}(Y \in A) = \Phi\left(\Phi^{-1}(\underline{p_A}) - \frac{\|\delta\|}{\sigma}\right) \tag{A.28}$$

$$\mathbb{P}(Y \in B) = \Phi\left(\Phi^{-1}(\overline{p_B}) + \frac{\|\delta\|}{\sigma}\right) \tag{A.29}$$

Finally, algebra shows that $\mathbb{P}(Y \in A) > \mathbb{P}(Y \in B)$ if and only if:

$$\|\delta\| < \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})) \tag{A.30}$$

which recovers the theorem statement. □

We now restate and prove Theorem 2.3.2, which shows that the bound in Theorem 2.3.1 is tight. The assumption below in Theorem 2.3.2 that $\underline{p_A} + \overline{p_B} \leq 1$ is mild: given any $\underline{p_A}$ and $\overline{p_B}$ which do not satisfy this condition, one could have always redefined $\overline{p_B} \leftarrow 1 - \underline{p_A}$ to obtain a Theorem 2.3.1 guarantee with a larger certified radius, so there is no reason to invoke Theorem 2.3.1 unless $\underline{p_A} + \overline{p_B} \leq 1$.

**Theorem 2 (restated).** *Assune $\underline{p_A} + \overline{p_B} \leq 1$. For any perturbation $\delta \in \mathbb{R}^d$ with $\|\delta\|_2 > R$, there exists a base classifier $f^*$ consistent with the observed class probabilities (Equation (A.16)) such that if $f^*$ is the base classifier for $g$, then $g(x+\delta) \neq c_A$.*

*Proof.* We re-use notation from the preceding proof.

Pick any class $c_B$ arbitrarily. Define $A$ and $B$ as above, and consider the function

$$f^*(x) := \begin{cases} c_A & \text{if } x \in A \\ c_B & \text{if } x \in B \\ \text{other classes} & \text{otherwise} \end{cases} \tag{A.31}$$

This function is well-defined, since $A \cap B = \emptyset$ provided that $\underline{p_A} + \overline{p_B} \leq 1$.

By construction, the function $f^*$ satisfies (Equation (A.16)) with equalities, since

$$\mathbb{P}(f^*(x+\varepsilon) = c_A) = \mathbb{P}(X \in A) = \underline{p_A} \qquad \mathbb{P}(f^*(x+\varepsilon) = c_B) = \mathbb{P}(X \in B) = \overline{p_B} \tag{A.32}$$

It follows from (Equation (A.28)) and (Equation (A.29)) that

$$\mathbb{P}(Y \in A) < \mathbb{P}(Y \in B) \iff \|\delta\|_2 > R \tag{A.33}$$

By assumption, $\|\delta\|_2 > R$, so $\mathbb{P}(Y \in A) < \mathbb{P}(Y \in B)$, or equivalently,

$$\mathbb{P}(f^*(x+\delta+\varepsilon) = c_A) < \mathbb{P}(f^*(x+\delta+\varepsilon) = c_B) \tag{A.34}$$

Therefore, if $f^*$ is the base classifier for $g$, then $g(x+\delta) \neq c_A$. $\square$

## Deferred Algebra

**Claim A.1.3.** $\mathbb{P}(X \in A) = \underline{p_A}$

*Proof.* Recall that $X \sim \mathcal{N}(x, \sigma^2 I)$ and $A = \{z : \delta^T(z - x) \leq \sigma\|\delta\|\Phi^{-1}(\underline{p_A})\}$.

$$\begin{aligned}
\mathbb{P}(X \in A) &= \mathbb{P}(\delta^T(X - x) \leq \sigma\|\delta\|\Phi^{-1}(\underline{p_A})) && \text{(A.35)}\\
&= \mathbb{P}(\delta^T \mathcal{N}(0, \sigma^2 I) \leq \sigma\|\delta\|\Phi^{-1}(\underline{p_A})) && \text{(A.36)}\\
&= \mathbb{P}(\sigma\|\delta\|Z \leq \sigma\|\delta\|\Phi^{-1}(\underline{p_A})) && (Z \sim \mathcal{N}(0,1))\\
&= \Phi(\Phi^{-1}(\underline{p_A})) && \text{(A.37)}\\
&= \underline{p_A} && \text{(A.38)}
\end{aligned}$$

□

**Claim A.1.4.** $\mathbb{P}(X \in B) = \overline{p_B}$

*Proof.* Recall that $X \sim \mathcal{N}(x, \sigma^2 I)$ and $B = \{z : \delta^T(z - x) \leq \sigma\|\delta\|\Phi^{-1}(1 - \overline{p_B})\}$.

$$\begin{aligned}
\mathbb{P}(X \in A) &= \mathbb{P}(\delta^T(X - x) \geq \sigma\|\delta\|\Phi^{-1}(1 - \overline{p_B})) && \text{(A.39)}\\
&= \mathbb{P}(\delta^T \mathcal{N}(0, \sigma^2 I) \geq \sigma\|\delta\|\Phi^{-1}(1 - \overline{p_B})) && \text{(A.40)}\\
&= \mathbb{P}(\sigma\|\delta\|Z \geq \sigma\|\delta\|\Phi^{-1}(1 - \overline{p_B})) && (Z \sim \mathcal{N}(0,1))\\
&= \mathbb{P}(Z \geq \Phi^{-1}(1 - \overline{p_B})) && \text{(A.41)}\\
&= 1 - \Phi(\Phi^{-1}(1 - \overline{p_B})) && \text{(A.42)}\\
&= \overline{p_B} && \text{(A.43)}
\end{aligned}$$

□

**Claim A.1.5.** $\mathbb{P}(Y \in A) = \Phi\left(\Phi^{-1}(\underline{p_A}) - \frac{\|\delta\|}{\sigma}\right)$

*Proof.* Recall that $Y \sim \mathcal{N}(x + \delta, \sigma^2 I)$ and $A = \{z : \delta^T(z - x) \leq \sigma\|\delta\|\Phi^{-1}(\underline{p_A})\}$.

$$\begin{aligned}
\mathbb{P}(Y \in A) &= \mathbb{P}(\delta^T(Y - x) \leq \sigma\|\delta\|\Phi^{-1}(\underline{p_A})) && \text{(A.44)}\\
&= \mathbb{P}(\delta^T \mathcal{N}(0, \sigma^2 I) + \|\delta\|^2 \leq \sigma\|\delta\|\Phi^{-1}(\underline{p_A})) && \text{(A.45)}\\
&= \mathbb{P}(\sigma\|\delta\|Z \leq \sigma\|\delta\|\Phi^{-1}(\underline{p_A}) - \|\delta\|^2) && (Z \sim \mathcal{N}(0,1))\\
&= \mathbb{P}\left(Z \leq \Phi^{-1}(\underline{p_A}) - \frac{\|\delta\|}{\sigma}\right) && \text{(A.46)}\\
&= \Phi\left(\Phi^{-1}(\underline{p_A}) - \frac{\|\delta\|}{\sigma}\right) && \text{(A.47)}
\end{aligned}$$

□

**Claim A.1.6.** $\mathbb{P}(Y \in B) = \Phi\left(\Phi^{-1}(\overline{p_B}) + \frac{\|\delta\|}{\sigma}\right)$

*Proof.* Recall that $Y \sim \mathcal{N}(x+\delta, \sigma^2 I)$ and $B = \{z : \delta^T(z-x) \geq \sigma\|\delta\|\Phi^{-1}(1-\overline{p_B})\}$.

$$\begin{aligned}
\mathbb{P}(Y \in B) &= \mathbb{P}(\delta^T(Y-x) \geq \sigma\|\delta\|\Phi^{-1}(1-\overline{p_B})) && \text{(A.48)}\\
&= \mathbb{P}(\delta^T\mathcal{N}(0,\sigma^2 I) + \|\delta\|^2 \geq \sigma\|\delta\|\Phi^{-1}(1-\overline{p_B})) && \text{(A.49)}\\
&= \mathbb{P}(\sigma\|\delta\|Z + \|\delta\|^2 \geq \sigma\|\delta\|\Phi^{-1}(1-\overline{p_B})) && (Z \sim \mathcal{N}(0,1))\\
&= \mathbb{P}\left(Z \geq \Phi^{-1}(1-\overline{p_B}) - \frac{\|\delta\|}{\sigma}\right) && \text{(A.50)}\\
&= \mathbb{P}\left(Z \leq \Phi^{-1}(\overline{p_B}) + \frac{\|\delta\|}{\sigma}\right) && \text{(A.51)}\\
&= \Phi\left(\Phi^{-1}(\overline{p_B}) + \frac{\|\delta\|}{\sigma}\right) && \text{(A.52)}
\end{aligned}$$

□

## A.2 Smoothing a Two-Class Linear Classifier

In this appendix, we analyze what happens when the base classifier $f$ is a two-class linear classifier $f(x) = \text{sign}(w^T x + b)$. To match the definition of $g$, we take $\text{sign}(\cdot)$ to be undefined when its argument is zero.

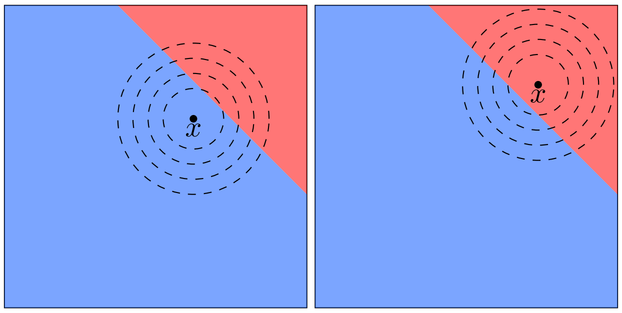


Figure A.2: Illustration of Proposition Proposition A.2.1. A binary linear classifier $f(x) = \text{sign}(w^T x + b)$ partitions $\mathbb{R}^d$ into two half-spaces, drawn here in blue and red. An isotropic Gaussian $\mathcal{N}(x, \sigma^2 I)$ will put more mass on whichever half-space its center $x$ lies in: in the figure on the left, $x$ is in the blue half-space and $\mathcal{N}(x, \sigma^2 I)$ puts more mass on the blue than on red. In the figure on the right, $x$ is in the red half-space and $\mathcal{N}(x, \sigma^2 I)$ puts more mass on red than on blue. Since the smoothed classifier's prediction $g(x)$ is defined to be whichever half-space $\mathcal{N}(x, \sigma^2 I)$ puts more mass in, and the base classifier's prediction $f(x)$ is defined to be whichever half-space $x$ is in, we have that $g(x) = f(x)$ for all $x$.

Our first result is that when $f$ is a two-class linear classifier, the smoothed classifier $g$ is identical to the base classifier $f$.

**Proposition A.2.1.** *If $f$ is a two-class linear classifier $f(x) = sign(w^T x + b)$, and $g$ is the smoothed version of $f$ with any $\sigma$, then $g(x) = f(x)$ for any $x$ (where $f$ is defined).*

*Proof.* From the definition of $g$,

$$\begin{aligned}
g(x) = 1 &\iff \mathbb{P}_\varepsilon(f(x+\varepsilon) = 1) > \frac{1}{2} && (\varepsilon \sim \mathcal{N}(0, \sigma^2 I)) \\
&\iff \mathbb{P}_\varepsilon\left(\text{sign}(w^T(x+\varepsilon) + b) = 1\right) > \frac{1}{2} && \text{(A.53)} \\
&\iff \mathbb{P}_\varepsilon\left(w^T x + w^T \varepsilon + b \geq 0\right) > \frac{1}{2} && \text{(A.54)} \\
&\iff \mathbb{P}\left(\sigma\|w\| Z \geq -w^T x - b\right) > \frac{1}{2} && (Z \sim \mathcal{N}(0,1)) \\
&\iff \mathbb{P}\left(Z \leq \frac{w^T x + b}{\sigma\|w\|}\right) > \frac{1}{2} && \text{(A.55)} \\
&\iff \frac{w^T x + b}{\sigma\|w\|} > 0 && \text{(A.56)} \\
&\iff w^T x + b > 0 && \text{(A.57)} \\
&\iff f(x) = 1 && \text{(A.58)}
\end{aligned}$$

□

A similar calculation shows that $g(x) = -1 \iff f(x) = -1$.

A two-class linear classifier $f(x) = \text{sign}(w^T x + b)$ is already certifiable: the distance from any point $x$ to the decision boundary is $(w^T x + b)/\|w\|_2$, and no distance with $\ell_2$ norm strictly less than this distance can possibly change $f$'s prediction. Let $g$ be a smoothed version of $f$. By Proposition Proposition A.2.1, $g$ is identical to $f$, so it follows that $g$ is truly robust around any input $x$ within the $\ell_2$ radius $(w^T x + b)/\|w\|_2$. We now show that Theorem 2.3.1 will certify this radius, rather than a smaller, over-conservative radius.

**Proposition A.2.2.** *If $f$ is a two-class linear classifier $f(x) = sign(w^T x + b)$, and $g$ is the smoothed version of $f$ with any $\sigma$, then invoking Theorem 2.3.1 at any $x$ (where $f$ is defined) with $\underline{p_A} = p_A$ and $\overline{p_B} = p_B$ will yield the certified radius $R = \frac{|w^T x + b|}{\|w\|}$.*

*Proof.* In binary classification, $p_A = 1 - p_B$, so Theorem 2.3.1 returns $R = \sigma\Phi^{-1}(\underline{p_A})$.

We have:

$$\begin{aligned}
p_A &= \mathbb{P}_\varepsilon(f(x+\varepsilon) = g(x)) && \text{(A.59)} \\
&= \mathbb{P}_\varepsilon(\text{sign}(w^T(x+\varepsilon) + b) = \text{sign}(w^T x + b)) \\
& && \text{(By Proposition Proposition A.2.1, } g(x) = f(x)) \\
&= \mathbb{P}_\varepsilon(\text{sign}(w^T x + \sigma\|w\| Z + b) = \text{sign}(w^T x + b)) && \text{(A.60)}
\end{aligned}$$

There are two cases: if $w^T x + b > 0$, then

$$\begin{aligned} p_A &= \mathbb{P}_\varepsilon(w^T x + \sigma\|w\| Z + b > 0) && \text{(A.61)} \\ &= \mathbb{P}_\varepsilon\left(Z > \frac{-w^T x - b}{\sigma\|w\|}\right) && \text{(A.62)} \\ &= \mathbb{P}_\varepsilon\left(Z < \frac{w^T x + b}{\sigma\|w\|}\right) && \text{(A.63)} \\ &= \Phi\left(\frac{w^T x + b}{\sigma\|w\|}\right) && \text{(A.64)} \end{aligned}$$

On the other hand, if $w^T x + b < 0$, then

$$\begin{aligned} p_A &= \mathbb{P}_\varepsilon(w^T x + \sigma\|w\| Z + b < 0) && \text{(A.65)} \\ &= \mathbb{P}_\varepsilon\left(Z < \frac{-w^T x - b}{\sigma\|w\|}\right) && \text{(A.66)} \\ &= \Phi\left(\frac{-w^T x - b}{\sigma\|w\|}\right) && \text{(A.67)} \end{aligned}$$

In either case, we have:

$$p_A = \Phi\left(\frac{|w^T x + b|}{\sigma\|w\|}\right) \tag{A.68}$$

Therefore, the bound in Theorem 1 returns a radius of

$$\begin{aligned} R &= \sigma\Phi^{-1}(p_A) && \text{(A.69)} \\ &= \frac{|w^T x + b|}{\|w\|} && \text{(A.70)} \end{aligned}$$

□

The previous two propositions imply that when $f$ is a two-class linear classifier, the Theorem 2.3.1 bound is "tight" in the sense that there always exists a class-changing perturbation just beyond the certified radius.[1]

[1]Note that this is a different sense of "tight" than the sense in which Theorem 2.3.2 proves that Theorem 2.3.1 is tight. Theorem 2.3.2 proves that for any fixed perturbation $\delta$ outside the radius certified by Theorem 2.3.1, there exists a base classifier $f$ for which $g(x + \delta) \neq g(x)$. In contrast, Proposition Proposition A.2.3 proves that for any fixed binary linear base classifier $f$, there exists a perturbation $\delta$ just outside the radius certified by Theorem 2.3.1 for which $g(x + \delta) \neq g(x)$.

**Proposition A.2.3.** *Let $f$ be a two-class linear classifier $f(x) = sign(w^T x + b)$, let $g$ be the smoothed version of $f$ for some $\sigma$, let $x$ be any point (where $f$ is defined), and let $R$ be the radius certified around $x$ by Theorem 2.3.1. Then for any radius $r > R$, there exists a perturbation $\delta$ with $\|\delta\|_2 = r$ for which $g(x + \delta) \neq g(x)$.*

*Proof.* By Proposition Proposition A.2.1 it suffices to show that there exists some perturbation $\delta$ with $\|\delta\|_2 = r$ for which $f(x + \delta) \neq f(x)$.

By Proposition Proposition A.2.2, we know that $R = \frac{|w^T x + b|}{\|w\|_2}$.

If $w^T x + b > 0$, consider the perturbation $\delta = -\frac{w}{\|w\|_2} r$. This perturbation satisfies $\|\delta\|_2 = r$ and

$$\begin{aligned} w^T(x+\delta) + b &= w^T x + b + w^T \delta && \text{(A.71)} \\ &= w^T x + b - \|w\|_2 r && \text{(A.72)} \\ &< w^T x + b - \|w\|_2 R && \text{(A.73)} \\ &= w^T x + b - |w^T x + b| && \text{(A.74)} \\ &= w^T x + b - (w^T x + b) && \text{(A.75)} \\ &= 0 && \text{(A.76)} \end{aligned}$$

implying that $f(x + \delta) = -1$.

Likewise, if $w^T x + b < 0$, then consider the perturbation $\delta = \frac{w}{\|w\|_2} r$. This perturbation satisfies $\|\delta\|_2 = r$ and $f(x + \delta) = -1$.

□

This special property of two-class linear classifiers is not true in general. In fact, it is possible to construct situations where $g$'s prediction around some point $x_0$ is robust at radius $\infty$, yet Theorem 2.3.1 only certifies a radius of $\tau$, where $\tau$ is arbitrarily close to zero.

**Proposition A.2.4.** *For any $\tau > 0$, there exists a base classifier $f$ and an input $x_0$ for which the corresponding smoothed classifier $g$ is robust around $x_0$ at radius $\infty$, yet Theorem 2.3.1 only certifies a radius of $\tau$ around $x_0$.*

*Proof.* Let $t = -\Phi^{-1}(\frac{1}{2}\Phi(\tau))$ and consider the following base classifier:

$$f(x) = \begin{cases} 1 & \text{if } x < -t \\ -1 & \text{if } -t \leq x \leq t \\ 1 & \text{if } x > t \end{cases} \tag{A.77}$$

Let $g$ be the smoothed version of $f$ with $\sigma = 1$. We will show that $g(x) = 1$ everywhere, implying that $g$'s prediction is robust around $x_0 = 0$ with radius $\infty$. Yet Theorem 2.3.1 only certifies a radius of $\tau$ around $x_0$.

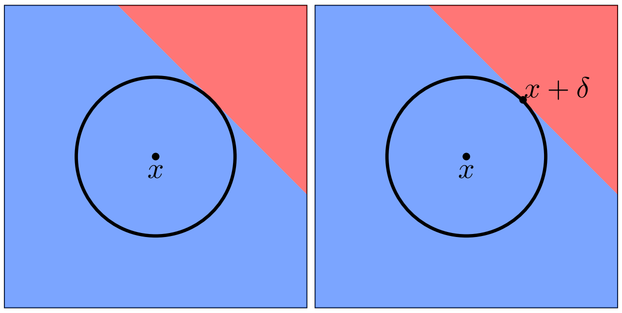


Figure A.3: **Left**: Illustration of of Proposition Proposition A.2.2. The red/blue half-spaces are the decision regions of both the base classifier $f$ and the smoothed classifier $g$. (Since the base classifier is binary linear, $g = f$ everywhere.) The black circle is the robustness radius $R$ certified by Theorem 2.3.1. **Right**: Illustration of Proposition Proposition A.2.3. For any $r > R$, there exists a perturbation $\delta$ with $\|\delta\|_2 = r$ for which $g(x + \delta) \neq g(x)$.

Let $Z \sim \mathcal{N}(0, 1)$. For any $x$, we have:

$$\begin{aligned}
\mathbb{P}(f(x+\varepsilon) = -1) &= \mathbb{P}(-t \leq x + \varepsilon \leq t) && \text{(A.78)}\\
&= \mathbb{P}[-t - x \leq Z \leq t - x] && \text{(A.79)}\\
&\leq \mathbb{P}[-t \leq Z \leq t]\\
&\text{(apply Lemma Lemma A.2.5 below with } \ell = -t - x)\\
&= 1 - 2\Phi(-t) && \text{(A.80)}\\
&= 1 - \Phi(\tau) && \text{(A.81)}\\
&< \frac{1}{2}. && \text{(A.82)}
\end{aligned}$$

Therefore, $g(x) = 1$ for all $x$.
Meanwhile, at $x_0 = 0$, we have:

$$\begin{aligned}
\mathbb{P}(f(x_0+\varepsilon) = 1) &= \mathbb{P}(f(\varepsilon) = 1) && \text{(A.83)}\\
&= \mathbb{P}(Z < -t \text{ or } Z > t) && \text{(A.84)}\\
&= 2\Phi(-t) && \text{(A.85)}\\
&= \Phi(\tau), && \text{(A.86)}
\end{aligned}$$

so by Theorem 2.3.1, the certified radius around $x_0$ is $R = \tau$.

□

The proof of Proposition Proposition A.2.4 employed the following lemma, which formalizes the visually obvious fact that out of all intervals of some fixed width $2t$, the interval with maximal mass under the standard normal distribution $Z$ is the interval $[-t, t]$.

**Lemma A.2.5.** *Let $Z \sim \mathcal{N}(0,1)$. For any $\ell \in \mathbb{R}$, $t > 0$, we have $\mathbb{P}(\ell \leq Z \leq \ell + 2t) \leq \mathbb{P}(-t \leq Z \leq t)$.*

*Proof.* Let $\phi$ be the PDF of the standard normal distribution. Since $\phi$ is symmetric about the origin (i.e. $\phi(x) = \phi(-x)\ \forall x$),

$$\mathbb{P}(-t \leq Z \leq t) = 2\int_0^t \phi(x)dx. \tag{A.87}$$

There are two cases to consider:

**Case 1:** The interval $[\ell, \ell + 2t]$ is entirely positive, i.e. $\ell \geq 0$, or $[\ell, \ell + 2t]$ is entirely negative, i.e. $\ell + 2t \leq 0$.

First, we use the fact that $\phi$ is symmetric about the origin to rewrite $\mathbb{P}(\ell \leq Z \leq \ell + 2t)$ as the probability that $Z$ falls in a non-negative interval $[a, a + 2t]$ for some $a$.

Specifically, if $\ell \geq 0$, then let $a = \ell$. Else, if $\ell + 2t \leq 0$, then let $a = -(\ell + 2t)$. We therefore have:

$$\mathbb{P}(\ell \leq Z \leq \ell + 2t) = \mathbb{P}(a \leq Z \leq a + 2t). \tag{A.88}$$

Therefore:

$$\mathbb{P}(-t \leq Z \leq t) - \mathbb{P}(\ell \leq Z \leq \ell + 2t) \tag{A.89}$$

$$= \int_0^t \phi(x)dx - \int_a^{a+t} \phi(x)dx + \int_0^t \phi(x)dx - \int_{a+t}^{a+2t} \phi(x)dx \tag{A.90}$$

$$= \int_a^{a+t} \phi(x-a)dx - \int_a^{a+t} \phi(x)dx + \int_{a+t}^{a+2t} \phi(x-a-t)dx - \int_{a+t}^{a+2t} \phi(x)dx \tag{A.91}$$

$$= \int_a^{a+t} [\phi(x-a) - \phi(x)]\, dx + \int_{a+t}^{a+2t} [\phi(x-a-t) - \phi(x)]\, dx \tag{A.92}$$

$$\geq \int_a^{a+t} 0\ dx + \int_{a+t}^{a+2t} 0\ dx \tag{A.93}$$

$$= 0 \tag{A.94}$$

where the inequality is because $\phi$ is monotonically decreasing on $[0, \infty)$.

**Case 2:** $I$ is partly positive, partly negative, i.e. $\ell < 0 < \ell + 2t$.

First, we use the fact that $\phi$ is symmetric about the origin to rewrite $\mathbb{P}(\ell \leq Z \leq \ell + 2t)$ as the sum of the probabilities that $Z$ falls in two non-negative intervals $[0, a]$ and $[0, b]$ for some $a, b$.

Specifically, let $a = \min(-\ell, \ell + 2t)$ and $b = \max(-\ell, \ell + 2t)$. We therefore have:

$$\mathbb{P}(\ell \leq Z \leq \ell + 2t) = \mathbb{P}(0 \leq Z \leq a) + \mathbb{P}(0 \leq Z \leq b). \tag{A.95}$$

Note that by construction, $a + b = 2t$, and $0 \leq a \leq t$ and $t \leq b \leq 2t$.

We have:

$$\begin{aligned}
&\mathbb{P}(-t \leq Z \leq t) - \mathbb{P}(\ell \leq Z \leq \ell + 2t) && \text{(A.96)}\\
&= \left[\int_0^t \phi(x)dx - \int_0^a \phi(x)dx\right] - \left[\int_0^b \phi(x)dx - \int_0^t \phi(x)dx\right] && \text{(A.97)}\\
&= \int_a^t \phi(x)dx - \int_t^b \phi(x)dx && \text{(A.98)}\\
&= \int_a^t \phi(x)dx - \int_t^{2t-a} \phi(x)dx && \text{(A.99)}\\
&= \int_a^t \phi(x)dx - \int_a^t \phi(x - a + t)dx && \text{(A.100)}\\
&= \int_a^t (\phi(x) - \phi(x - a + t))dx && \text{(A.101)}\\
&\geq \int_a^t 0\,dx && \text{(A.102)}\\
&= 0 && \text{(A.103)}
\end{aligned}$$

where the inequality is again because $\phi$ is monotonically decreasing on $[0, \infty)$.

□

## A.3 Practical Algorithms

In this appendix, we elaborate on the prediction and certification algorithms described in Section Section 2.3.2. The pseudocode in Section Section 2.3.2 makes use of several helper functions:

- SAMPLEUNDERNOISE($f$, $x$, num, $\sigma$) works as follows:
  1. Draw num samples of noise, $\varepsilon_1 \dots \varepsilon_{\text{num}} \sim \mathcal{N}(0, \sigma^2 I)$.
  2. Run the noisy images through the base classifier $f$ to obtain the predictions $f(x + \varepsilon_1), \dots, f(x + \varepsilon_{\text{num}})$.
  3. Return the counts for each class, where the count for class $c$ is defined as $\sum_{i=1}^{\text{num}} \mathbf{1}[f(x + \varepsilon_i) = c]$.
- BINOMPVALUE($n_A$, $n_A + n_B$, $p$) returns the p-value of the two-sided hypothesis test that $n_A \sim \text{Binomial}(n_A + n_B, p)$. Using `scipy.stats.binom_test`, this can be implemented as: `binom_test(nA, nA + nB, p)`.
- LOWERCONFBOUND($k$, $n$, $1 - \alpha$) returns a one-sided $(1 - \alpha)$ lower confidence interval for the Binomial parameter $p$ given that $k \sim \text{Binomial}(n, p)$. In other words, it returns some number $\underline{p}$ for which $\underline{p} \leq p$ with probability at least $1 - \alpha$ over the sampling of $k \sim \text{Binomial}(n, p)$. Following Lecuyer et al. [2019], we chose to use the Clopper-Pearson confidence interval, which inverts the Binomial CDF [Clopper and Pearson, 1934]. Using `statsmodels.stats.proportion.proportion_confint`, this can be implemented as

  ```
  proportion_confint(k, n, alpha=2*alpha, method="beta")[0]
  ```

### A.3.1 Prediction

The randomized algorithm given in pseudocode as PREDICT leverages the hypothesis test given in Hung and Fithian [2019] for identifying the top category of a multinomial distribution. PREDICT has one tunable hyperparameter, $\alpha$. When $\alpha$ is small, PREDICT abstains frequently but rarely returns the wrong class. When $\alpha$ is large, PREDICT usually makes a prediction, but may often return the wrong class.

We now prove that with high probability, PREDICT will either return $g(x)$ or abstain.

**Proposition Proposition 2.3.3 (restated).** *With probability at least $1 - \alpha$ over the randomness in* PREDICT, PREDICT *will either abstain or return $g(x)$. (Equivalently: the probability that* PREDICT *returns a class other than $g(x)$ is at most $\alpha$.)*

*Proof.* For notational convenience, define $p_c = \mathbb{P}(f(x+\varepsilon) = c)$. Let $c_A = \max_c p_c$.

Notice that by definition, $g(x) = c_A$.

We can describe the randomized procedure PREDICT as follows:

1. Sample a vector of class counts $\{n_c\}_{c\in\mathcal{Y}}$ from Multinomial$(\{p_c\}_{c\in\mathcal{Y}}, n)$.
2. Let $\widehat{c}_A = \arg\max_c n_c$ be the class whose count is largest. Let $n_A$ and $n_B$ be the largest count and the second-largest count, respectively.
3. If the p-value of the two-sided hypothesis test that $n_A$ is drawn from Binom $\left(n_A + n_B, \frac{1}{2}\right)$ is less than $\alpha$, then return $\widehat{c}_A$. Else, abstain.

The quantities $c_A$ and the $p_c$'s are fixed but unknown, while the quantities $\widehat{c}_A$, the $n_c$'s, $n_A$, and $n_B$ are random.

We'd like to prove that the probability that PREDICT returns a class other than $c_A$ is at most $\alpha$. PREDICT returns a class other than $c_A$ if and only if (1) $\widehat{c}_A \neq c_A$ and (2) PREDICT does not abstain.

We have:

$$\mathbb{P}(\text{PREDICT returns class } \neq c_A) = \mathbb{P}(\widehat{c}_A \neq c_A, \text{PREDICT does not abstain}) \tag{A.104}$$

$$= \mathbb{P}(\widehat{c}_A \neq c_A)\ \mathbb{P}(\text{PREDICT does not abstain}|\widehat{c}_A \neq c_A) \tag{A.105}$$

$$\leq \mathbb{P}(\text{PREDICT does not abstain}|\widehat{c}_A \neq c_A) \tag{A.106}$$

Recall that PREDICT does not abstain if and only if the p-value of the two-sided hypothesis test that $n_A$ is drawn from Binom$(n_A + n_B, \frac{1}{2})$ is less than $\alpha$. Theorem 1 in Hung and Fithian [2019] proves that the conditional probability that this event occurs given that $\widehat{c}_A \neq c_A$ is exactly $\alpha$. That is,

$$\mathbb{P}(\text{PREDICT does not abstain}|\widehat{c}_A \neq c_A) = \alpha \tag{A.107}$$

Therefore, we have:

$$\mathbb{P}(\text{PREDICT returns class } \neq c_A) \leq \alpha \tag{A.108}$$

□

### A.3.2 Certification

The certification task is: given some input $x$ and a randomized smoothing classifier described by $(f, \sigma)$, return both (1) the prediction $g(x)$ and (2) a radius $R$ in which this prediction is certified robust. This task requires identifying the class $c_A$ with maximal weight in $f(x + \varepsilon)$, estimating a lower bound $\underline{p_A}$ on $p_A := \mathbb{P}(f(x +$

$\varepsilon) = c_A)$ and estimating an upper bound $\overline{p_B}$ on $p_B := \max_{c \neq c_A} \mathbb{P}(f(x+\varepsilon) = c)$ (Figure Figure 2.1).

Suppose for simplicity that we already knew $c_A$ and needed to obtain $\underline{p_A}$. We could collect $n$ samples of $f(x+\varepsilon)$, count how many times $f(x+\varepsilon) = c_A$, and use a Binomial confidence interval to obtain a lower bound on $p_A$ that holds with probability at least $1-\alpha$ over the $n$ samples.

However, estimating $\underline{p_A}$ and $\overline{p_B}$ while simultaneously identifying the top class $c_A$ is a little bit tricky, statistically speaking. We propose a simple two-step procedure. First, use $n_0$ samples from $f(x+\varepsilon)$ to take a guess $\widehat{c}_A$ at the identity of the top class $c_A$. In practice we observed that $f(x+\varepsilon)$ tends to put most of its weight on the top class, so $n_0$ can be set very small. Second, use $n$ samples from $f(x+\varepsilon)$ to obtain some $\underline{p_A}$ and $\overline{p_B}$ for which $\underline{p_A} \leq p_A$ and $\overline{p_B} \geq p_B$ with probability at least $1-\alpha$. We observed that it is much more typical for the mass of $f(x+\varepsilon)$ not allocated to $c_A$ to be allocated entirely to one runner-up class than to be allocated uniformly over all remaining classes. Therefore, the quantity $1 - \underline{p_A}$ is a reasonably tight upper bound on $p_B$. Hence, we simply set $\overline{p_B} = 1 - \underline{p_A}$, so our bound becomes

$$R = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(1 - \underline{p_A})) \tag{A.109}$$

$$= \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) + \Phi^{-1}(\underline{p_A})) \tag{A.110}$$

$$= \sigma\Phi^{-1}(\underline{p_A}) \tag{A.111}$$

The full procedure is described in pseudocode as CERTIFY. If $\underline{p_A} < \frac{1}{2}$, we abstain from making a certification; this can occur especially if $\widehat{c}_A \neq g(x)$, i.e. if we misidentify the top class using the first $n_0$ samples of $f(x+\varepsilon)$.

**Proposition Proposition 2.3.4 (restated).** *With probability at least $1-\alpha$ over the randomness in* CERTIFY, *if* CERTIFY *returns a class $\widehat{c}_A$ and a radius $R$ (i.e. does not abstain), then we have the robustness guarantee*

$$g(x+\delta) = \widehat{c}_A \quad \textit{whenever} \quad \|\delta\|_2 < R \tag{A.112}$$

*Proof.* From the contract of LOWERCONFBOUND, we know that with probability at least $1-\alpha$ over the sampling of $\varepsilon_1 \ldots \varepsilon_n$, we have $\underline{p_A} \leq \mathbb{P}[f(x+\varepsilon) = \widehat{c}_A]$. Notice that CERTIFY returns a class and radius only if $\underline{p_A} > \frac{1}{2}$ (otherwise it abstains). If $\underline{p_A} \leq \mathbb{P}[f(x+\varepsilon) = \widehat{c}_A]$ and $\frac{1}{2} < \underline{p_A}$, then we can invoke Theorem 2.3.1 with $\overline{p_B} = 1 - \underline{p_A}$ to obtain the desired guarantee. □

## A.4 Estimating the Certified Test-Set Accuracy

In this appendix, we show how to convert the "approximate certified test accuracy" considered in the main paper into a lower bound on the true certified test accuracy that holds with high probability over the randomness in CERTIFY.

Consider a classifier $g$, a test set $S = \{(x_1, c_1) \dots (x_m, c_m)\}$, and a radius $r$. For each example $i \in [m]$, let $z_i$ indicate whether $g$'s prediction at $x_i$ is both correct and robust at radius $r$, i.e.

$$z_i = \mathbf{1}[g(x_i + \delta) = c_i \;\; \forall \|\delta\|_2 < r] \tag{A.113}$$

The certified test set accuracy of $g$ at radius $r$ is defined as $\frac{1}{m}\sum_{i=1}^m z_i$. If $g$ is a randomized smoothing classifier, we cannot compute this quantity exactly, but we can estimate a lower bound that holds with arbitrarily high probability over the randomness in CERTIFY. In particular, suppose that we run CERTIFY with failure rate $\alpha$ on each example $x_i$ in the test set. Let the Bernoulli random variable $Y_i$ denote the event that on example $i$, CERTIFY returns the correct label $c_A = c_i$ and a certified radius $R$ which is greater than $r$. Let $Y = \sum_{i=1}^m Y_i$. In the main paper, we referred to $Y/m$ as the "approximate certified accuracy." It is "approximate" because $Y_i = 1$ does not mean that $z_i = 1$. Rather, from Proposition Proposition 2.3.4, we know the following: if $z_i = 0$, then $\mathbb{P}(Y_i = 1) \le \alpha$. We now show how to exploit this fact to construct a one-sided confidence interval for the unobserved quantity $\frac{1}{m}\sum_{i=1}^m z_i$ using the observed quantities $Y$ and $m$.

**Theorem A.4.1.** *For any $\rho > 0$, with probability at least $1 - \rho$ over the randomness in* CERTIFY,

$$\frac{1}{m}\sum_{i=1}^m z_i \ge \frac{1}{1-\alpha}\left(\frac{Y}{m} - \alpha - \sqrt{\frac{2\alpha(1-\alpha)\log(1/\rho)}{m}} - \frac{\log(1/\rho)}{3m}\right) \tag{A.114}$$

*Proof.* Let $m_{\text{good}} = \sum_{i=1}^m z_i$ and $m_{\text{bad}} = \sum_{i=1}^m (1 - z_i)$ be the number of test examples on which $z_i = 1$ or $z_i = 0$, respectively. We model $Y_i \sim \text{Bernoulli}(p_i)$, where $p_i$ is in general unknown. Let $Y_{\text{good}} = \sum_{i:z_i=1} Y_i$ and $Y_{\text{bad}} = \sum_{i:z_i=0} Y_i$. The quantity of interest, the certified accuracy $\frac{1}{m}\sum_{i=1}^m z_i$, is equal to $m_{\text{good}}/m$. However, we only observe $Y = Y_{\text{good}} + Y_{\text{bad}}$.

Note that if $z_i = 0$, then $p_i \le \alpha$, so we have $\mathbb{E}[Y_i] = p_i \le \alpha$ and assuming $\alpha \le \frac{1}{2}$, we have $\text{Var}[Y_i] = p_i(1-p_i) \le \alpha(1-\alpha)$.

Since $Y_{\text{bad}}$ is a sum of $m_{\text{bad}}$ independent random variables each bounded between zero and one, with $\mathbb{E}[Y_{\text{bad}}] \le \alpha m_{\text{bad}}$ and $\text{Var}(Y_{\text{bad}}) \le m_{\text{bad}}\alpha(1-\alpha)$, Bernstein's inequality guarantees that with probability at least $1 - \rho$ over the randomness in CERTIFY,

$$Y_{\text{bad}} \leq \alpha m_{\text{bad}} + \sqrt{2 m_{\text{bad}} \alpha (1-\alpha) \log(1/\rho)} + \frac{\log(1/\rho)}{3} \tag{A.115}$$

From now on, we manipulate this inequality — remember that it holds with probability at least $1-\rho$.

Since $Y = Y_{\text{good}} + Y_{\text{bad}}$, may write

$$Y_{\text{good}} \geq Y - \alpha m_{\text{bad}} - \sqrt{2 m_{\text{bad}} \alpha (1-\alpha) \log(1/\rho)} - \frac{\log(1/\rho)}{3} \tag{A.116}$$

Since $m_{\text{good}} \geq Y_{\text{good}}$, we may write

$$m_{\text{good}} \geq Y - \alpha m_{\text{bad}} - \sqrt{2 m_{\text{bad}} \alpha (1-\alpha) \log(1/\rho)} - \frac{\log(1/\rho)}{3} \tag{A.117}$$

Since $m_{\text{good}} + m_{\text{bad}} = m$, we may write

$$m_{\text{good}} \geq \frac{1}{1-\alpha} \left( Y - \alpha\, m - \sqrt{2 m_{\text{bad}} \alpha (1-\alpha) \log(1/\rho)} - \frac{\log(1/\rho)}{3} \right) \tag{A.118}$$

Finally, in order to make this confidence interval depend only on observables, we use $m_{\text{bad}} \leq m$ to write

$$m_{\text{good}} \geq \frac{1}{1-\alpha} \left( Y - \alpha\, m - \sqrt{2 m \alpha (1-\alpha) \log(1/\rho)} - \frac{\log(1/\rho)}{3} \right) \tag{A.119}$$

Dividing both sides of the inequality by $m$ recovers the theorem statement.

□

## A.5 ImageNet and CIFAR-10 Results

### A.5.1 Certification

Tables Table A.1 and Table A.2 show the approximate certified top-1 test set accuracy of randomized smoothing on ImageNet and CIFAR-10 with various noise levels $\sigma$. By "approximate certified accuracy," we mean that we ran CERTIFY on a subsample of the test set, and for each $r$ we report the fraction of examples on which CERTIFY (a) did not abstain, (b) returned the correct class, and (c) returned a radius $R$ greater than $r$. There is some probability (at most $\alpha$) that any example's certification is inaccurate. We used $\alpha = 0.001$ and $n = 100000$. On CIFAR-10 our base classifier was a 110-layer residual network and we certified the full test set; on ImageNet our base classifier was a ResNet-50 and we certified a subsample of 500 points. Note that the certified accuracy at $r = 0$ is just the standard accuracy of the smoothed classifier. See Appendix Appendix A.10 for more experimental details.

| | $r = 0.0$ | $r = 0.5$ | $r = 1.0$ | $r = 1.5$ | $r = 2.0$ | $r = 2.5$ | $r = 3.0$ |
|---|---|---|---|---|---|---|---|
| $\sigma = 0.25$ | **0.67** | **0.49** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $\sigma = 0.50$ | 0.57 | 0.46 | **0.37** | **0.29** | 0.00 | 0.00 | 0.00 |
| $\sigma = 1.00$ | 0.44 | 0.38 | 0.33 | 0.26 | **0.19** | **0.15** | **0.12** |

Table A.1: Approximate certified test accuracy on ImageNet. Each row is a setting of the hyperparameter $\sigma$, each column is an $\ell_2$ radius. The entry of the best $\sigma$ for each radius is bolded. For comparison, random guessing would attain 0.001 accuracy.

| | $r = 0.0$ | $r = 0.25$ | $r = 0.5$ | $r = 0.75$ | $r = 1.0$ | $r = 1.25$ | $r = 1.5$ |
|---|---|---|---|---|---|---|---|
| $\sigma = 0.12$ | **0.83** | 0.60 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| $\sigma = 0.25$ | 0.77 | **0.61** | 0.42 | 0.25 | 0.00 | 0.00 | 0.00 |
| $\sigma = 0.50$ | 0.66 | 0.55 | **0.43** | **0.32** | 0.22 | 0.14 | 0.08 |
| $\sigma = 1.00$ | 0.47 | 0.41 | 0.34 | 0.28 | **0.22** | **0.17** | **0.14** |

Table A.2: Approximate certified test accuracy on CIFAR-10. Each row is a setting of the hyperparameter $\sigma$, each column is an $\ell_2$ radius. The entry of the best $\sigma$ for each radius is bolded. For comparison, random guessing would attain 0.1 accuracy.

### A.5.2 Prediction

Table Table A.3 shows the performance of PREDICT as the number of Monte Carlo samples $n$ is varied between 100 and 10,000. Suppose that for some test example $(x, c)$, PREDICT returns the label $\widehat{c}_A$. We say that this prediction was *correct* if $\widehat{c}_A = c$ and we say that this prediction was *accurate* if $\widehat{c}_A = g(x)$. For example, a prediction could be correct but inaccurate if $g$ is wrong at $x$, yet PREDICT accidentally returns the correct class. Ideally, we'd like PREDICT to be both correct and accurate.

With $n = 100$ Monte Carlo samples and a failure rate of $\alpha = 0.001$, PREDICT is cheap to evaluate (0.15 seconds on our hardware) yet it attains relatively high top-1 accuracy of 65% on the ImageNet test set, and only abstains 12% of the time. When we use $n =$ 10,000 Monte Carlo samples, PREDICT takes longer to evaluate (15 seconds), yet only abstains 4% of the time. Interestingly, we observe from Table Table A.3 that most of the abstentions when $n = 100$ were for examples on which $g$ was wrong, so in practice we would lose little accuracy by taking $n$ to be as small as 100.

| N | CORR, ACC | CORR, INACC | INCORR, ACC | INCORR, INACC | ABSTAIN |
|---|---|---|---|---|---|
| 100 | 0.65 | 0.00 | 0.23 | 0.00 | 0.12 |
| 1000 | 0.68 | 0.00 | 0.28 | 0.00 | 0.04 |
| 10000 | 0.69 | 0.00 | 0.30 | 0.00 | 0.01 |

Table A.3: Performance of PRECICT as $n$ is varied. The dataset was ImageNet and $\sigma = 0.25$, $\alpha = 0.001$. Each column shows the fraction of test examples which ended up in one of five categories; the prediction at $x$ is "correct" (CORR) if PREDICT returned the true label (otherwise it is INCORR), while the prediction is "accurate" (ACC) if PREDICT returned $g(x)$ (otherwise it is INACC). Computing $g(x)$ exactly is not possible, so in order to determine whether PREDICT was accurate, we took the gold standard to be the top class over $n =$100,000 Monte Carlo samples.

## A.6 Training with Noise

As mentioned in section Section 2.3.3, in the experiments for this paper, we followed Lecuyer et al. [2019] and trained the base classifier by minimizing the cross-entropy loss with Gaussian data augmentation. We now provide some justification for this idea.

Let $\{(x_1, c_1), \ldots, (x_n, c_n)\}$ be a training dataset. We assume that the base classifier takes the form $f(x) = \arg\max_{c \in \mathcal{Y}} f_c(x)$, where each $f_c$ is the scoring function for class $c$.

Suppose that our goal is to maximize the sum of of the log-probabilities that $f$ will classify each $x_i + \varepsilon$ as $c_i$:

$$\sum_{i=1}^{n} \log \mathbb{P}_\varepsilon(f(x_i + \varepsilon) = c_i) = \sum_{i=1}^{n} \log \mathbb{E}_\varepsilon \, \mathbf{1}\left[\arg\max_c f_c(x_i + \varepsilon) = c_i\right] \tag{A.120}$$

Recall that the softmax function can be interpreted as a continuous, differentiable approximation to $\arg\max$:

$$\mathbf{1}\left[\arg\max_c f_c(x_i + \varepsilon) = c_i\right] \approx \frac{\exp(f_{c_i}(x_i + \varepsilon))}{\sum_{c \in \mathcal{Y}} \exp(f_c(x_i + \varepsilon))} \tag{A.121}$$

Therefore, our objective is approximately equal to:

$$\sum_{i=1}^{n} \log \mathbb{E}_\varepsilon \left[\frac{\exp(f_{c_i}(x_i + \varepsilon))}{\sum_{c \in \mathcal{Y}} \exp(f_c(x_i + \varepsilon))}\right] \tag{A.122}$$

By Jensen's inequality and the concavity of $\log$, this quantity is lower-bounded by:

$$\sum_{i=1}^{n} \mathbb{E}_\varepsilon \left[\log \frac{\exp(f_{c_i}(x_i + \varepsilon))}{\sum_{c \in \mathcal{Y}} \exp(f_c(x_i + \varepsilon))}\right] \tag{A.123}$$

which is the negative of the cross-entropy loss under Gaussian data augmentation.

Therefore, minimizing the cross-entropy loss under Gaussian data augmentation will maximize (Equation (A.122)), which will approximately maximize (Equation (A.120)).

## A.7 Noise Level can Scale with Input Resolution

Since our robustness guarantee equation 2.3 in Theorem 2.3.1 does not explicitly depend on the data dimension $d$, one might worry that randomized smoothing is less effective for images in high resolution — certifying a fixed $\ell_2$ radius is "less impressive" for, say, $224 \times 224$ image than for a $56 \times 56$ image. However, it turns out that in high resolution, images can be corrupted with larger levels of isotropic Gaussian noise while still preserving their content. This fact is made clear by Figure Figure A.4, which shows an image at high and low resolution corrupted by Gaussian noise with the same variance.full The class ("hummingbird") is easy to discern from the high-resolution noisy image, but not from the low-resolution noisy image. As a consequence, in high resolution one can take $\sigma$ to be larger while still being able to obtain a base classifier that classifies noisy images accurately. Since our Theorem 2.3.1 robustness guarantee scales linearly with $\sigma$, this means that in high resolution one can certify larger radii.

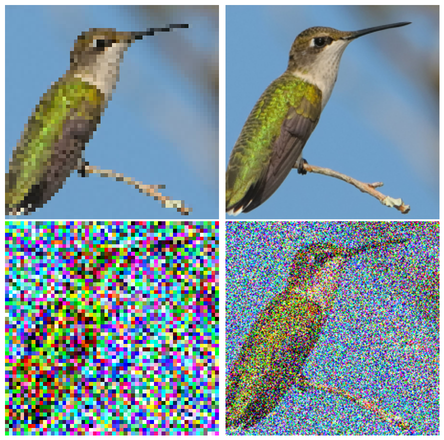


Figure A.4: **Top**: An ImageNet image from class "hummingbird" in resolutions 56x56 (left) and 224x224 (right). **Bottom**: the same images corrupted by isotropic Gaussian noise at $\sigma = 0.5$. On noiseless images the class is easy to distinguish no matter the resolution, but on noisy data the class is much easier to distinguish when the resolution is high.

The argument above can be made rigorous, though we first need to decide what it means for two images to be high- and low-resolution versions of each other. Here we present one solution:

Let $\mathcal{X}$ denote the space of "high-resolution" images in dimension $2k \times 2k \times 3$, and let $\mathcal{X}'$ denote the space of "low-resolution" images in dimension $k \times k \times 3$. Let

$\textsc{AvgPool} : \mathcal{X} \to \mathcal{X}'$ be the function which takes as input an image $x$ in dimension $2k \times 2k \times 3$, averages together every 2x2 square of pixels, and outputs an image in dimension $k \times k \times 3$.

Equipped with these definitions, we can say that $(x, x') \in \mathcal{X} \times \mathcal{X}'$ are a high/low resolution image pair if $x' = \textsc{AvgPool}(x)$.

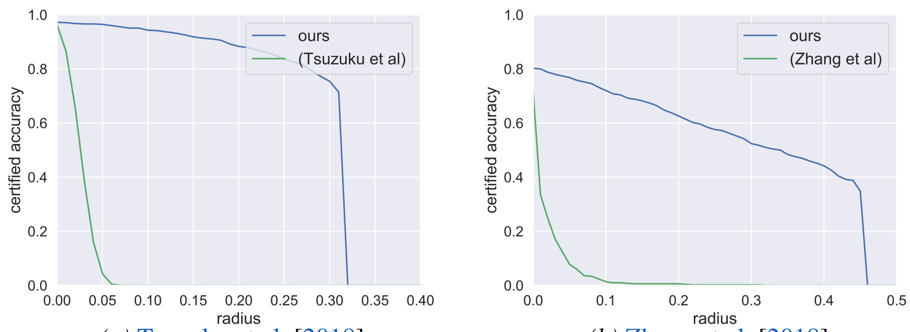
**Proposition A.7.1.** *Given any smoothing classifier $g' : \mathcal{X}' \to \mathcal{Y}$, one can construct a smoothing classifier $g : \mathcal{X} \to \mathcal{Y}$ with the following property: for any $x \in \mathcal{X}$ and $x' = \textsc{AvgPool}(x)$, $g$ predicts the same class at $x$ that $g'$ predicts at $x'$, but is certifiably robust at twice the radius.*

*Proof.* Given some smoothing classifier $g' = (f', \sigma')$ from $\mathcal{X}'$ to $\mathcal{Y}$, define $g$ to be the smoothing classifier $(f, \sigma)$ from $\mathcal{X}$ to $\mathcal{Y}$ with noise level $\sigma = 2\sigma'$ and base classifier $f(x) = f'(\textsc{AvgPool}(x))$. Note that the average of four independent copies of $\mathcal{N}(0, (2\sigma)^2)$ is distributed as $\mathcal{N}(0, \sigma^2)$. Therefore, for any high/low-resolution image pair $x' = \textsc{AvgPool}(x)$, the random variable $\textsc{AvgPool}(x + \varepsilon)$, where $\varepsilon \sim \mathcal{N}(0, (2\sigma)^2 I_{2k \times 2k \times 3})$, is equal in distribution to the random variable $x' + \varepsilon'$, where $\varepsilon' \sim \mathcal{N}(0, \sigma^2 I_{k \times k \times 3})$. Hence, $f(x + \varepsilon) = f'(\textsc{AvgPool}(x + \varepsilon))$ has the same distribution as $f'(x' + \varepsilon')$. By the definition of $g$, this means that $g(x) = g'(x')$, Additionally, by Theorem 2.3.1, since $\sigma = 2\sigma'$, this means that $g$'s prediction at $x$ is certifiably robust at twice the radius as $g'$'s prediction at $x'$. $\square$

## A.8 Additional Experiments

### A.8.1 Comparisons to Baselines

Figure Figure A.5 compares the certified accuracy of a smoothed 20-layer resnet to that of the released models from two recent works on certified $\ell_2$ robustness: the Lipschitz approach from Tsuzuku et al. [2018] and the approach from Zhang et al. [2018]. Note that in these experiments, the base classifier for smoothing was larger than the networks of competing approaches. The comparison to Zhang et al. [2018] is on CIFAR-10, while the comparison to Tsuzuku et al. [2018] is on SVHN. Note that for each comparison, we preprocessed the dataset to follow the preprocessing used when the baseline was trained; therefore, the radii reported for CIFAR-10 here are not comparable to the radii reported elsewhere in this paper. Full experimental details are in Appendix Appendix A.10.



*(a)* Tsuzuku et al. [2018] *(b)* Zhang et al. [2018]

Figure A.5: Randomized smoothing with a 20-layer resnet base classifier attains higher certified accuracy than the released models from two recent works on certified $\ell_2$ robustness.

### A.8.2 High-Probability Guarantees

Appendix Appendix A.4 details how to use CERTIFY to obtain a lower bound on the certified test accuracy at radius $r$ of a randomized smoothing classifier that holds with high probability over the randomness in CERTIFY. In the main paper, we declined to do this and simply reported the approximate certified test accuracy, defined as the fraction of test examples for which CERTIFY gives the correct prediction and certifies it at radius $r$. Of course, with some probability (guaranteed to be less than $\alpha$), each of these certifications is wrong.

However, we now demonstrate empirically that there is a negligible difference between a proper high-probability lower bound on the certified accuracy and the approximate version that we reported in the paper. We created a randomized smoothing classifier $g$ on ImageNet with a ResNet-50 base classifier and noise level $\sigma = 0.25$. We used CERTIFY with $\alpha = 0.001$ to certify a subsample of 500 examples from the ImageNet test set. From this we computed the approximate certified test accuracy at each radius $r$. Then we used the correction from Appendix Appendix A.4 with $\rho = 0.001$ to obtain a lower bound on the certified test accuracy at $r$ that holds pointwise with probability at least $1 - \rho$ over the randomness in CERTIFY. Figure Figure A.6 plots both quantities as a function of $r$. Observe that the difference is so negligible that the lines almost overlap.

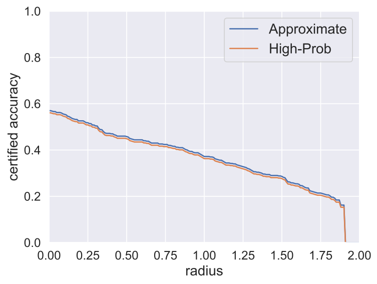
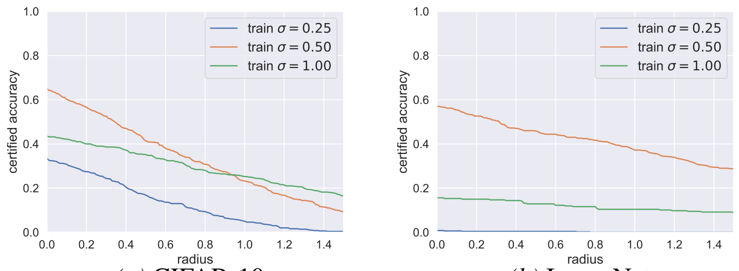


Figure A.6: The difference between the approximate certified accuracy, and a high-probability lower bound on the certified accuracy, is negligible.

### A.8.3 How Much Noise to Use When Training the Base Classifier?

In the main paper, whenever we created a randomized smoothing classifier $g$ at noise level $\sigma$, we always trained the corresponding base classifier $f$ with Gaussian data augmentation at noise level $\sigma$. In Figure Figure A.7, we show the effects of training the base classifier with a different level of Gaussian noise. Observe that $g$ has a lower certified accuracy if $f$ was trained using a different noise level. It seems to be worse to train with noise $< \sigma$ than to train with noise $> \sigma$.



*(a)* CIFAR-10 *(b)* ImageNet

Figure A.7: Vary training noise while holding prediction noise fixed at $\sigma = 0.50$.

## A.9 Derivation of Prior Randomized Smoothing Guarantees

In this appendix, we derive the randomized smoothing guarantees of Lecuyer et al. [2019] and Li et al. [2018a] using the notation of our paper. Both guarantees take same general form as ours, except with a different expression for $R$:

**Theorem (generic guarantee):** *Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$. Let $g$ be defined as in (Equation (2.1)). Suppose $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0,1]$ satisfy:*

$$\mathbb{P}(f(x+\varepsilon) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} \mathbb{P}(f(x+\varepsilon) = c) \tag{A.124}$$

*Then $g(x+\delta) = c_A$ for all $\|\delta\|_2 < R$.*

For convenience, define the notation $X \sim \mathcal{N}(x, \sigma^2 I)$ and $Y \sim \mathcal{N}(x+\delta, \sigma^2 I)$.

### A.9.1 Lecuyer et al. [2019]

Lecuyer et al. [2019] proved a version of the generic robustness guarantee in which

$$R = \sup_{0 < \beta \leq \min\left(1, \frac{1}{2}\log\frac{\underline{p_A}}{\overline{p_B}}\right)} \frac{\sigma\beta}{\sqrt{2\log\left(\frac{1.25(1+\exp(\beta))}{\underline{p_A} - \exp(2\beta)\overline{p_B}}\right)}} \tag{A.125}$$

*Proof.* In order to avoid notation that conflicts with the rest of this paper, we use $\beta$ and $\gamma$ where Lecuyer et al. [2019] used $\epsilon$ and $\delta$.

Suppose that we have some $0 < \beta \leq 1$ and $\gamma > 0$ such that

$$\sigma^2 = \frac{\|\delta\|^2}{\beta^2} 2\log\frac{1.25}{\gamma} \tag{A.126}$$

The “Gaussian mechanism” from differential privacy guarantees that:

$$\mathbb{P}(f(X) = c_A) \leq \exp(\beta)\mathbb{P}(f(Y) = c_A) + \gamma \tag{A.127}$$

and, symmetrically,

$$\mathbb{P}(f(Y) = c_B) \leq \exp(\beta)\mathbb{P}(f(X) = c_B) + \gamma \tag{A.128}$$

See Lecuyer et al. [2019], Lemma 2 for how to obtain this form from the standard form of the $(\beta, \gamma)$ DP definition.

Fix a perturbation $\delta$. To guarantee that $g(x + \delta) = c_A$, we need to show that $\mathbb{P}(f(Y) = c_A) > \mathbb{P}(f(Y) = c_B)$ for each $c_B \neq c_A$.

Together, (Equation (A.127)) and (Equation (A.128)) imply that to guarantee $\mathbb{P}(f(Y) = c_A) > \mathbb{P}(f(Y) = c_B)$ for any $c_B$, it suffices to show that:

$$\mathbb{P}(f(X) = c_A) > \exp(2\beta)\mathbb{P}(f(X) = c_B) + \gamma(1 + \exp(\beta)) \tag{A.129}$$

Therefore, in order to guarantee that$\mathbb{P}(f(Y) = c_A) > \mathbb{P}(f(Y) = c_B)$ for each $c_B \neq c_A$, by (Equation (A.124)) it suffices to show:

$$\underline{p_A} > \exp(2\beta)\overline{p_B} + \gamma(1 + \exp(\beta)) \tag{A.130}$$

Now, inverting (Equation (A.126)), we obtain:

$$\gamma = 1.25 \exp\left(-\frac{\sigma^2\beta^2}{2\|\delta\|^2}\right) \tag{A.131}$$

Plugging (Equation (A.131)) into (Equation (A.130)), we see that to guarantee $\mathbb{P}(f(Y) = c_A) \geq \mathbb{P}(f(Y) = c_B)$ it suffices to show that:

$$\underline{p_A} > \exp(2\beta)\overline{p_B} + 1.25 \exp\left(-\frac{\sigma^2\beta^2}{2\|\delta\|^2}\right)(1 + \exp(\beta)) \tag{A.132}$$

which rearranges to:

$$\frac{\underline{p_A} - \exp(2\beta)\overline{p_B}}{1.25(1 + \exp(\beta))} > \exp\left(-\frac{\sigma^2\beta^2}{2\|\delta\|^2}\right) \tag{A.133}$$

Since the RHS is always positive, and the denominator on the LHS is always positive, this condition can only possibly hold if the numerator on the LHS is positive. Therefore, we need to restrict $\beta$ to

$$0 < \beta \leq \min\left(1, \frac{1}{2}\log\frac{\underline{p_A}}{\overline{p_B}}\right) \tag{A.134}$$

The condition (Equation (A.133)) is equivalent to:

$$\|\delta\|^2 \log \frac{1.25(1+\exp(\beta))}{\underline{p_A} - \exp(2\beta)\overline{p_B}} < \frac{\sigma^2\beta^2}{2} \tag{A.135}$$

Since $\underline{p_A} \leq 1$ and $\overline{p_B} \geq 0$, the denominator in the LHS is $\leq 1$ which is in turn $\leq$ the numerator on the LHS. Therefore, the term inside the log in the LHS is greater than 1, so the log term on the LHS is greater than zero. Therefore, we may divide both sides of the inequality by the log term on the LHS to obtain:

$$\|\delta\|^2 < \frac{\sigma^2\beta^2}{2\log\left(\frac{1.25(1+\exp(\beta))}{\underline{p_A}-\exp(2\beta)\overline{p_B}}\right)} \tag{A.136}$$

Finally, we take the square root and maximize the bound over all valid $\beta$ (Equation (A.134)) to yield:

$$\|\delta\| < \sup_{0<\beta\leq\min\left(1,\frac{1}{2}\log\frac{\underline{p_A}}{\overline{p_B}}\right)} \frac{\sigma\beta}{\sqrt{2\log\left(\frac{1.25(1+\exp(\beta))}{\underline{p_A}-\exp(2\beta)\overline{p_B}}\right)}} \tag{A.137}$$

□

Figure Figure A.8a plots this bound at varying settings of the tuning parameter $\beta$, while Figure Figure A.8c plots how the bound varies with $\beta$ for a fixed $\underline{p_A}$ and $\overline{p_B}$.

### A.9.2 Li et al. [2018a]

Li et al. [2018a] proved a version of the generic robustness guarantee in which

$$R = \sup_{\alpha>0} \sigma\sqrt{-\frac{2}{\alpha}\log\left(1-\underline{p_A}-\overline{p_B}+2\left(\frac{1}{2}(\underline{p_A}^{1-\alpha}+\overline{p_B}^{1-\alpha})^{1-\alpha}\right)\right)} \tag{A.138}$$

*Proof.* A generalization of KL divergence, the $\alpha$-Renyi divergence is an information theoretic measure of distance between two distributions. It is parameterized by some $\alpha > 0$. The $\alpha$-Renyi divergence between two discrete distributions $P$ and $Q$ is defined as:

$$D_\alpha(P||Q) := \frac{1}{\alpha-1}\log\left(\sum_{i=1}^{k}\frac{p_i^\alpha}{q_i^{\alpha-1}}\right) \tag{A.139}$$

In the continuous case, this sum is replaced with an integral. The divergence is undefined when $\alpha = 1$ since a division by zero occurs, but the limit of $D_\alpha(P||Q)$ as $\alpha \to 1$ is the KL divergence between $P$ and $Q$.

Li et al. [2018a] prove that if $P$ is a discrete distribution for which the highest probability class has probability $\geq \underline{p_A}$ and all other classes have probability $\leq \overline{p_B}$, then for any other discrete distribution $Q$ for which

$$D_\alpha(P||Q) < -\log\left(1 - \underline{p_A} - \overline{p_B} + 2\left(\frac{1}{2}(\underline{p_A}^{1-\alpha} + \overline{p_B}^{1-\alpha})^{1-\alpha}\right)\right) \tag{A.140}$$

the highest-probability class in $Q$ is guaranteed to be the same as the highest-probability class in $P$.

We now apply this result to the discrete distributions $P = f(X)$ and $Q = f(Y)$. If $D_\alpha(f(X)||f(Y))$ satisfies (Equation (A.140)), then it is guaranteed that $g(x) = g(x + \delta)$.

The data processing inequality states that applying a function to two random variables can only decrease the $\alpha$-Renyi divergence between them. In particular,

$$D_\alpha(f(X)||f(Y)) \leq D_\alpha(X||Y) \tag{A.141}$$

There is a closed-form expression for the $\alpha$-Renyi divergence between two Gaussians:

$$D_\alpha(X||Y) = \frac{\alpha\|\delta\|^2}{2\sigma^2} \tag{A.142}$$

Therefore, we can guarantee that $g(x + \delta) = c_A$ so long as

$$\frac{\alpha\|\delta\|^2}{2\sigma^2} < -\log\left(1 - \underline{p_A} - \overline{p_B} + 2\left(\frac{1}{2}(\underline{p_A}^{1-\alpha} + \overline{p_B}^{1-\alpha})^{1-\alpha}\right)\right) \tag{A.143}$$

which simplifies to

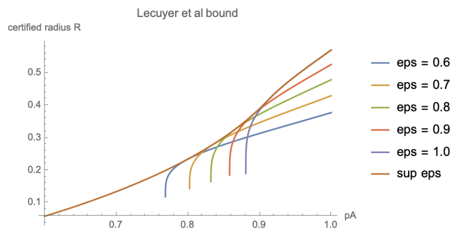
$$\|\delta\| < \sigma\sqrt{-\frac{2}{\alpha}\log\left(1 - \underline{p_A} - \overline{p_B} + 2\left(\frac{1}{2}(\underline{p_A}^{1-\alpha} + \overline{p_B}^{1-\alpha})^{1-\alpha}\right)\right)} \tag{A.144}$$

Finally, since this result holds for any $\alpha > 0$, we may maximize over $\alpha$ to obtain the largest possible certified radius:

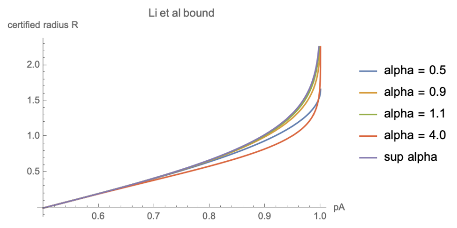
$$\|\delta\| < \sup_{\alpha>0} \sigma\sqrt{-\frac{2}{\alpha}\log\left(1 - \underline{p_A} - \overline{p_B} + 2\left(\frac{1}{2}(\underline{p_A}^{1-\alpha} + \overline{p_B}^{1-\alpha})^{1-\alpha}\right)\right)} \tag{A.145}$$

□

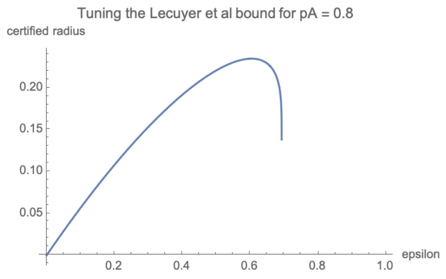
Figure Figure A.8b plots this bound at varying settings of the tuning parameter $\alpha$, while figure Figure A.8d plots how the bound varies with $\alpha$ for a fixed $\underline{p_A}$ and $\overline{p_B}$.



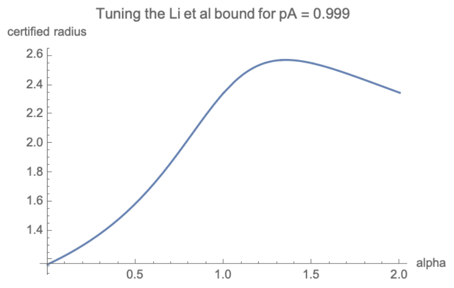
*(a)* The Lecuyer et al. [2019] bound over several settings of $\beta$. The brown line is the pointwise supremum over all eligible $\beta$, computed numerically.



*(b)* The Li et al. [2018a] bound over several settings of $\alpha$. The purple line is the pointwise supremum over all eligible $\alpha$, computed numerically.



*(c)* Tuning the Lecuyer et al. [2019] bound wrt $\beta$ when $\underline{p_A} = 0.8, \overline{p_B} = 0.2$



*(d)* Tuning the Li et al. [2018a] bound wrt $\alpha$ when $\underline{p_A} = 0.999, \overline{p_B} = 0.0001$

## A.10 Experiment Details

### A.10.1 Comparison to Baselines

We compared randomized smoothing against three recent approaches for $\ell_2$-robust classification [Tsuzuku et al., 2018, Wong et al., 2018, Zhang et al., 2018]. Tsuzuku et al. [2018] and Wong et al. [2018] propose both a robust training method and a complementary certification mechanism, while Zhang et al. [2018] propose a method to certify generically trained networks. In all cases we compared against networks provided by the authors. We compared against Wong et al. [2018] and Zhang et al. [2018] on CIFAR-10, and we compared against Tsuzuku et al. [2018] on SVHN.

In image classification it is common practice to preprocess a dataset by subtracting from each channel the mean over the dataset, and dividing each channel by the standard deviation over the dataset. However, we wanted to report certified radii in the original image coordinates rather than in the standardized coordinates. Therefore, throughout most of this work we *first* added the Gaussian noise, and *then* standardized the channels, before feeding the image to the base classifier. (In the practical PyTorch implementation, the first layer of the base classifier was a layer that standardized the input.) However, all of the baselines we compared against provided pre-trained networks which assumed that the dataset was first preprocessed in a specific way. Therefore, when comparing against the baselines we also preprocessed the datasets first, so that we could report certified radii that were directly comparable to the radii reported by the baseline methods.

**Comparison to Wong et al. [2018]** Following Wong et al. [2018], the CIFAR-10 dataset was preprocessed by subtracting $(0.485, 0.456, 0.406)$ and dividing by $(0.225, 0.225, 0.225)$.

While the body of the Wong et al. [2018] paper focuses on $\ell_\infty$ certified robustness, their algorithm naturally extends to $\ell_2$ certified robustness, as developed in the appendix of the paper. We used three $\ell_2$-trained residual networks publicly released by the authors, each trained with a different setting of their hyperparameter $\epsilon \in \{0.157, 0.628, 2.51\}$. We used code publicly released by the authors at `https://github.com/locuslab/convex_adversarial/blob/master/examples/cifar_evaluate.py` to compute the robustness radius of test images. The code accepts a radius and returns TRUE (robust) or FALSE (not robust); we incorporated this subroutine into a binary search procedure to find the largest radius for which the code returned TRUE.

For randomized smoothing we used $\sigma = 0.6$ and a 20-layer residual network base classifier. We ran CERTIFY with $n_0 = 100$, $n = 100{,}000$ and $\alpha = 0.001$.

For both methods, we certified the full CIFAR-10 test set.

**Comparison to Tsuzuku et al. [2018]** Following Tsuzuku et al. [2018], the SVHN dataset was not preprocessed except that pixels were divided by 255 so as to lie within [0, 1].

We compared against a pretrained network provided to us by the authors in which the hyperparameter of their method was set to $c = 0.1$. The network was a wide residual network with 16 layers and a width factor of 4. We used the authors' code at `https://github.com/ytsmiling/lmt` to compute the robustness radius of test images.

For randomized smoothing we used $\sigma = 0.1$ and a 20-layer residual network base classifier. We ran CERTIFY with $n_0 = 100$, $n = 100{,}000$ and $\alpha = 0.001$.

For both methods, we certified the whole SVHN test set.

**Comparison to Zhang et al. [2018]** Following Zhang et al. [2018], the CIFAR-10 dataset was preprocessed by subtracting 0.5 from each pixel.

We compared against the `cifar_7_1024_vanilla` network released by the authors, which is a 7-layer MLP. We used the authors' code at `https://github.com/IBM/CROWN-Robustness-Certification` to compute the robustness radius of test images.

For randomized smoothing we used $\sigma = 1.2$ and a 20-layer residual network base classifier. We ran CERTIFY with $n_0 = 100$, $n = 100{,}000$ and $\alpha = 0.001$.

For randomized smoothing, we certified the whole CIFAR-10 test set. For Zhang et al. [2018], we certified every fourth image in the CIFAR-10 test set.

### A.10.2 ImageNet and CIFAR-10 Experiments

Our code is available at `http://github.com/locuslab/smoothing`.

In order to report certified radii in the original coordinates, we *first* added Gaussian noise, and *then* standardized the data. Specifically, in our PyTorch implementation, the first layer of the base classifier was a normalization layer that performed a channel-wise standardization of its input. For CIFAR-10 we subtracted the dataset mean $(0.4914, 0.4822, 0.4465)$ and divided by the dataset standard deviation $(0.2023, 0.1994, 0.2010)$. For ImageNet we subtracted the dataset mean $(0.485, 0.456, 0.406)$ and divided by the standard deviation $(0.229, 0.224, 0.225)$.

For both ImageNet and CIFAR-10, we trained the base classifier with random horizontal flips and random crops (in addition to the Gaussian data augmentation discussed explicitly in the paper). On ImageNet we trained with synchronous SGD on four NVIDIA RTX 2080 Ti GPUs; training took approximately three days.

On ImageNet our base classifier used the ResNet-50 architecture provided in `torchvision`. On CIFAR-10 we used a 110-layer residual network from `https://github.com/bearpaw/pytorch-classification`.

On ImageNet we certified every 100-th image in the validation set, for 500 images total. On CIFAR-10 we certified the whole test set.

In Figure Figure 2.8 (**middle**) we fixed $\sigma = 0.25$ and $\alpha = 0.001$ while varying the number of samples $n$. We did not actually vary the number of samples $n$ that we simulated: we kept this number fixed at 100,000 but varied the number that we fed the Clopper-Pearson confidence interval.

In Figure Figure 2.8 (**right**), we fixed $\sigma = 0.25$ and $n =$100,000 while varying $\alpha$.

### A.10.3 Adversarial Attacks

As discussed in Section Section 2.4, we subjected smoothed classifiers to a projected gradient descent-style adversarial attack. We now describe the details of this attack.

Let $f$ be the base classifier and let $\sigma$ be the noise level. Following Li et al. [2018a], given an example $(x, c) \in \mathbb{R}^d \times \mathcal{Y}$ and a radius $r$, we used a projected gradient descent style adversarial attack to optimize the objective:

$$\underset{\delta:\|\delta\|_2<r}{\arg\max}\ \mathbb{E}_{\varepsilon\sim\mathcal{N}(0,\sigma^2 I)}\left[\ell(f(x+\delta+\varepsilon), c)\right] \tag{A.146}$$

where $\ell$ is the softmax loss function. (Breaking notation with the rest of the paper in which $f$ returns a class, the function $f$ here refers to the function that maps an image in $\mathbb{R}^d$ to a vector of classwise scores.)

At each iteration of the attack, we drew $k$ samples of noise, $\varepsilon_1 \dots \varepsilon_k \sim \mathcal{N}(0, \sigma^2 I)$, and followed the stochastic gradient $g_t = \sum_{i=1}^{k} \nabla_{\delta_t} \ell(f(x+\delta_t+\varepsilon_k), c)$.

As is typical, we used a "steepest ascent" update rule, which, for the $\ell_2$ norm, means that we normalized the gradient before applying the update. The overall PGD update is: $\delta_{t+1} = \text{proj}_r\left(\delta_t + \eta\frac{g_t}{\|g_t\|}\right)$ where the function $\text{proj}_r$ that projects its input onto the ball $\{z : \|z\|_2 \leq r\}$ is given by $\text{proj}_r(z) = \frac{rz}{\max(r, \|z\|_2)}$. We used a constant step size $\eta$ and a fixed number $T$ of PGD iterations.

In practice, our step size was $\eta = 0.1$, we used $T = 20$ steps of PGD, and we computed the stochastic gradient using $k = 1000$ Monte Carlo samples.

Unfortunately, the objective we optimize (Equation (A.146)) is not actually the attack objective of interest. To force a misclassification, an attacker needs to find some perturbation $\delta$ with $\|\delta\|_2 < r$ and some class $c_B$ for which

$$\mathbb{P}_{\varepsilon\sim\mathcal{N}(0,\sigma^2 I)}(f(x+\delta+\varepsilon) = c_B) \geq \mathbb{P}_{\varepsilon\sim\mathcal{N}(0,\sigma^2 I)}(f(x+\delta+\varepsilon) = c) \tag{A.147}$$

Effective adversarial attacks against randomized smoothing are outside the scope of this paper.

## A.11 Examples of Noisy Images

We now show examples of CIFAR-10 and ImageNet images corrupted with varying levels of noise.

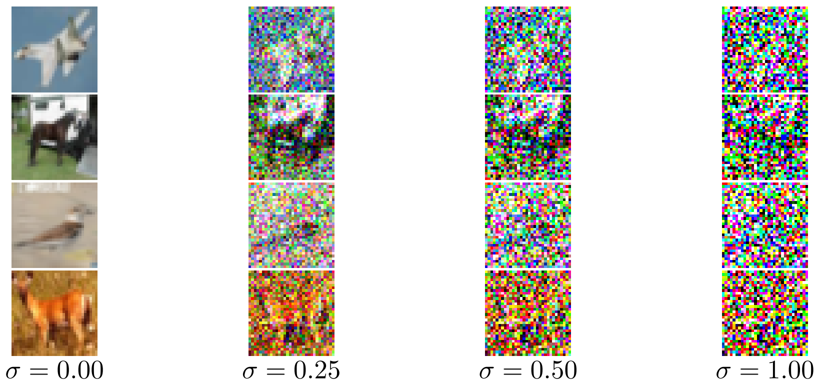


Figure A.9: CIFAR-10 images additively corrupted by varying levels of Gaussian noise $\mathcal{N}(0, \sigma^2 I)$. Pixel values greater than 1.0 (=255) or less than 0.0 (=0) were clipped to 1.0 or 0.0.

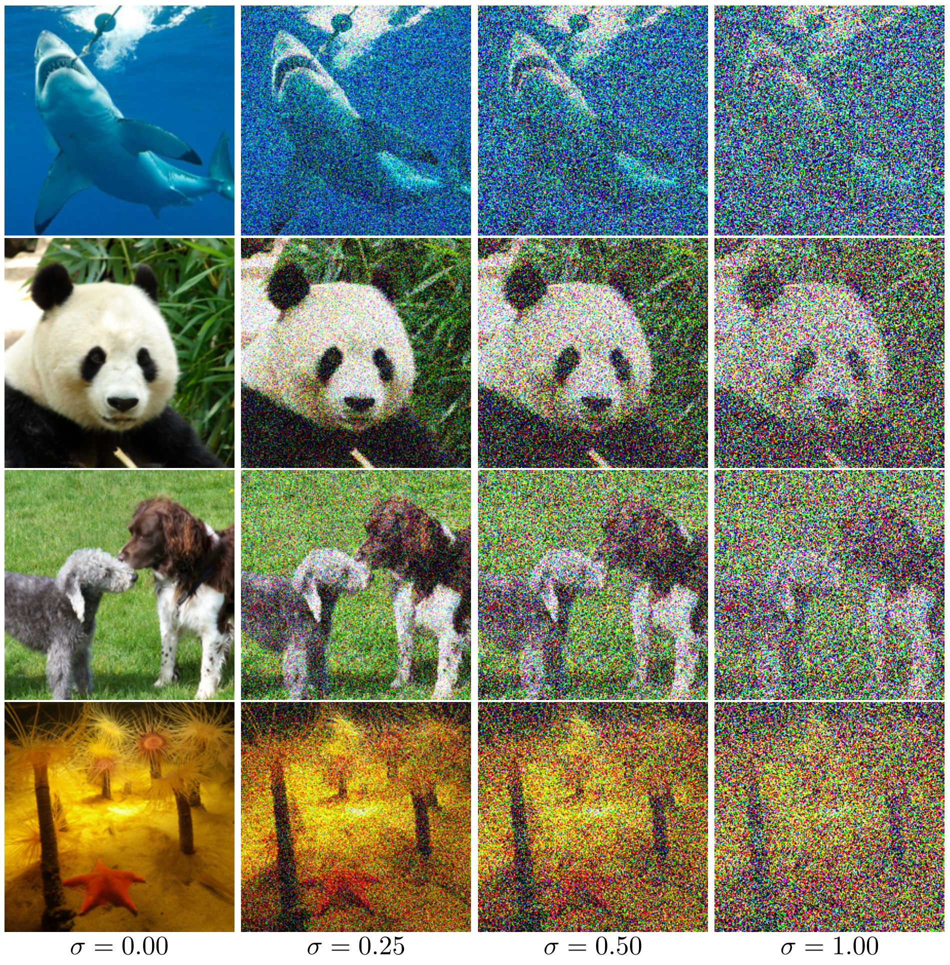


Figure A.10: ImageNet images additively corrupted by varying levels of Gaussian noise $\mathcal{N}(0, \sigma^2 I)$. Pixel values greater than 1.0 (=255) or less than 0.0 (=0) were clipped to 1.0 or 0.0.

# Appendix B

# Appendix for Chapter 3

## B.1 Generic Randomized Smoothing Algorithm

**Algorithm 8** Generic randomized smoothing procedure

```
**Input:** function $\phi : \mathcal{X} \to \{0, 1\}$, number of samples $N$, smoothing distribution $\mu$, test point to predict $x_0$, failure probability $\delta > 0$.
**for** $i = 1, \ldots, N$ **do**
    Sample $x_i \sim \mu(x_0)$ and compute $y_i = \phi(x_i)$.
**end for**
Compute approximate smoothed output
    $$\widehat{g}(\mu, \phi) = \mathbf{1}\left\{\left(\frac{1}{N}\sum_{i=1}^{N} y_i\right) \geq \frac{1}{2}\right\}.$$
Compute bound $\widehat{G}(\mu, \phi)$ such that with probability $\geq 1 - \delta$
    $$\widehat{G}(\mu, \phi) \begin{cases} \leq G(\mu, \phi) & \text{if } \widehat{g}(\mu, \phi) = 1 \\ \geq G(\mu, \phi) & \text{if } \widehat{g}(\mu, \phi) = 0. \end{cases}$$
**Output:** Prediction $\widehat{g}(\mu, \phi)$ and probability bound $\widehat{G}(\mu, \phi)$, or abstention if $\widehat{g}(\mu, \phi) \neq \text{sign}(\widehat{G}(\mu, \phi) - \frac{1}{2})$.
```

## B.2 The Multi-Class Setting

Although the notation and algorithms are slightly more complex, all the methods we have discussed in the main paper can be extended to the multi-class setting. In this case, we consider a class label $y \in \{1, \dots, K\}$, and we again seek some smoothed prediction such that the classifier's prediction on a new point will not change with some number $r$ flips of the labels in the training set.

### B.2.1 Randomized Smoothing in the Multi-Class Case

We here extend our notation to the case of more than two classes. Recall our original definition of $G$,

$$G(\mu, \phi) = \mathbf{E}_{x \sim \mu}[\phi(x)] = \int_{\mathcal{X}} \mu(x)\phi(x)dx,$$

where $\phi : \mathcal{X} \to \{0, 1\}$. More generally, consider a classifier $\phi : \mathcal{X} \to [K]$, outputting the index of one of $K$ classes. Under this formulation, for a given class $c \in [K]$, we have

$$G(\mu, \phi, c) = \mathbf{E}_{x \sim \mu}[\phi_c(x)] = \int_{\mathcal{X}} \mu(x)\phi_c(x)dx,$$

where $\phi_c(x) = \mathbf{1}\{\phi(x) = c\}$ is the indicator function for if $\phi(x)$ outputs the class $c$. In this case, the hard threshold $g$ is evaluated by returning the class with the highest probability. That is,

$$g(\mu, \phi) = \arg\max_c G(\mu, \phi, c).$$

### B.2.2 Linearization and Chernoff Bound Approach for the Multi-Class Case

Using the same linearization approach as in the binary case, we can formulate an analogous approach which forgoes the need to actually perform random sampling at all and instead directly bounds the randomized classifier using the Chernoff bound.

Adopting the same notation as in the main text, the equivalent least-squares classifier for the multi-class setting finds some set of weights

$$\widehat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

where $\boldsymbol{Y} \in \{0, 1\}^{n \times K}$ is a binary matrix with each row equal to a one-hot encoding of the class label (note that the resulting $\widehat{\boldsymbol{\beta}} \in \mathbb{R}^{k \times K}$ is now a matrix, and we let $\widehat{\boldsymbol{\beta}}_i$

refer to the $i$th column). At prediction time, the predicted class of some new point $x_{n+1}$ is simply given by the prediction with the highest value, i.e.,

$$\widehat{y}_{n+1} = \arg\max_i \widehat{\boldsymbol{\beta}}_i^T h(x_{n+1}).$$

Alternatively, following the same logic as in the binary case, this same prediction can be written in terms of the $\boldsymbol{\alpha}$ variable as

$$\widehat{y}_{n+1} = \arg\max_i \boldsymbol{\alpha}^T \boldsymbol{Y}_i$$

where $\boldsymbol{Y}_i$ denotes the $i$th column of $\boldsymbol{Y}_i$.

In our randomized smoothing setting, we again propose to flip the class of any label with probability $q$, selecting an alternative label uniformly at random from the remaining $K-1$ labels. Assuming that the predicted class label is $i$, we wish to bound the probability that

$$P(\boldsymbol{\alpha}^T \boldsymbol{Y}_i < \boldsymbol{\alpha}^T \boldsymbol{Y}_{i'})$$

for all alternative classes $i'$. By the Chernoff bound, we have that

$$\begin{aligned} \log P(\boldsymbol{\alpha}^T \boldsymbol{Y}_i < \boldsymbol{\alpha}^T \boldsymbol{Y}_{i'}) &= \log P(\boldsymbol{\alpha}^T (\boldsymbol{Y}_i - \boldsymbol{Y}_{i'}) \leq 0) \\ &\leq \min_{t>0} \left\{ \sum_{j=1}^n \log \mathbf{E}\left[ e^{-t\boldsymbol{\alpha}_j (\boldsymbol{Y}_{ji} - \boldsymbol{Y}_{ji'})} \right] \right\}. \end{aligned}$$

The random variable $\boldsymbol{Y}_{ji} - \boldsymbol{Y}_{ji'}$ takes on three different distributions depending on if $y_j = i$, if $y_j = i'$, or if $y_j \neq i$ and $y_j \neq i'$. Specifically, this variable can take on the terms $+1, 0, -1$ with the associated probabilities

$$\begin{aligned} P(\boldsymbol{Y}_{ji} - \boldsymbol{Y}_{ji'} = +1) &= \begin{cases} 1-q & \text{if } y_j = i, \\ q/(K-1) & \text{otherwise.} \end{cases} \\ P(\boldsymbol{Y}_{ji} - \boldsymbol{Y}_{ji'} = -1) &= \begin{cases} 1-q & \text{if } y_j = i', \\ q/(K-1) & \text{otherwise.} \end{cases} \\ P(\boldsymbol{Y}_{ji} - \boldsymbol{Y}_{ji'} = 0) &= \begin{cases} q(K-2)/(K-1) & \text{if } y_j = i \text{ or } y_j = i', \\ 1-2q/(K-1) & \text{otherwise.} \end{cases} \end{aligned}$$

Combining these cases directly into the Chernoff bound gives

$$
\begin{aligned}
&\log P(\boldsymbol{\alpha}^T \mathbf{Y}_i < \boldsymbol{\alpha}^T \mathbf{Y}_{i'}) && \text{(B.1)}\\
\leq \min_{t>0} \Bigg\{ &\sum_{j:y_j=i} \log\left((1-q)e^{-t\boldsymbol{\alpha}_j} + q\frac{K-2}{K-1} + \frac{q}{K-1}e^{t\boldsymbol{\alpha}_j}\right) && \text{(B.2)}\\
&+ \sum_{j:y_j=i'} \log\left(\frac{q}{K-1}e^{-t\boldsymbol{\alpha}_j} + q\frac{K-2}{K-1} + (1-q)e^{t\boldsymbol{\alpha}_j}\right) && \text{(B.3)}\\
&+ \sum_{j:y_j\neq i, y_j\neq i'} \log\left(\frac{q}{K-1}e^{-t\boldsymbol{\alpha}_j} + 1 - 2\frac{q}{K-1} + \frac{q}{K-1}e^{t\boldsymbol{\alpha}_j}\right)\Bigg\}. && \text{(B.4)}
\end{aligned}
$$

Again, this problem is convex in $t$, and so can be solved efficiently using Newton's method. And again since the reverse case can be computed via the same expression we can similarly optimize this in an unconstrained fashion. Specifically, we can do this for every pair of classes $i$ and $i'$, and return the $i$ which gives the smallest lower bound for the worst-case choice of $i'$.

### B.2.3 KL Divergence Bound

To compute actual certification radii, we will derive the KL divergence bound for the the case of $K$ classes. Let $\mu, \rho$ be defined as in Section 3.4, except that as in the previous section when a label is flipped with probability $q$ it is changed to one of the other $K-1$ classes uniformly at random. Let $\mu_i$ and $\rho_i$ refer to the independent measures on each dimension which collectively make up the factorized distributions $\mu$ and $\rho$ (i.e., $\mu(x) = \prod_{i=1}^{d} \mu_i(x)$). Further, let $Y_1^i$ be the $i^{th}$ element of $Y_1$, meaning it is the "original" label which may or may not be flipped when sampling from $\mu$. First noting that each dimension of the distributions $\mu$ and $\rho$ are independent, we have

$$\begin{aligned}
\mathrm{KL}(\rho \parallel \mu) &= \sum_{i=1}^{n} \mathrm{KL}(\rho_i \parallel \mu_i) && \text{(B.5)}\\
&= \sum_{i:\rho_i \neq \mu_i} \mathrm{KL}(\rho_i \parallel \mu_i) && \text{(B.6)}\\
&= r\left(\sum_{j=1}^{K} \rho_i(j) \log\left(\frac{\rho_i(j)}{\mu_i(j)}\right)\right) && \text{(B.7)}\\
&= r\left(\rho_i(Y_1^i) \log\left(\frac{\rho_i(Y_1^i)}{\mu_i(Y_1^i)}\right) + \rho_i(Y_2^i) \log\left(\frac{\rho_i(Y_2^i)}{\mu_i(Y_2^i)}\right)\right) && \text{(B.8)}\\
&= r\left((1-q) \log\left(\frac{1-q}{\frac{q}{K-1}}\right) + \frac{q}{K-1} \log\left(\frac{\frac{q}{K-1}}{1-q}\right)\right) && \text{(B.9)}\\
&= r\left(1 - \frac{Kq}{K-1}\right) \log\left(\frac{(1-q)(K-1)}{q}\right). && \text{(B.10)}
\end{aligned}$$

Plugging in the robustness guarantee (3.3), we have that $g(\mu, \phi) = g(\rho, \phi)$ so long as

$$r \leq \frac{\log(4p(1-p))}{2(1 - \frac{Kq}{K-1}) \log\left(\frac{q}{(1-q)(K-1)}\right)}.$$

Setting $K = 2$ recovers the divergence term (3.4) and the bound (3.5).

## B.3 Description of Label-Flipping Attacks

### B.3.1 Attacks on Undefended Classifiers

Due to the dearth of existing work on label-flipping attacks for deep networks, our attacks on MNIST and Dogfish were quite straightforward; we expect significant improvements could be made to tighten this upper bound.

For Dogfish, we used a pre-trained Inception network [Szegedy et al., 2016] to evaluate the influence of each training point with respect to the loss of each test point [Koh and Liang, 2017]. As in prior work, we froze all but the top layer of the network for retraining. Once we obtained the most influential points, we flipped the first one and recomputed approximate influence using only the top layer for efficiency. After each flip, we recorded which points were classified differently and maintained for each test point the successful attack which required the fewest flips. When this was finished, we also tried the reverse of each attack to see if any of them could be achieved with even fewer flips.

For MNIST we implemented two similar attacks and kept the best attack for each test point. The first attack simply ordered training labels by their $\ell_2$ distance from the test point in feature space, as a proxy for influence. We then tried flipping these one at a time until the prediction changed, and we also tried the reverse. The second attack was essentially the same as the Dogfish attack, ordering the test points by influence. To calculate influence we again assumed a frozen feature map; specifically, using the same notation as Koh and Liang [2017], the influence of flipping the label of a training point $z = (x, y)$ to $z^- = (x, 1 - y)$ on the loss at the test point $z_{\text{test}}$ is:

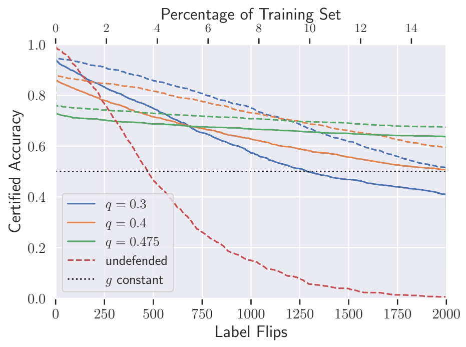
$$\begin{aligned}\frac{dL(z_{\text{test}}, \widehat{\theta}_{\epsilon,z^-,-z})}{d\epsilon} &= \nabla_\theta L(z_{\text{test}}, \widehat{\theta})^T \frac{d\widehat{\theta}_{\epsilon,z^-,-z}}{d\epsilon} \\ &\approx -\nabla_\theta L(z_{\text{test}}, \widehat{\theta})^T H_{\widehat{\theta}}^{-1} \left(\nabla_\theta L(z^-, \widehat{\theta}) - \nabla_\theta L(z, \widehat{\theta})\right).\end{aligned}$$

For logistic regression these values can easily be computed in closed form.

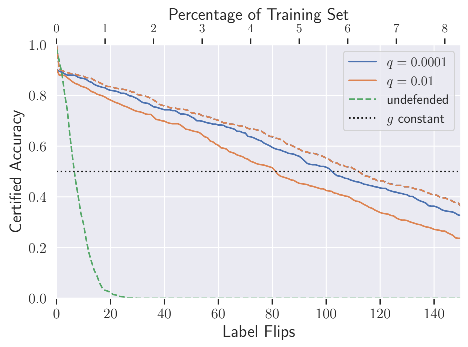
### B.3.2 Attacks on Our Classifier

Recall that our theoretical classifier outputs a prediction based on $P(\boldsymbol{\alpha}^T\mathbf{y} \geq 1/2)$, where the randomness is over the label flips of $\mathbf{y}$. More specifically, the classifier's output is based on a weighted majority vote of "sub-classifiers", each of which is a simple linear classifier which outputs $\mathbf{1}\{\boldsymbol{\alpha}^T\widehat{\mathbf{y}} \geq 1/2\}$ for its own labels $\widehat{\mathbf{y}}$. The sub-classifier's vote is weighted by its probability under the smoothing distribution, which depends only on $\|\mathbf{y} - \widehat{\mathbf{y}}\|_0$ (and is monotonically decreasing in this value). It is clear that the optimal attack to reduce $P(\boldsymbol{\alpha}^T\mathbf{y} \geq 1/2)$ is to flip the labels which will push the inner product $\boldsymbol{\alpha}^T\mathbf{y}$ as much as possible towards the incorrect label: flipping labels by their change to the inner product will add weight to the votes of the most overall number of incorrect sub-classifiers, pushing our smoothed classifier to be incorrect.

Here we make a subtle distinction: while this attack is optimal for the purpose of reducing $P(\boldsymbol{\alpha}^T\mathbf{y} \geq 1/2)$, it is *not necessarily optimal* against our classifier, even though this probability represents how our classifier (theoretically) makes a prediction. This is because in practice, we never actually compute $P(\boldsymbol{\alpha}^T\mathbf{y} \geq 1/2)$. Instead, recall from equation 3.6 that we use the Chernoff inequality to tightly bound this probability. Thus, while the attack described above is optimal for reducing the *true* probability (and therefore the theoretical robustness), it is technically possible that a different attack would cause a looser Chernoff bound, more effectively reducing our bound on the probability. In essence, our attack is optimal for modifying the LHS of equation 3.6, but not necessarily the RHS, which is ultimately how our classifier *actually* makes predictions.



*(a)* Binary MNIST (classes 1 and 7)



*(b)* Dogfish

Figure B.1: MNIST 1/7 and Dogfish certified lower bounds (solid) compared to empirical upper bounds (dashed) of our classifier and the undefended classifier. Our classifier's upper and lower bounds are reasonably close, and they get closer as $q$ decreases. The gap is due to the potential looseness of the Chernoff bound, though in practice we would expect the true robustness to be closer to the upper bound.

With that said, the existence of an attack which causes the Chernoff bound to return a particularly sub-optimal bound seems debatable. So, while we present these results as an empirical upper bound, we believe it would not be inappropriate to also view them as an *approximate* lower bound. Of course, the actual lower bound returned by our classifier is still a genuinely guaranteed certificate. Figure B.1a displays the result of our attack on MNIST 1/7, with the undefended classifier for comparison. Observe that the empirical upper bounds (dashed lines) track the guaranteed lower bounds (solid lines) reasonably closely. The gap is under 10% accuracy and shrinks as the noise $q$ decreases. Further, this empirical robust accuracy outperforms the undefended classifier's empirical robust accuracy by an even larger margin. Figure B.1b presents the same results on the Dogfish dataset. Our empirical attacks had very similar success rates for all values of $q$, so we only plot two values along with the undefended classifier. We again observe a tight correspondence between upper and lower bounds which gets tighter with smaller $q$.

## B.4 Additional Tables of Results

To supplement the line plots, for each dataset and noise parameter we present here precise certified test set accuracy at specific numbers of label flips. When available, for comparison we also provide the undefended classifier's empirical accuracy when subjected to our label-flipping attack as detailed in Appendix B.3.1.

For each number of label flips, the noise hyperparameter setting which results in the highest certified accuracy is in bold.

| **MNIST 1/7** ($n = 13007$, 2 classes) | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **Number of Label Flips** | | | | | | |
| **Noise *q* ↓** | 1 | 10 | 100 | 500 | 1000 | 1500 | 2000 |
| (undefended) | (.9903) | (.9815) | (.9163) | (.4674) | (.1503) | (.0388) | (.0065) |
| 0.3 | **.9399** | **.9320** | **.8918** | **.7470** | .5726 | .4681 | .4089 |
| 0.4 | .8659 | .8571 | .8248 | .7152 | .6283 | .5566 | .5072 |
| 0.45 | .7855 | .7767 | .7540 | .7004 | .6556 | .6218 | .5950 |
| 0.475 | .7294 | .7262 | .7118 | .6873 | **.6674** | **.6503** | **.6378** |

Table B.1: Certified test set accuracy on MNIST 1/7 (Figure 3.1a), with the undefended classifier's empirical robust accuracy for comparison. Random guessing or a constant classifier would attain 50% accuracy.

| **Full MNIST** ($n = 60000$, 10 classes) | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **Number of Label Flips** | | | | | | |
| **Noise *q* ↓** | 1 | 10 | 100 | 200 | 300 | 400 | 500 |
| 0.0125 | .5693 | .5689 | **.5212** | **.4292** | .3333 | .2446 | .1706 |
| 0.025 | **.5713** | **.5701** | .5053 | .4040 | .2999 | .2096 | .1407 |
| 0.05 | .5495 | .5486 | .4954 | .4160 | **.3400** | **.2633** | **.2012** |

Table B.2: Certified test set accuracy on Full MNIST (Figure 3.1b). Random guessing or a constant classifier would attain 10% accuracy.

| **CIFAR10** ($n = 50000$, 10 classes) | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **Number of Label Flips** | | | | | | |
| **Noise *q* ↓** | 1 | 10 | 50 | 100 | 200 | 300 | 400 |
| 0.012 | **.7180** | **.7158** | **.6800** | **.6234** | **.4493** | **.2520** | **.1201** |
| 0.025 | .7068 | .7017 | .6597 | .5949 | .4051 | .1870 | .0548 |
| 0.1 | .7040 | .6876 | .6230 | .5384 | .3019 | .0981 | .0213 |

Table B.3: Certified test set accuracy on CIFAR10 (Figure 3.2). Random guessing or a constant classifier would attain 10% accuracy.

| **Dogfish** ($n$ = 1800, 2 classes) | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **Number of Label Flips** | | | | | | |
| **Noise $q$ ↓** | 1 | 10 | 25 | 50 | 75 | 100 | 150 |
| (undefended) | (.9367) | (.2933) | (.0050) | (.0000) | (.0000) | (.0000) | (.0000) |
| 0.0001 | .8950 | **.8667** | **.8083** | **.7150** | **.6233** | **.5167** | **.3267** |
| 0.001 | .8950 | .8550 | .7967 | .6967 | .5917 | .4750 | .3017 |
| 0.01 | .8800 | .8333 | .7583 | .6617 | .5283 | .4250 | .2367 |
| 0.05 | **.9367** | .7833 | .7033 | .5567 | .4350 | .3200 | .1583 |

Table B.4: Certified test set accuracy on Dogfish (Figure 3.3), with the undefended classifier's empirical robust accuracy for comparison. Random guessing or a constant classifier would attain 50% accuracy.

| **IMDB Sentiment Analysis** ($n$ = 25000, 2 classes) | | | | | | | |
|---|---|---|---|---|---|---|---|
| | **Number of Label Flips** | | | | | | |
| **Noise $q$ ↓** | 1 | 10 | 25 | 50 | 100 | 200 | 300 |
| 0.01 | .6275 | .5980 | .5882 | .5686 | **.5392** | **.4804** | **.4412** |
| 0.025 | .6364 | .6154 | .5944 | .5594 | .5105 | .4406 | .3287 |
| 0.05 | .5878 | .5344 | .5038 | .4656 | .4160 | .3206 | .2519 |
| 0.1 | **.7585** | **.7034** | **.6469** | **.5806** | .4788 | .3263 | .2135 |

Table B.5: Certified test set accuracy on the IMDB Sentiment Analysis dataset (Figure 3.4). Random guessing or a constant classifier would attain 50% accuracy.

## B.5 Additional Plots

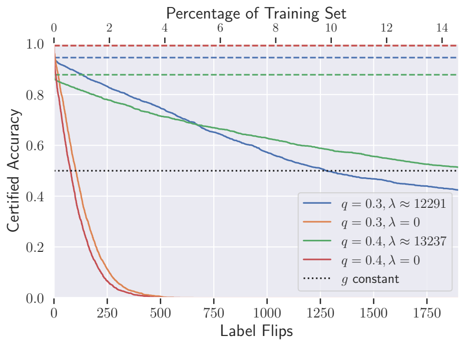


Figure B.2: MNIST 1/7 test set certified accuracy with and without $\ell_2$ regularization in the computation of $\boldsymbol{\alpha}$. Note that the unregularized solution achieves almost 100% non-robust accuracy, but certifies significantly lower robustness. This implies that the "training" process is not robust enough to label noise, hence the lower margin by the ensemble. In comparison, the regularized solution achieves significantly higher margins, at a slight cost in overall accuracy.

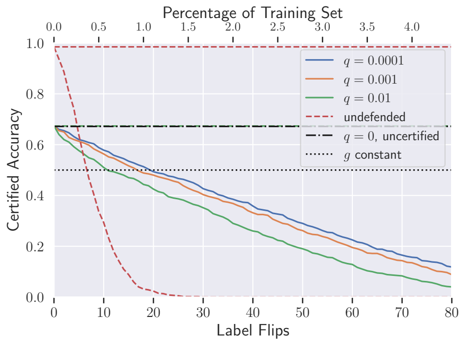


Figure B.3: Dogfish test set certified accuracy using features learned with RICA [Le, 2013]. While not as performant as the pre-trained features, our classifier still achieves reasonable certified accuracy—note that the certified lines are lower bounds, while the undefended line is an upper bound. As demonstrated in the main body, deep unsupervised features significantly boost performance, but require a larger dataset.

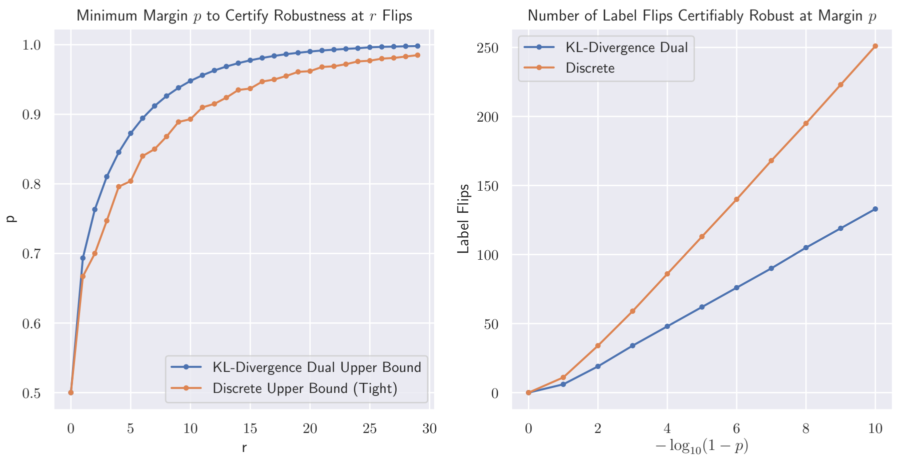


Figure B.4: **Left:** Required margin $p$ to certify a given number of label flips using the generic KL bound (3.5) versus the tight discrete bound (3.2). **Right:** The same comparison, but inverted, showing the certifiable robustness for a given margin. The tight bound certifies robustness to approximately twice as many label flips. Both plots are with $q = 0.4$.

# Appendix C

# Appendix for Chapter 4

## C.1 Additional Notation for the Appendix

To avoid overloading, we use $\phi, F$ for the standard Gaussian PDF and CDF respectively. We write $\mathcal{S}^c$ to denote the set complement of a set $\mathcal{S}$. We write $|\cdot|$ to denote entrywise absolute value.

## C.2 Proof of Proposition Proposition 4.4.1

Recall the IRM objective:

$$\begin{aligned} \min_{\Phi,\beta} \quad & \mathbb{E}_{(x,y)\sim p(x,y)}[-\log \sigma(y \cdot \widehat{\beta}^\top \Phi(x))] \\ \text{subject to} \quad & \frac{\partial}{\partial \widehat{\beta}} \mathbb{E}_{(x,y)\sim p^e}[-\log \sigma(y \cdot \widehat{\beta}^\top \Phi(x))] = 0. \; \forall e \in \mathcal{E}. \end{aligned}$$

Concretely, we represent $\Phi$ as some parametrized function $\Phi_\theta$, over whose parameters $\theta$ we then optimize. The derivative of the negative log-likelihood for logistic regression with respect to the $\beta$ coefficients is well known:

$$\frac{\partial}{\partial \widehat{\beta}} \left[ -\log \sigma(y \cdot \widehat{\beta}^\top \Phi_\theta(x)) \right] = (\sigma(\widehat{\beta}^\top \Phi_\theta(x)) - \mathbf{1}\{y = 1\}) \Phi_\theta(x). \tag{C.1}$$

Suppose we recover the true invariant features $\Phi_\theta(x) = \begin{bmatrix} z_c \\ \mathbf{0} \end{bmatrix}$ and coefficients $\widehat{\beta} = \begin{bmatrix} \beta \\ \mathbf{0} \end{bmatrix}$ (in other words, we allow for the introduction of new features). Then the

IRM constraint becomes:

$$0 = \frac{\partial}{\partial \widehat{\beta}} \mathbb{E}_{(x,y)\sim p^e}[-\log \sigma(y \cdot \widehat{\beta}^\top \Phi_\theta(x))] \tag{C.2}$$

$$= \int_{\mathcal{Z}} p^e(z_c) \sum_{y\in\{\pm 1\}} p^e(y \mid z_c) \frac{\partial}{\partial \widehat{\beta}} \left[ -\log \sigma(y \cdot \beta^\top z_c) \right] \, dz_c \tag{C.3}$$

$$= \int_{\mathcal{Z}} p^e(z_c) \Phi_\theta(x) \left[ \sigma(\widehat{\beta}^\top z_c)(\sigma(\widehat{\beta}^\top z_c) - 1) + (1 - \sigma(\widehat{\beta}^\top z_c))\sigma(\widehat{\beta}^\top z_c) \right] \, dz_c. \tag{C.4}$$

Since $\widehat{\beta}$ is constant across environments, this constraint is clearly satisfied for every environment, and is therefore also the minimizing $\widehat{\beta}$ for the training data as a whole.

Considering now the derivative with respect to the featurizer $\Phi_\theta$:

$$\frac{\partial}{\partial \theta} \left[ -\log \sigma(y \cdot \widehat{\beta}^\top \Phi_\theta(x)) \right] = (\sigma(\widehat{\beta}^\top \Phi_\theta(x)) - \mathbf{1}\{y = 1\}) \frac{\partial}{\partial \theta} \widehat{\beta}^\top \Phi_\theta(x). \tag{C.5}$$

Then the derivative of the loss with respect to these parameters is

$$\int_{\mathcal{Z}} p^e(z_c) \left( \frac{\partial}{\partial \theta} \widehat{\beta}^\top \Phi_\theta(x) \right) \left[ \sigma(\beta^\top z_c)(\sigma(\beta^\top z_c) - 1) + (1 - \sigma(\beta^\top z_c))\sigma(\beta^\top z_c) \right] dz_c \tag{C.6}$$

and is equal to 0. So, the invariant classifier is a stationary point with respect to the feature map parameters as well.

## C.3 Results from Section Section 4.5

### C.3.1 Proof of Theorem 4.5.1

We begin by formally stating the non-degeneracy condition. Consider any environmental mean $\mu_e$, and suppose it can be written as a linear combination of the others means with coefficients $\alpha^e$:

$$\mu_e = \sum_i \alpha_i^e \mu_i. \tag{C.7}$$

Then the environments are considered non-degenerate if the following inequality holds for any such set of coefficients:

$$\sum_i \alpha_i^e \neq 1, \tag{C.8}$$

and furthermore that the following ratio is different for at least two different environments $a, b$:

$$\exists \alpha^a, \alpha^b. \ \frac{\sigma_a^2 - \sum_i \alpha_i^a \sigma_i^2}{1 - \sum_i \alpha_i^a} \neq \frac{\sigma_b^2 - \sum_i \alpha_i^b \sigma_i^2}{1 - \sum_i \alpha_i^b}. \tag{C.9}$$

The first inequality says that none of the environmental means are are an affine combination of the others; in other words, they lie in *general linear position*, which is the same requirement as Arjovsky et al. [2019]. The other inequality is a similarly lax non-degeneracy requirement regarding the relative scale of the variances. It is clear that the set of environmental parameters that do not satisfy Equations Equation (C.8) and Equation (C.9) has measure zero under any absolutely continuous density, and similarly, if $E \leq d_e$ then the environmental means will be linearly independent almost surely.

We can now proceed with the proof, beginning with some helper lemmas:

**Lemma C.3.1.** *Suppose we observe $E$ environments $\mathcal{E} = \{e_1, e_2, \ldots, e_E\}$, each with environmental mean of dimension $d_e \geq E$, such that all environmental means are linearly independent. Then there is a unique unit-norm vector $p$ such that*

$$p^\top \mu_e = \sigma_e^2 \tilde{\mu} \ \ \forall e \in \mathcal{E}, \tag{C.10}$$

*where $\tilde{\mu}$ is the largest scalar which admits such a solution.*

*Proof.* Let $v_1, v_2, \ldots, v_E$ be a set of basis vectors for $\text{span}\{\mu_1, \mu_2, \ldots, \mu_E\}$. Each mean can then be expressed as a combination of these basis vectors: $u_i = \sum_{j=1}^E \alpha_{ij} v_j$. Since the means are linearly independent, we can combine these coefficients into a single invertible matrix

$$U = \begin{bmatrix} \alpha_{11} & \alpha_{21} & \ldots & \alpha_{E1} \\ \alpha_{12} & \alpha_{22} & \ldots & \alpha_{E2} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{1E} & \alpha_{2E} & \ldots & \alpha_{EE} \end{bmatrix}. \tag{C.11}$$

We can then combine the constraints (Equation (C.10)) as

$$U^\top p_\alpha = \boldsymbol{\sigma} \triangleq \begin{bmatrix} \sigma_1^2 \\ \sigma_2^2 \\ \vdots \\ \sigma_E^2 \end{bmatrix}, \tag{C.12}$$

where $p_\alpha$ denotes our solution expressed in terms of the basis vectors $\{v_i\}_{i=1}^E$. This then has the solution

$$p_\alpha = U^{-T}\boldsymbol{\sigma}. \tag{C.13}$$

This defines the entire space of solutions, which consists of $p_\alpha$ plus any element of the remaining $(d_e - E)$-dimensional orthogonal subspace. However, we want $p$ to be unit-norm—observe that the current vector solves Equation (C.10) with $\tilde{\mu} = 1$, which means that after normalizing we get $\tilde{\mu} = \frac{1}{\|p_\alpha\|}$. Adding any element of the orthogonal subspace would only increase the norm of $p$, decreasing $\tilde{\mu}$. Thus, the unique maximizing solution is

$$p_\alpha = \frac{U^{-T}\boldsymbol{\sigma}}{\|U^{-T}\boldsymbol{\sigma}\|}, \quad \text{with} \quad \tilde{\mu} = \frac{1}{\|U^{-T}\boldsymbol{\sigma}\|}. \tag{C.14}$$

Finally, $p_\alpha$ has to be rotated back into the original space by defining $p = \sum_{i=1}^E p_{\alpha i} v_i$. $\square$

**Lemma C.3.2.** *Assume $f$ is linear. Suppose we observe $E \leq d_e$ environments whose means are linearly independent. Then there exists a linear $\Phi$ with* $\text{rank}(\Phi) = d_c + d_e + 1 - E$ *whose output depends on the environmental features, yet the optimal classifier on top of $\Phi$ is invariant.*

*Proof.* We will begin with the case when $E = d_e$ and then show how to modify this construction for when $E < d_e$. Consider defining

$$\Phi = \begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix} \circ f^{-1} \tag{C.15}$$

with

$$M = \begin{bmatrix} — & p^\top & — \\ — & 0 & — \\ & \vdots & \\ — & 0 & — \end{bmatrix}. \tag{C.16}$$

Here, $p \in \mathbb{R}^{d_c}$ is defined as the unit-norm vector solution to

$$p^\top \mu_e = \sigma_e^2 \tilde{\mu} \quad \forall e \tag{C.17}$$

such that $\tilde{\mu}$ is maximized—such a vector is guaranteed to exist by Lemma C.3.1. Thus we get $\Phi(x) = \begin{bmatrix} z_c \\ p^\top z_e \end{bmatrix}$, which is of rank $d_c + 1$ as desired. Define $\tilde{z}_e = p^\top z_e$, which means that $\tilde{z}_e \mid y \sim \mathcal{N}(y \cdot \sigma_e^2 \tilde{\mu}, \sigma_e^2)$. For each environment we have

$$p(y \mid z_c, \tilde{z}_e) = \frac{p(z_c, \tilde{z}_e, y)}{p(z_c, \tilde{z}_e)} \tag{C.18}$$

$$= \frac{\sigma(y \cdot \beta_c^\top z_c) p(\tilde{z}_e \mid y \cdot \sigma_e^{e2} \tilde{\mu}, \sigma_e^2)}{[\sigma(y \cdot \beta_c^\top z_c) p(\tilde{z}_e \mid y \cdot \sigma_e^{e2} \tilde{\mu}, \sigma_e^2) + \sigma(-y \cdot \beta_c^\top z_c) p(\tilde{z}_e \mid -y \cdot \sigma_e^{e2} \tilde{\mu}, \sigma_e^2)]} \tag{C.19}$$

$$= \frac{\sigma(y \cdot \beta_c^\top z_c) \exp(y \cdot \tilde{z}_e \tilde{\mu})}{[\sigma(y \cdot \beta_c^\top z_c) \exp(y \cdot \tilde{z}_e \tilde{\mu}) + \sigma(-y \cdot \beta_c^\top z_c) \exp(-y \cdot \tilde{z}_e \tilde{\mu})]} \tag{C.20}$$

$$= \frac{1}{1 + \exp(-y \cdot (\beta_c^\top z_c + 2\tilde{z}_e \tilde{\mu}))}. \tag{C.21}$$

The log-odds of $y$ is linear in these features, so the optimal classifier is

$$\widehat{\beta} = \begin{bmatrix} \beta_c \\ 2\tilde{\mu} \end{bmatrix}, \tag{C.22}$$

which is the same for all environments.

Now we show how to modify this construction for when $E < d_e$. If we remove one of the environmental means, $\Phi$ trivially maintains its feasibility. Note that since they are linearly independent, the mean which was removed has a component in a direction orthogonal to the remaining means. Call this component $p'$, and consider redefining $M$ as

$$M = \begin{bmatrix} — & p^\top & — \\ — & p'^\top & — \\ — & 0 & — \\ & \vdots & \\ — & 0 & — \end{bmatrix}. \tag{C.23}$$

The distribution of $\tilde{z}_e$ in each of the remaining dimensions is normal with mean 0, which means a corresponding coefficient of 0 is optimal for all environments. So the classifier $\widehat{\beta}$ remains optimal for all environments, yet we've added another row to $M$ which increases the dimensionality of its span, and therefore the rank of $\Phi$, by 1. Working backwards, we can repeat this process for each additional mean, such that $\text{rank}(\Phi) = d_c + 1 + (d_e - E)$, as desired. Note that for $E = 1$ any $\Phi$ will be vacuously feasible. □

**Lemma C.3.3.** *Suppose we observe $E$ environments $\mathcal{E} = \{e_1, e_2, \dots, e_E\}$ whose parameters satisfy the non-degeneracy conditions (Equation (C.8), Equation (C.9)). Let $\Phi(x) = Az_c + Bz_e$ be any feature vector which is a linear function of the invariant and environmental features, and suppose the optimal $\widehat{\beta}$ on top of $\Phi$ is invariant. If $E > d_e$, then $\widehat{\beta} = 0$ or $B = 0$.*

*Proof.* Write $\Phi = [A|B]$ where $A \in \mathbb{R}^{d\times d_c}, B \in \mathbb{R}^{d\times d_e}$ and define

$$\bar{\mu}_e = \Phi \begin{bmatrix} \mu_c \\ \mu_e \end{bmatrix} = A\mu_c + B\mu_e, \tag{C.24}$$

$$\bar{\Sigma}_e = \Phi \begin{bmatrix} \sigma_c^2 I_{d_c} & 0 \\ 0 & \sigma_e^2 I_{d_e} \end{bmatrix} \Phi^\top = \sigma_c^2 AA^\top + \sigma_e^2 BB^\top. \tag{C.25}$$

Without loss of generality we assume $\bar{\Sigma}$ is invertible (if it is not, we can consider just the subspace in which it is—outside of this space, the features have no variance and therefore cannot carry information about the label). By Lemma C.6.2, the optimal coefficient for each environment is $2\bar{\Sigma}_e^{-1}\bar{\mu}_e$. In order for this vector to be invariant, it must be the same across environments; we write it as a constant $\widehat{\beta}$. Suppose $\bar{\mu}_e = 0$ for some environment $e$—then the claim is trivially true with $\widehat{\beta} = 0$. We therefore proceed under the assumption that $\bar{\mu}_e \neq 0 \ \forall e \in \mathcal{E}$.

With this fact, we have that $\forall e \in \mathcal{E}$,

$$\begin{aligned} &\widehat{\beta} = 2(\sigma_c^2 AA^\top + \sigma_e^2 BB^\top)^{-1}(A\mu_c + B\mu_e) \\ \iff &(\sigma_c^2 AA^\top + \sigma_e^2 BB^\top)\widehat{\beta} = 2A\mu_c + 2B\mu_e \\ \iff &\sigma_e^2 BB^\top \widehat{\beta} - 2B\mu_e = 2A\mu_c - \sigma_c^2 AA^\top \widehat{\beta}. \end{aligned} \tag{C.26}$$

Define the vector $\mathbf{v} = 2A\mu_c - \sigma_c^2 AA^\top \widehat{\beta}$. We will show that for any $\widehat{\beta}$, $A$, with probability 1 only $B = 0$ can satisfy Equation (C.26) for every environment. If $E > d_e$, then there exists at least one environmental mean which can be written as a linear combination of the others. Without loss of generality, denote the parameters of this environment as $(\bar{\mu}, \bar{\sigma}^2)$ and write $\bar{\mu} = \sum_{i=1}^{d_e} \alpha_i \mu_i$. However, note that by assumption the means lie in general linear position, and therefore we actually have at least $d_e$ sets of coefficients $\alpha$ for which this holds. Rearranging Equation (C.26), we get

$$\bar{\sigma}^2 BB^\top \widehat{\beta} - \mathbf{v} = 2B\bar{\mu} \tag{C.27}$$

$$= \sum_{i=1}^{d_e} \alpha_i 2B\mu_i \tag{C.28}$$

$$= \sum_{i=1}^{d_e} \alpha_i \left[\sigma_i^2 BB^\top \widehat{\beta} - \mathbf{v}\right], \tag{C.29}$$

and rearranging once more yields

$$\left(\bar{\sigma}^2 - \sum \alpha_i \sigma_i^2\right) BB^\top \widehat{\beta} = \left(1 - \sum \alpha_i\right) \mathbf{v}. \tag{C.30}$$

By assumption, $(1 - \sum \alpha_i)$ is non-zero. We can therefore rewrite this as

$$\widehat{\alpha} BB^\top \widehat{\beta} = \mathbf{v}, \tag{C.31}$$

where $\widehat{\alpha} = \frac{\bar{\sigma}^2 - \sum \alpha_i \sigma_i^2}{1 - \sum \alpha_i}$ is a scalar. As the vectors $BB^\top \widehat{\beta}$ and $\mathbf{v}$ are both independent of the environment, this can only hold true if $\widehat{\alpha}$ is fixed for all environments or if both $BB^\top \widehat{\beta}, \mathbf{v}$ are 0. The former is false by assumption, so the the latter must hold.

As a result, we see that Equation (C.26) reduces to

$$B\mu_e = 0 \quad \forall e \in \mathcal{E}. \tag{C.32}$$

As the span of the observed $\mu_e$ is all of $\mathbb{R}^{d_e}$, this is only possible if $B = 0$. □

We are now ready to prove the main claim. We restate the theorem here for convenience:

**Theorem 4.5.1** (Linear case)**.** *Assume $f$ is linear. Suppose we observe $E$ training environments. Then the following hold:*

1. *Suppose $E > d_e$. Under mild non-degeneracy conditions, any linear featurizer $\Phi$ with an invariant optimal regression vector $\widehat{\beta}$ uses only invariant features, and it therefore has identical risk on all possible environments.*
2. *If $E \leq d_e$ and the environmental means $\mu_e$ are linearly independent, then there exists a linear $\Phi$ with $\operatorname{rank}(\Phi) = d_c + d_e + 1 - E$ whose output depends on the environmental features, yet the optimal classifier on top of $\Phi$ is invariant. Further, both the logistic and 0-1 risks of this $\Phi$ and its corresponding $\widehat{\beta}$ are strictly lower than those of the invariant classifier.*

*Proof.*
1. Since $\Phi, f$ are linear, we can write $\Phi(x) = Az_c + Bz_e$ for some matrices $A, B$. Assume the non-degeneracy conditions (Equation (C.8), Equation (C.9)) hold. By Lemma C.3.3, one of $B = 0$ or $\widehat{\beta} = 0$ holds. Thus, $\Phi, \widehat{\beta}$ uses only invariant features. Since the joint distribution $p^e(z_c, y)$ is invariant, this classifier has identical risk across all environments.
2. The existence of such a classifier is proven by Lemma C.3.2. It remains to show that the risk of this discriminator is lower than that of the invariant classifier. Observe that these features are non-degenerate independent random

variables with support over all of $\mathbb{R}$, and therefore by Lemma C.6.1, dropping the $\tilde{z}_e$ term and using

$$\Phi(x) = [z_c], \quad \widehat{\beta} = \begin{bmatrix} \beta_c \\ \beta_0 \end{bmatrix} \tag{C.33}$$

results in strictly higher risk. The proof is completed by noting that this definition is precisely the invariant classifier.

□

### C.3.2 Experiments for Theorem 4.5.1

To corroborate our theoretical findings, we run an experiment on data drawn from our model to see at what point IRM is able to recover a generalizing classifier. We generated data precisely according to our model in the linear setting, with $d_c = 3, d_e = 6$. The environmental means were drawn from a multivariate Gaussian prior; we randomly generated the invariant parameters and the parameters of the prior such that using the invariant features gave reasonable accuracy ($71.9\%$) but the environmental features would allow for almost perfect accuracy on in-distribution test data ($99.8\%$). Thus, the goal was to see if IRM could successfully learn a classifier which ignores meaningful covariates $z_e$, to the detriment of its training performance but to the benefit of OOD generalization. We chose equal class marginals ($\eta = 0.5$).

Figure C.1 shows the result of five runs of IRM, each with different environmental parameters but the same invariant parameters (the training data itself was redrawn for each run). We found that optimizing for the IRM objective was quite unstable, frequently collapsing to the ERM solution unless $\lambda$ and the optimizer learning rate were carefully tuned. This echoes the results of Krueger et al. [2020] who found that tuning $\lambda$ during training to specific values at precisely the right time is essential for good performance. To prevent collapse, we kept the same environmental prior and found a single setting for $\lambda$ and the learning rate which resulted in reasonable performance across all five runs. At test time, we evaluated the trained classifiers on additional, unseen environments whose parameters were drawn from the same prior. To simulate distribution shift, we evaluated the classifiers on the same data but with the environmental means negated. Thus the correlations between the environmental features $z_e$ and the label $y$ were reversed.

Observe that the results closely track the expected outcome according to Theorem 4.5.1: up until $E = d_e$, IRM essentially matches ERM in performance both in-distribution and under distribution shift. As soon as we cross that threshold of observed environments, the classifier learned via IRM begins to perform drastically better under distribution shift, behaving more like the optimal invariant classifier.

We did however observe that occasionally the invariant solution would be found after only $E = d_e = 6$ environments; we conjecture that this is because at this point the feasible-yet-not-invariant classifier with lower objective value presented in Theorem 4.5.1 is precisely a single point, as opposed to a multi-dimensional subspace, and therefore might be difficult for the optimizer to find.

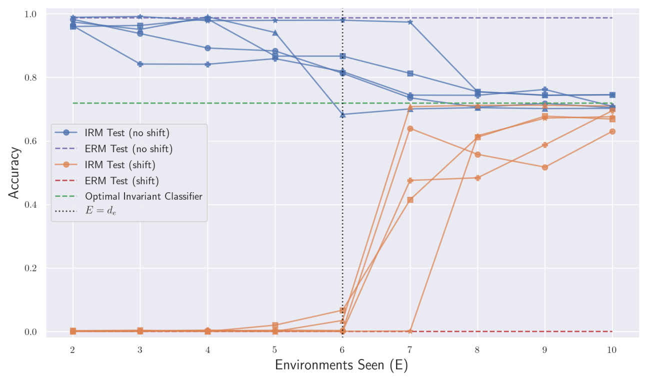


Figure C.1: Performance of classifiers learned with IRM (5 different runs) and ERM (dashed lines) on test distributions where the correlation between environmental features and the label is consistent (no shift) or reversed (shift). The dashed green line is the performance of the optimal invariant classifier. Observe that up until $E = d_e$, IRM consistently returns a classifier with performance similar to ERM: good generalization without distribution shift, but catastrophic failure when the correlation is reversed. In contrast, once $E > d_e$, IRM is able to recover a $\Phi, \widehat{\beta}$ with performance similar to that of the causal classifier.

### C.3.3 Proof of Theorem 4.5.3

**Theorem 4.5.3.** *Suppose we observe $E \leq d_e$ environments, such that all environmental means are linearly independent. Then there exists a feasible $\Phi, \widehat{\beta}$ which uses* only environmental features *and achieves lower 0-1 risk than the invariant classifier on every environment $e$ such that $\sigma_e \tilde{\mu} > \sigma_c^{-1} \|\mu_c\|$ and $2\sigma_e \tilde{\mu} \sigma_c^{-1} \|\mu_c\| \geq |\beta_0|$.*

*Proof.* We consider the non-invariant classifier constructed as described in Lemma C.3.2, but dropping the invariant features and coefficients. By Lemma C.6.2, the optimal coefficients for the invariant and non-invariant classifiers are

$$\widehat{\beta}_{caus} = \begin{bmatrix} 2\sigma_c^{-2}\mu_c \\ \beta_0 \end{bmatrix} \qquad \text{and} \qquad \widehat{\beta}_{non-caus} = \begin{bmatrix} 2\tilde{\mu} \\ \beta_0 \end{bmatrix}, \tag{C.34}$$

respectively. Therefore, the 0-1 risk of the invariant classifier is precisely

$$\eta \mathbb{P}(2\sigma_c^{-2}\mu_c^\top z_c + \beta_0 < 0) + (1-\eta)\mathbb{P}(-2\sigma_c^{-2}\mu_c^\top z_c + \beta_0 > 0) \tag{C.35}$$

$$= \eta F\left(-\sigma_c^{-1}\|\mu_c\| - \frac{\beta_0 \sigma_c}{2\|\mu_c\|}\right) + (1-\eta) F\left(-\sigma_c^{-1}\|\mu_c\| + \frac{\beta_0 \sigma_c}{2\|\mu_c\|}\right), \tag{C.36}$$

where $F$ is the Gaussian CDF. By the same reasoning, the 0-1 risk of the non-invariant classifier is

$$\eta F\left(-\sigma_e \tilde{\mu} - \frac{\beta_0}{2\sigma_e \tilde{\mu}}\right) + (1-\eta) F\left(-\sigma_e \tilde{\mu} + \frac{\beta_0}{2\sigma_e \tilde{\mu}}\right). \tag{C.37}$$

Define $\alpha = \sigma_c^{-1}\|\mu_c\|$ and $\gamma = \sigma_e \tilde{\mu}$. By monotonicity of the Gaussian CDF, the former risk is higher than the latter if

$$\alpha + \frac{\beta_0}{2\alpha} \leq \gamma + \frac{\beta_0}{2\gamma}, \tag{C.38}$$

$$\alpha - \frac{\beta_0}{2\alpha} < \gamma - \frac{\beta_0}{2\gamma}. \tag{C.39}$$

Without loss of generality, we will prove these inequalities for $\beta_0 \geq 0$; an identical argument proves it for $\beta_0 < 0$ but with the '$\leq$' and '$<$' swapped.

Suppose $\gamma > \alpha$ (the first condition). Then Equation (C.39) is immediate. Finally, for Equation (C.38), observe that

$$\gamma + \frac{\beta_0}{2\gamma} \geq \alpha + \frac{\beta_0}{2\alpha} \tag{C.40}$$

$$\iff \gamma - \alpha \geq \frac{\beta_0}{2\alpha} - \frac{\beta_0}{2\gamma} = \frac{(\gamma-\alpha)\beta_0}{2\gamma\alpha} \tag{C.41}$$

$$\iff 2\gamma\alpha \geq \beta_0, \tag{C.42}$$

which is the second condition. □

### C.3.4 Simulations of Magnitude of Environmental Features

As discussed in Section 4.5, analytically quantifying the solution $\tilde{\mu}$ to the equation in Lemma C.3.1 is difficult; instead, we present simulations to give a sense of how often these conditions would hold in practice.

For each choice of environmental dimension $d_e$, we generated a "base" correlation $b \sim \mathcal{N}(0, I_{d_e})$ as the mean of the prior over environmental means $\mu_e$. Each of these $\mu_e$ was then drawn from $\mathcal{N}(b, 4I_{d_e})$—thus, while they all came from the same prior, the noise in the draw of each $\mu_e$ was significantly larger than the bias induced by the prior. We then solved for the precise value $\sigma_e \tilde{\mu}$, with the same variance $\sigma_e^2$ for all environments, chosen as a hyperparameter. The shaded area represents a 95% confidence interval over 20 runs.

The dotted lines are $\sqrt{d_c}$. If we imagine the invariant parameters are drawn from a standard Gaussian prior, then this is precisely $\mathbb{E}[\sigma_c^{-1} \|\mu_c\|]$. Thus, the point where $\sigma_e \tilde{\mu}$ crosses these dotted lines is approximately how many environments would need to be observed before the non-invariant classifier has higher risk than the invariant classifier. We note that this value is quite large, on the order of $d_e - d_c$.

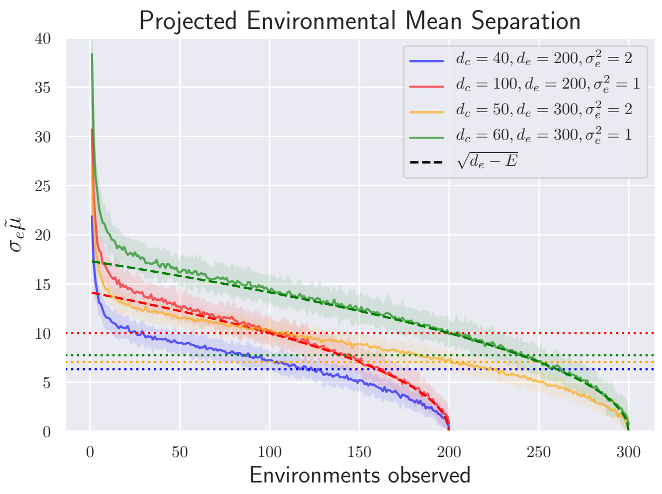


Figure C.2: Simulations to evaluate $\sigma_e \tilde{\mu}$ for varying ratios of $\frac{d_e}{d_c}$. When $\sigma_e^2 = 1$, the value closely tracks $\sqrt{d_e - E}$, giving a crossover point of $d_e - d_c$. These results imply the conditions of Theorem 4.5.3 are very likely to hold in the high-dimensional setting.

## C.4 Theorem 4.6.3 and Discussion

### C.4.1 Proof of Theorem 4.6.3

We again begin with helper lemmas.

Our classifier $\Phi$ is constructed to recover the environmental features only if they fall within a set $\mathcal{B}^c$. The following lemma shows that since only the environmental features contribute to the gradient penalty, the penalty can be bounded as a function of the measure and geometry of that set. This is used together with Lemmas Lemma C.6.3 and Lemma C.6.4 to bound the overall penalty of our constructed classifier.

**Lemma C.4.1.** *Suppose we observe environments $\mathcal{E} = \{e_1, e_2, \ldots\}$. Given a set $\mathcal{B} \subseteq \mathbb{R}^{d_e}$, consider the classifier defined by Equation (C.61). Then for any environment $e$, the penalty term of this classifier in Equation (4.5) is bounded as*

$$\|\nabla_{\widehat{\beta}}\mathcal{R}^e(\Phi, \widehat{\beta})\|^2 \leq \left\| \mathbb{P}(z_e \in \mathcal{B}^c)\mathbb{E}[|z_e| \mid z_e \in \mathcal{B}^c] \right\|_2^2. \tag{C.43}$$

*Proof.* We write out the precise form of the gradient for an environment $e$:

$$\nabla_{\widehat{\beta}}\mathcal{R}^e(\Phi, \widehat{\beta}) = \int_{\mathcal{Z}_c \times \mathcal{Z}_e} \Bigg[ p^e(z_c, z_e) \left( \sigma(\widehat{\beta}^\top \Phi(f(z_c, z_e))) - p^e(y = 1 \mid z_c, z_e) \right) \tag{C.44}$$

$$\Phi(f(z_c, z_e)) \Bigg] \, d(z_c, z_e). \tag{C.45}$$

Observe that since $z_c \perp\!\!\!\perp z_e \mid y$, the optimal invariant coefficients are unchanged, and therefore the gradient in the invariant dimensions is 0. We can split the gradient in the environmental dimensions into two integrals:

$$\int_{\mathcal{Z}_c \times \mathcal{B}} p^e(z_c, z_e) \left[ \sigma(\beta_c^\top z_c + \beta_0) - p^e(y = 1 \mid z_c, z_e) \right] [0] \, d(z_c, z_e) \tag{C.46}$$

$$+ \int_{\mathcal{Z}_c \times \mathcal{B}^c} p^e(z_c, z_e) \left[ \sigma(\beta_c^\top z_c + \beta_{e;\text{ERM}}^\top z_e + \beta_0) - \sigma(\beta_c^\top z_c + \beta_e^\top z_e + \beta_0) \right] \times [z_e] \, d(z_c, z_e). \tag{C.47}$$

Since the features are 0 within $\mathcal{B}$, the first term reduces to 0. For the second term, note that $\forall\, x, y \in \mathbb{R},\ |\sigma(x) - \sigma(y)| \leq 1$, and therefore

$$|\nabla_{\widehat{\beta}}\mathcal{R}^e(\Phi, \widehat{\beta})| \leq \int_{\mathcal{Z}_c \times \mathcal{B}^c} p^e(z_c, z_e)[|z_e|] \, d(z_c, z_e). \tag{C.48}$$

We can marginalize out $z_c$, and noting that we want to bound the squared norm,

$$\|\nabla_{\widehat{\beta}}\mathcal{R}^e(\Phi, \widehat{\beta})\|^2 \leq \left\| \int_{\mathcal{B}^c} p^e(z_e)[|z_e|] \, dz_e \right\|_2^2 \tag{C.49}$$

$$= \left\| \mathbb{P}(z_e \in \mathcal{B}^c)\mathbb{E}[|z_e| \mid z_e \in \mathcal{B}^c] \right\|_2^2. \tag{C.50}$$

□

This next lemma says that if the environmental mean of the test distribution is sufficiently separated from each of the training means, with high probability a sample from this distribution will fall outside of $\mathcal{B}_r$, and therefore $\Phi_\epsilon, \widehat{\beta}$ will be equivalent to the ERM solution.

**Lemma C.4.2.** *For a set of $E$ environments $\mathcal{E} = \{e_1, e_2, \ldots, e_E\}$ and any $\epsilon > 1$, construct $\mathcal{B}_r$ as in Equation (C.60) and define $\Phi_\epsilon$ using $\mathcal{B}_r$ as in Equation (C.61). Suppose we now test on a new environment with parameters $(\mu_{E+1}, \sigma^2_{E+1})$, and assume Equation (C.55) holds with parameter $\delta$. Define $k = \min_{e\in\mathcal{E}} \frac{\sigma_e^2}{\sigma^2_{E+1}}$. Then with probability $\geq 1 - \frac{2E}{\sqrt{k\pi}\delta}\exp\{-k\delta^2\}$ over the draw of an observation from this new environment, we have*

$$\Phi_\epsilon(x) = f^{-1}(x) = \begin{bmatrix} z_c \\ z_e \end{bmatrix}. \tag{C.51}$$

*Proof.* By Equation (C.55) our new environmental mean is sufficiently far away from all the label-conditional means of the training environments. In particular, for any environment $e \in \mathcal{E}$ and any label $y \in \{\pm 1\}$, the $\ell_2$ distance from that mean to $\mu_{E+1}$ is at least $(\sqrt{\epsilon} + \delta)\sigma_e\sqrt{d_e}$.

Recall that $\mathcal{B}_r$ is the union of balls $\pm B_e$, where $B_e$ is the ball of $\ell_2$ radius $\sqrt{\epsilon\sigma_e^2 d_e}$ centered at $\mu_e$. For each environment $e$, consider constructing the halfspace which is perpendicular to the line connecting $\mu_e$ and $\mu_{E+1}$ and tangent to $B_e$. This halfspace fully contains $B_e$, and therefore the measure of $B_e$ is upper bounded by that of the halfspace.

By rotational invariance of the Gaussian distribution, we can rotate this halfspace into one dimension and the measure will not change. The center of the ball is $(\sqrt{\epsilon}+\delta)\sigma_e\sqrt{d_e}$ away from the mean $\mu_{E+1}$, so accounting for its radius, the distance from the mean to the halfspace is $\delta\sigma_e\sqrt{d_e}$. The variance of the rotated distribution one dimension is $\sigma^2_{E+1}d_e$, so the measure of this halfspace is upper bounded by

$$1 - \Phi\left(\frac{\delta\sigma_e\sqrt{d_e}}{\sqrt{\sigma^2_{E+1}d_e}}\right) \leq \Phi\left(-\sqrt{k}\delta\right) \tag{C.52}$$

$$\leq \frac{1}{\sqrt{k\pi}\delta}\exp\{-k\delta^2\}, \tag{C.53}$$

using results from Kschischang [2017]. There are $2E$ such balls comprising $\mathcal{B}_r$, which can be combined via union bound. □

With these two lemmas, we now state the full version of Theorem 4.6.3, with the main difference being that it allows for any environmental variance.

**Theorem C.4.3** (Non-linear case, full)**.** *Suppose we observe $E$ environments $\mathcal{E} = \{e_1, e_2, \dots, e_E\}$. Then, for any $\epsilon > 1$, there exists a featurizer $\Phi_\epsilon$ which, combined with the ERM-optimal classifier $\widehat{\beta} = [\beta_c, \beta_{e;ERM}, \beta_0]^\top$, satisfies the following properties, where we define $p_{\epsilon,d_e} := \exp\{-d_e \min((\epsilon - 1), (\epsilon - 1)^2)/8\}$:*

1. *Define $\sigma^2_{\max} = \max_e \sigma^2_e$. Then the regularization term of $\Phi_\epsilon, \widehat{\beta}$ is bounded as*

$$\frac{1}{E} \sum_{e \in \mathcal{E}} \|\nabla_{\widehat{\beta}} \mathcal{R}^e(\Phi_\epsilon, \widehat{\beta})\|^2 \in \mathcal{O}\left(p^2_{\epsilon,d_e} \left[\epsilon d_e \sigma^4_{\max} \exp\{2\epsilon\sigma^2_{\max}\} + \overline{\|\mu\|^2}\right]\right). \tag{C.54}$$

2. *$\Phi_\epsilon, \widehat{\beta}$ exactly matches the invariant classifier on at least a $1 - p_{\epsilon,d_e}$ fraction of the training set. On the remaining inputs, it matches the ERM-optimal solution.*

*Further, for any test distribution with environmental parameters $(\mu_{E+1}, \sigma^2_{E+1})$, suppose the environmental mean $\mu_{E+1}$ is sufficiently far from the training means:*

$$\forall e \in \mathcal{E}, \min_{y \in \{\pm 1\}} \|\mu_{E+1} - y \cdot \mu_e\| \geq (\sqrt{\epsilon} + \delta)\sigma_e \sqrt{d_e} \tag{C.55}$$

*for some $\delta > 0$. Define the constants:*

$$k = \min_{e \in \mathcal{E}} \frac{\sigma^2_e}{\sigma^2_{E+1}} \tag{C.56}$$

$$q = \frac{2E}{\sqrt{k\pi}\delta} \exp\{-k\delta^2\}. \tag{C.57}$$

*Then the following holds:*

3. *$\Phi_\epsilon, \widehat{\beta}$ is equivalent to the ERM-optimal classifier on at least a $1 - q$ fraction of the test distribution.*
4. *Under Assumption Assumption 4.6.2, suppose it holds that*

$$\mu_{E+1} = -\sum_{e \in \mathcal{E}} \alpha_e \mu_e \tag{C.58}$$

*for some set of coefficients $\{\alpha_e\}_{e \in \mathcal{E}}$. Then for any $c \in \mathbb{R}$, so long as*

$$\sum_{e \in \mathcal{E}} \alpha_e \frac{\|\mu_e\|^2}{\sigma^2_e} \geq \frac{\|\mu_c\|^2/\sigma^2_c + |\beta_0|/2 + c\sigma_{ERM}}{1 - \gamma}, \tag{C.59}$$

*the 0-1 risk of $\Phi_\epsilon, \widehat{\beta}$ is lower bounded by $F(2c) - q$.*

*Proof.* Define $r = \sqrt{\epsilon\sigma_e^2 d_e}$ and construct $\mathcal{B}_r \subset \mathbb{R}^{d_e}$ as

$$\mathcal{B}_r = \left[\bigcup_{e\in\mathcal{E}} B_r(\mu_e)\right] \cup \left[\bigcup_{e\in\mathcal{E}} B_r(-\mu_e)\right], \tag{C.60}$$

where $B_r(\alpha)$ is the ball of $\ell_2$-norm radius $r$ centered at $\alpha$. Further construct $\Phi_\epsilon$ using $\mathcal{B}_r$ as follows:

$$\Phi_\epsilon(x) = \begin{cases} \begin{bmatrix} z_c \\ 0 \end{bmatrix}, & z_e \in \mathcal{B}_r \\ \begin{bmatrix} z_c \\ z_e \end{bmatrix}, & z_e \in \mathcal{B}_r^c, \end{cases} \quad \text{and} \quad \widehat{\beta} = \begin{bmatrix} \beta_c \\ \widehat{\beta}_e \\ \beta_0 \end{bmatrix}. \tag{C.61}$$

Without loss of generality, fix an environment $e$.

1. By Lemma C.4.1, the squared gradient norm is upper bounded by

$$\|\nabla_{\widehat{\beta}} \mathcal{R}^e(\Phi_\epsilon, \widehat{\beta})\|^2 \le \left\| \mathbb{P}(z_e \in \mathcal{B}_r^c)\mathbb{E}[|z_e| \mid z_e \in \mathcal{B}_r^c] \right\|_2^2. \tag{C.62}$$

Define $B_e := B_r(\mu_e)$, and observe that $\mathcal{B}_r^c \subseteq B_e^c$. Since $|z_e|$ is non-negative,

$$\mathbb{P}(z_e \in \mathcal{B}_r^c)\mathbb{E}[|z_e| \mid z_e \in \mathcal{B}_r^c] \le \mathbb{P}(z_e \in B_e^c)\mathbb{E}[|z_e| \mid z_e \in B_e^c] \tag{C.63}$$

(this inequality is element-wise). Plugging this into Equation (C.62) yields

$$\|\nabla_{\widehat{\beta}} \mathcal{R}^e(\Phi_\epsilon, \widehat{\beta})\|^2 \le \left\| \mathbb{P}(z_e \in B_e^c)\mathbb{E}[|z_e| \mid z_e \in B_e^c] \right\|_2^2 \tag{C.64}$$

$$= [\mathbb{P}(z_e \in B_e^c)]^2 \left\| \mathbb{E}[|z_e| \mid z_e \in B_e^c] \right\|_2^2. \tag{C.65}$$

Define $p = \mathbb{P}(z_e \in \mathcal{B}_r^c) \le \mathbb{P}(z_e \in B_e^c)$. By Lemma C.6.3,

$$p \le p_{\epsilon,d_e} = e^{-d_e \min((\epsilon-1),(\epsilon-1)^2)/8}. \tag{C.66}$$

Combining Lemmas Lemma C.6.4 and Lemma C.6.5 gives

$$\left\| \mathbb{E}[|z_e| \mid z_e \in B_e^c] \right\|_2^2 \le 2d_e \left[\sigma \frac{\phi(r/\sqrt{d_e})}{F(-r/\sqrt{d})}\right]^2 + 2\|\mu_e\|^2 \tag{C.67}$$

$$\le d_e\sigma_e^2 \exp\left\{2\epsilon(\sigma_e^2 - 1/2)\right\} \left[\epsilon\sigma_e^2 + 1\right] + 2\|\mu_e\|^2. \tag{C.68}$$

Putting these two bounds together, we have

$$\|\nabla_{\widehat{\beta}}\mathcal{R}^e(\Phi_\epsilon,\widehat{\beta})\|^2 \in \mathcal{O}\left(p^2_{\epsilon,d_e}\left[\epsilon d_e\sigma^4_{\max}\exp\{2\epsilon\sigma^2_{\max}\}+\|\mu_e\|^2\right]\right), \tag{C.69}$$

and averaging this value across environments gives the result.

2. $\Phi_\epsilon, \widehat{\beta}$ is equal to the invariant classifier on $\mathcal{B}_r$ and the ERM solution on $\mathcal{B}_r^c$. The claim then follows from Lemma C.6.3.
3. This follows directly from Lemma C.4.2.
4. With Equation (C.58), we have that

$$\begin{aligned}\beta_{e;\text{ERM}}^\top\mu_{E+1} &= -\sum_{e\in\mathcal{E}}\alpha_e\beta_{e;\text{ERM}}^\top\mu_e && \text{(C.70)}\\ &\le -2(1-\gamma)\sum_{e\in\mathcal{E}}\alpha_e\frac{\|\mu_e\|^2}{\sigma_e^2} && \text{(C.71)}\\ &\le -2(1-\gamma)\frac{\|\mu_c\|^2/\sigma_c^2+|\beta_0|/2+c\sigma_{\text{ERM}}}{1-\gamma} && \text{(C.72)}\\ &= -(2\|\mu_c\|^2/\sigma_c^2+|\beta_0|+2c\sigma_{\text{ERM}}). && \text{(C.73)}\end{aligned}$$

where we have applied Assumption 1 in the first inequality and Equation (C.59) in the second. Consider the full set of features $\Phi_\epsilon(x)=f^{-1}(x)$, and without loss of generality assume $y=1$. The label-conditional distribution of the resulting logit is

$$\beta_c^\top z_c+\beta_{e;\text{ERM}}^\top z_e+\beta_0\sim\mathcal{N}\left(\beta_c^\top\mu_c+\beta_{e;\text{ERM}}^\top\mu_{E+1}+\beta_0,\sigma^2_{\text{ERM}}\right). \tag{C.74}$$

Therefore, the 0-1 risk is equal to the probability that this logit is negative. This is precisely

$$\begin{aligned}&F\left(-\frac{\beta_c^\top\mu_c+\beta_{e;\text{ERM}}^\top\mu_{E+1}+\beta_0}{\sigma_{\text{ERM}}}\right) && \text{(C.75)}\\ &\ge F\left(\frac{(2\|\mu_c\|^2/\sigma_c^2+|\beta_0|+2c\sigma_{\text{ERM}})-2\|\mu_c\|^2/\sigma_c^2-|\beta_0|}{\sigma_{\text{ERM}}}\right) && \text{(C.76)}\\ &= F\left(\frac{2c\sigma_{\text{ERM}}}{\sigma_{\text{ERM}}}\right) && \text{(C.77)}\\ &= F(2c). && \text{(C.78)}\end{aligned}$$

Observe that by the previous part, $\Phi_\epsilon\neq f^{-1}$ on at most a q fraction of observations, so the risk of our classifier $\Phi_\epsilon,\widehat{\beta}$ can differ from that of $f^{-1},\widehat{\beta}$

by at most $q$. Therefore our classifier's risk is lower bounded by $F(2c) - q$. In particular, choosing $c = 1$ recovers the statement in the main body.

$\square$

### C.4.2 Discussion of Conditions and Assumption

To see just how often we can expect the conditions for Theorem C.4.3 to hold, we can do a Fermi approximation based on the expectations of each of the terms. A reasonable prior for the environmental means is a multivariate Gaussian $\mathcal{N}(m, \Sigma)$. We might expect them to be very concentrated (with $\mathrm{Tr}(\Sigma)$ small), or perhaps to have a strong bias (with $\|m\|^2 \gg \mathrm{Tr}(\Sigma)$). For simplicity we treat the variances $\sigma_c^2, \sigma_e^2$ as constants. Then the expected separation between any two means from this distribution is

$$\mathbb{E}[\|\mu_1 - \mu_2\|] = \mathbb{E}_{x\sim\mathcal{N}(0,2\Sigma)}[\|x\|] \approx \sqrt{2\,\mathrm{Tr}(\Sigma)}. \tag{C.79}$$

In high dimensions this value will tightly concentrate around the mean, which is in $\mathcal{O}(\sqrt{d_e})$. On the other hand, even a slight deviation from this separation, to $\Omega(\sqrt{d_e \log E})$, means $\delta \in \Omega(\sqrt{\log E})$, which implies $q \in \mathcal{O}(1/E)$; this is plenty small to ensure worse-than-random error on the test distribution.

Now we turn our attention to the second condition (Equation (C.59)). The expected squared norm of each mean is $d_e$, and in high dimensions we expect them to be reasonably orthogonal (as a rough approximation; this is technically not true with a non-centered Gaussian). Then so long as $\sum_i \alpha_i \in \Omega(1)$, the left-hand side of Equation (C.59) is approximately $d_e$. On the other hand, treating $\gamma$ as a constant, the right-hand side is close to $d_c + \sqrt{d_c + d_e} \in \mathcal{O}(d_c + \sqrt{d_e})$. Thus, Equation (C.59) is quite likely to hold for any mean $\mu_{E+1}$ with the same scale as the training environments but with reversed correlations—again, this is *exactly the situation* where IRM hopes to outperform ERM, and we have shown that it does not.

We can also do a quick analysis of Assumption Assumption 4.6.2 under this prior: the ERM-optimal non-invariant coefficient will be approximately $2m/\sigma_e^2$ with high probability, meaning $\widehat{\beta}^\top \mu \approx 2\|m\|^2/\sigma_e^2$ for every environment. Thus, this vector will be $\gamma$-close to optimal with $\gamma \approx 1$ for every environment with high probability.

## C.5 Extensions to Alternative Objectives

### C.5.1 Extensions for the Linear Case

Observe that the constraint of Equation (4.4) is strictly stronger than that of Bellot and van der Schaar [2020]; when the former is satisfied, the penalty term of the latter is necessarily 0. It is thus trivial to extend all results in the Section 4.5 to this objective. As another example, consider the risk-variance-penalized objective of Krueger et al. [2020]:

$$\min_{\Phi,\widehat{\beta}} \quad \frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}} \mathcal{R}^e(\Phi,\widehat{\beta}) + \lambda \mathrm{Var}_{e\in\mathcal{E}}\left(\mathcal{R}^e(\Phi,\widehat{\beta})\right), \tag{C.80}$$

It is simple to extend Theorem 4.5.1 under an additional assumption:

**Corollary C.5.1** (Extension to Theorem 4.5.1)**.** *Assume $f$ is linear. Suppose we observe $E \leq d_e$ environments with linearly independent means and identical variance $\sigma_e^2$. Consider minimizing empirical risk subject to a penalty on the risk variance (Equation (C.80)). Then there exists a $\Phi, \widehat{\beta}$ dependent on the non-invariant features which achieves a lower objective value than the invariant classifier for* any *choice of regularization parameter $\lambda \in [0,\infty]$.*

*Proof.* Consider the featurizer $\Phi$ constructed in Lemma C.3.2. If the environmental variance is constant, then the label-conditional distribution of the environmental features,

$$z_e \mid y \sim \mathcal{N}(y \cdot \tilde{\mu}\sigma_e^2, \sigma_e^2), \tag{C.81}$$

is also invariant. This implies that the optimal $\widehat{\beta}$ also has constant risk across the environments, meaning the penalty term is 0, and as a result the objective does not depend on the choice of $\lambda$. As in Theorem 4.5.1, invoking Lemma C.6.1 implies that the overall risk is lower than that of the invariant classifier. □

As mentioned in Section 4.5, this additional requirement of constant variance is due to the assumptions underlying the design of the objective—REx expects invariance of the conditional distribution $p(y \mid \Phi(x))$, in contrast with the strictly weaker invariance of $\mathbb{E}[y \mid \Phi(x)]$ assumed by IRM. This might seem to imply that REx is a more robust objective, but this does not convey the entire picture. The conditions for the above corollary are just one possible failure case for REx; by extending Theorem 4.5.3 to this objective, we see that REx is just as prone to bad solutions:

**Corollary C.5.2** (Extension to Theorem 4.5.3)**.** *Suppose we observe $E \leq d_e$ environments, such that all environmental means are linearly independent. Then there exists a $\Phi, \widehat{\beta}$ which uses* only environmental features *and, under any choice of $\lambda \in [0, \infty]$, achieves a lower objective value than the invariant classifier under 0-1 loss on every environment $e$ such that $\tilde{\mu} > \sigma_c^{-1}\|\mu_c\| + \frac{|\beta_0|}{2\sigma_c^{-1}\|\mu_c\|}$.*

*Proof.* We follow the proof of Theorem 4.5.3, except when solving for $p$ as in Lemma C.3.1 we instead find the unit-norm vector such that

$$p^\top \mu_e = \sigma_e \tilde{\mu} \quad \forall e \in \mathcal{E}. \tag{C.82}$$

Observe that by setting $\Phi(x) = [p^\top z_e]$ and $\widehat{\beta} = [1]$, the 0-1 risk in a given environment is

$$\eta F(-\tilde{\mu}\sigma_e/\sigma_e) + (1-\eta)F(-\tilde{\mu}\sigma_e/\sigma_e) = F(-\tilde{\mu}), \tag{C.83}$$

which is independent of the environment. Further, by carrying through the same proof as in Theorem 4.5.3, we get that this non-invariant classifier has lower 0-1 risk so long as

$$\alpha + \frac{|\beta_0|}{2\alpha} \leq \tilde{\mu}, \tag{C.84}$$

where $\alpha = \sigma_c^{-1}\|\mu_c\|$ □

Though $\tilde{\mu}$ here is not exactly the same value because of the slightly different solution (Equation (C.82)), it depends upon the geometry of the training environments in the same way—it is the same as taking the square root of each of the variances. We can therefore expect this condition to hold in approximately the same situations, which we empirically verify by replicating Figure C.2 with the modified equation below.

### C.5.2 Extensions for the Non-Linear Case

The failure of these objectives in the non-linear regime is even more straightforward, as we can keep unchanged the constructed classifier from Theorem 4.6.3. Observe that parts 2-4 of the theorem do not involve the objective itself, and therefore do not require modification.

To see that part 1 still holds, note that since the constructed classifier matches the invariant classifier on $1-p$ of the observations, its risk across environments can only vary on the remaining $p$ fraction: as the 0-1 risk on this fraction is bounded between 0 and $p$, it is immediate that the variance of the environmental risks is upper bounded by $\frac{p^2}{4}$, which is in $\mathcal{O}(p^2)$ as before. Applying this argument to the other objectives yields similar results.

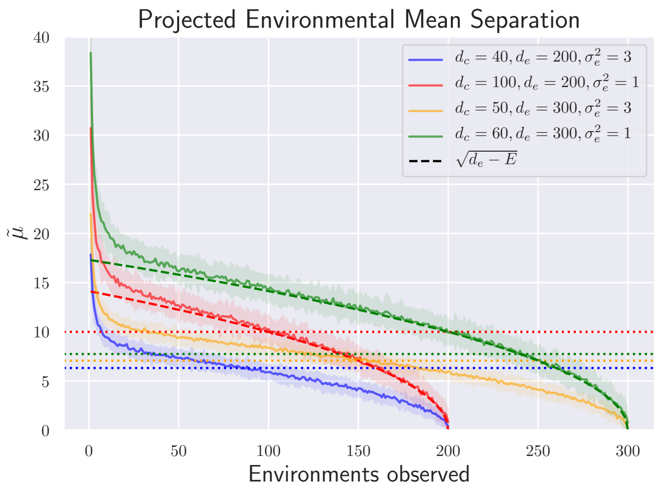


Figure C.3: Simulations to evaluate $\tilde{\mu}$ for varying ratios of $\frac{d_e}{d_c}$. When $\sigma_e^2 = 1$, the value closely tracks $\sqrt{d_e - E}$. Due to the similarity of Equation (C.82) to Equation (C.10), it makes sense that the results are very similar to those presented in Figure C.2.

## C.6 Technical Lemmas

**Lemma C.6.1.** *Consider solving the standard logistic regression problem*

$$z \sim p(z) \in \mathbb{R}^k, \tag{C.85}$$

$$y = \begin{cases} +1 & \textit{w.p. } \sigma(\beta^\top z), \\ -1 & \textit{w.p. } \sigma(-\beta^\top z). \end{cases} \tag{C.86}$$

*Assume that none of the latent dimensions are degenerate—$\forall S \subseteq [k]$, $\mathbb{P}(\beta_S^\top z_S \neq 0) > 0$, and no feature can be written as a linear combination of the other features. Then for any distribution $p(z)$, any classifier $f(z) = \sigma(\beta_S^\top z_S)$ that uses a strict subset of the features $S \subsetneq [k]$ has strictly higher risk with logistic loss than the Bayes classifier $f^*(z) = \sigma(\beta^\top z)$. This additionally holds for 0-1 loss if $\beta_{-S}^\top z_{-S}$ has greater magnitude and opposite sign of $\beta_S^\top z_S$ with non-zero probability.*

*Proof.* The Bayes classifier suffers the minimal expected loss for each observation z. Therefore, another classifier has positive excess risk if and only if it disagrees with the Bayes classifier on a set of non-zero measure. Consider the set of values $z_{-S}$ such that $\beta_{-S}^\top z_{-S} \neq 0$. Then on this set we have

$$f^*(\beta^\top z) = \sigma(\beta_S^\top z_S + \beta_{-S}^\top z_{-S}) \neq \sigma(\beta_S^\top z_S) = f(z). \tag{C.87}$$

Since these values occur with positive probability, $f$ has strictly higher logistic risk than $f^*$. By the same argument, there exists a set of positive measure under which

$$f^*(\beta^\top z) = \operatorname{sign}(\beta_S^\top z_S + \beta_{-S}^\top z_{-S}) \neq \operatorname{sign}(\beta_S^\top z_S) = f(z), \tag{C.88}$$

and so $f$ also has strictly higher 0-1 risk. $\square$

**Lemma C.6.2.** *For any feature vector which is a linear function of the invariant and environmental features $\tilde{z} = Az_c + Bz_e$, any optimal corresponding coefficient for an environment $e$ is of the form*

$$2(AA^\top \sigma_c^2 + BB^\top \sigma_e^2)^+ (A\mu_c + B\mu_e), \tag{C.89}$$

*where $G^+$ is a generalized inverse of $G$.*

*Proof.* We begin by evaluating a closed form for $p^e(y \mid \tilde{z})$. We have:

$$\begin{aligned} & p^e(y \mid Az_c + Bz_e = \tilde{z}) && \text{(C.90)} \\ =& \frac{p(Az_c + Bz_e = \tilde{z} \mid y)p(y)}{p^e(Az_c + Bz_e = \tilde{z})} && \text{(C.91)} \\ =& \frac{p^e(Az_c + Bz_e = \tilde{z} \mid y)}{p^e(Az_c + Bz_e = \tilde{z} \mid y) + p^e(Az_c + Bz_e = \tilde{z} \mid -y)} && \text{(C.92)} \\ =& \frac{1}{1 + \frac{p^e(Az_c + Bz_e = \tilde{z} \mid -y)}{p^e(Az_c + Bz_e = \tilde{z} \mid y)}}. && \text{(C.93)} \end{aligned}$$

Now we need a closed form expression for $p(Az_c + Bz_e = \tilde{z} \mid y)$. Noting that $z_c \perp\!\!\!\perp z_e \mid y$, this is a convolution of the two independent Gaussian densities, which is the density of their sum. In other words,

$$Az_c + Bz_e \mid y \sim \mathcal{N}(y \cdot \overbrace{(A\mu_c + B\mu_e)}^{\bar{\mu}}, \overbrace{AA^\top \sigma_c^2 + BB^\top \sigma_e^2}^{\bar{\Sigma}}). \tag{C.94}$$

Thus,

$$p^e(Az_c + Bz_e = \tilde{z} \mid y) = \frac{1}{\big(2\pi|\bar{\Sigma}|\big)^{k/2}} \exp\left\{-\frac{1}{2}(\tilde{z} - y \cdot \bar{\mu})^\top \bar{\Sigma}^+ (\tilde{z} - y \cdot \bar{\mu})\right\}. \tag{C.95}$$

Canceling common terms, we get

$$p^e(y = 1 \mid Az_c + Bz_e = \tilde{z}) = \frac{1}{1 + \frac{p^e(Az_c+Bz_e=\tilde{z}|-y)}{p^e(Az_c+Bz_e=\tilde{z}|y)}} \tag{C.96}$$

$$= \frac{1}{1 + \exp\left\{-y \cdot 2\tilde{z}^\top \bar{\Sigma}^+ \bar{\mu}\right\}} \tag{C.97}$$

$$= \sigma\left(y \cdot 2\tilde{z}^\top \bar{\Sigma}^+ \bar{\mu}\right). \tag{C.98}$$

Therefore, given a feature vector $\tilde{z}$, the optimal coefficient vector is $2\bar{\Sigma}^+\bar{\mu}$. □

**Lemma C.6.3.** *For any environment e with parameters $\mu_e, \sigma_e^2$ and any $\epsilon > 1$, define*

$$B := B_{\sqrt{\epsilon\sigma_e^2 d_e}}(\mu_e), \tag{C.99}$$

*where $B_r(\alpha)$ is the ball of $\ell_2$-norm radius $r$ centered at $\alpha$. Then for an observation drawn from $p^e$, we have*

$$\mathbb{P}_{z_e \sim p^e}(z_e \in B^c) \leq \exp\left\{-\frac{d_e \min((\epsilon-1),(\epsilon-1)^2)}{8}\right\}. \tag{C.100}$$

*Proof.* Without loss of generality, suppose $y = 1$. We have

$$\mathbb{P}(z_e \in B) \geq \mathbb{P}_{z_e \sim \mathcal{N}(\mu_e, \sigma_e^2 I)}\left(\|z_e - \mu_e\| \leq \sqrt{\epsilon\sigma_e^2 d_e}\right) \tag{C.101}$$

$$= \mathbb{P}_{z_e \sim \mathcal{N}(0, \sigma_e^2 I)}\left(\|z_e\| \leq \sqrt{\epsilon\sigma_e^2 d_e}\right) \tag{C.102}$$

$$= \mathbb{P}_{z_e \sim \mathcal{N}(0, I)}\left(\|z_e\|^2 \leq \epsilon d_e\right). \tag{C.103}$$

Each term in the squared norm of $z_e$ is a random variable with distribution $\chi_1^2$, which means their sum has mean $d_e$ and is sub-exponential with parameters $(2\sqrt{d_e}, 4)$. By standard sub-exponential concentration bounds we have

$$\mathbb{P}_{z_e \sim \mathcal{N}(0, I)}\left(\|z_e\|^2 \geq \epsilon d_e\right) \leq \exp\left\{-\frac{d_e \min((\epsilon-1),(\epsilon-1)^2)}{8}\right\}, \tag{C.104}$$

which immediately implies the claim. □

**Lemma C.6.4.** *Let $z \sim \mathcal{N}(\mu, \sigma^2 I_d)$ be a multivariate isotropic Gaussian in $d$ dimensions, and for some $r > 0$ define $B$ as the ball of $\ell_2$ radius $r$ centered at $\mu$. Then we have*

$$\left\| \mathbb{E}[|z| \mid z \in B^c] \right\|_2^2 \leq 2d \left[ \sigma \frac{\phi(r/\sqrt{d})}{F(-r/\sqrt{d})} \right]^2 + 2\|\mu\|^2, \tag{C.105}$$

*where $\phi, F$ are the standard Gaussian PDF and CDF.*

*Proof.* Observe that

$$\begin{aligned} \mathop{\mathbb{E}}_{z\sim\mathcal{N}(\mu,\sigma^2 I_d)} [|z| \mid z \in B^c] &= \mathop{\mathbb{E}}_{z\sim\mathcal{N}(\mu,\sigma^2 I_d)} [|z| \mid \|z-\mu\| > r] &\text{(C.106)}\\ &= \mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,\sigma^2 I_d)} [|z+\mu| \mid \|z\| > r] &\text{(C.107)}\\ &\leq \mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,\sigma^2 I_d)} [|z| \mid \|z\| > r] + |\mu|. &\text{(C.108)} \end{aligned}$$

Now, consider the expectation for an individual dimension, and note that $|z_i| > \frac{r}{\sqrt{d}}\ \forall i \implies \|z\| > r$. So because the dimensions are independent, conditioning on this event can only increase the expectation:

$$\begin{aligned} \mathop{\mathbb{E}}_{z\sim\mathcal{N}(0,\sigma^2 I_d)} [|z_i| \mid \|z\| > r] &\leq \mathop{\mathbb{E}}_{z_i\sim\mathcal{N}(0,\sigma^2)} \left[ |z_i| \mid |z_i| > \frac{r}{\sqrt{d}} \right] &\text{(C.109)}\\ &= \mathop{\mathbb{E}}_{z_i\sim\mathcal{N}(0,\sigma^2)} \left[ z_i \mid z_i > \frac{r}{\sqrt{d}} \right], &\text{(C.110)} \end{aligned}$$

where the equality is because the distribution is symmetric about 0. This last term is known as the *conditional tail expectation* of a Gaussian and is available in closed form:

$$\mathop{\mathbb{E}}_{z_i\sim\mathcal{N}(0,\sigma^2)} \left[ z_i \mid z_i > \frac{r}{\sqrt{d}} \right] = \sigma \frac{\phi(F^{-1}(\alpha))}{1-\alpha}, \tag{C.111}$$

where $\alpha = F(r/\sqrt{d})$. Combining the above results, squaring with $(a+b)^2 \leq 2(a^2+b^2)$, and summing over dimensions, we get

$$\begin{aligned} \left\| \mathbb{E}[|z| \mid z \in B^c] \right\|_2^2 &\leq 2 \sum_{i=1}^{d} \mathop{\mathbb{E}}_{z_i\sim\mathcal{N}(0,\sigma^2)} \left[ z_i \mid z_i > \frac{r}{\sqrt{d}} \right]^2 + 2\|\mu\|^2 &\text{(C.112)}\\ &= 2d \left[ \sigma \frac{\phi(r/\sqrt{d})}{F(-r/\sqrt{d})} \right]^2 + 2\|\mu\|^2, &\text{(C.113)} \end{aligned}$$

as desired. □

**Lemma C.6.5.** *For $\sigma, \epsilon > 0$, define $r = \sqrt{\epsilon}\sigma$. Then*

$$\left[\frac{\phi(r)}{F(-r)}\right]^2 \leq \frac{1}{2} \exp\left\{2\epsilon(\sigma^2 - 1/2)\right\} \left[\epsilon\sigma^2 + 1\right]. \tag{C.114}$$

*Proof.* We have

$$\phi(r) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{\epsilon}{2}\right\} \tag{C.115}$$

and

$$F(-r) \geq \frac{2\exp\{-\epsilon\sigma^2\}}{\sqrt{\pi}(\sqrt{\epsilon}\sigma + \sqrt{\epsilon\sigma^2 + 2})} \tag{C.116}$$

(see Kschischang [2017]). Dividing them gives

$$\frac{\phi(r)}{F(-r)} \leq \frac{1}{2\sqrt{2}} \exp\{\epsilon(\sigma^2 - 1/2)\} \left[\sqrt{\epsilon}\sigma + \sqrt{\epsilon\sigma^2 + 2}\right]. \tag{C.117}$$

Squaring and using $(a+b)^2 \leq 2(a^2 + b^2)$ yields the result. □

# Appendix D

# Appendix for Chapter 5

## D.1 Proof of Lemma 5.5.1

We use the following lemma to discretize the space of orthonormal matrices $\mathcal{Q} = \{Q : QQ^\top = I_k, Q \in \mathbb{R}^{k \times d_s}\}$. For any $Q, Q' \in \mathcal{Q}$, we define the metric $\rho(Q, Q') = \|Q^\top Q - Q'^\top Q'\|_F$. We recall the following lemma about the existence of a cover of $\mathcal{Q}$ with respect to the metric $\rho$:

**Lemma D.1.1** (Proposition 8 of Pajor [1998]). *For $1 \leq k \leq d_s/2$, there exists absolute constant $c_3$ and covering $\tilde{\mathcal{Q}} \subset \mathcal{Q}$ such that for all $\epsilon > 0$, $|\tilde{\mathcal{Q}}| \leq (c_3\sqrt{k}/\epsilon)^{k(d_s-k)}$, and $\forall Q^* \in \mathcal{Q}$, $\exists Q \in \tilde{\mathcal{Q}}$ such that $\rho(Q, Q^*) \leq \epsilon$.*

For any odd integer $e < E$, define $\Delta_2^e = \Sigma_2^e - \Sigma_2^{e+1} = (\overline{\Sigma_2^e} - \overline{\Sigma_2^{e+1}}) + (G_e G_e^\top - G_{e+1} G_{e+1}^\top)$.

For any $Q \in \mathcal{Q}$, let $q_i$ be the $i$-th row of $Q$, for $i \in [k]$. Let $Z_{ije} = (q_i^\top \Delta_2^e q_j)^2$. Define $A_{ije} = q_i^\top (\overline{\Sigma_2^e} - \overline{\Sigma_2^{e+1}}) q_j$, and $A = \sum_{\text{odd } e<E, i,j\in[k], i\neq j} A_{ije}^2$. The main lemma below shows that the sum of $Z_{ije}$'s are bounded away from 0.

**Lemma D.1.2.** *There exists constants $c_1, c_2, b_1, b_2 > 0$ such that for any integer $2 \leq k \leq d_s/2$, for all $E$ satisfying*

$$b_1 \frac{d_s - k}{k-1} \max\left\{1, \log\left(\frac{D}{(k-1)d_s}\right), \log\left(\frac{d_s}{k-1}\right)\right\} < E < b_2 d_s, \quad \text{(D.1)}$$

*where $\max_e \|\overline{\Sigma_2^e}\|_2^2 \leq D$ for some constant $D$, with probability $1 - c_1 \exp(-d_s)$, for all $Q \in \mathcal{Q}$,*

$$\sum_{\text{odd } e<E, i,j\in[k], i\neq j} Z_{ije} > c_2(A + Ek(k-1)d_s). \quad \text{(D.2)}$$

*Proof.* For any odd $e < E$ and $i \in [k]$, by definition

$$\sum_{j \neq i} Z_{ije} = \sum_{j \neq i} (A_{ije} + q_i^\top G_e G_e^\top q_j - q_i^\top G_{e+1} G_{e+1}^\top q_j)^2 \tag{D.3}$$

Define $V_{i,e} = G_e q_i$ for $e \in [E], i \in [k]$. For fixed orthonormal $Q$, $V_{i,e} \sim \mathcal{N}(0, I_{d_s})$ and the ensemble $\{V_{i,e}\}_{i \in [k], e \in [E]}$'s is independent. Therefore

$$q_i^\top G_e G_e^\top q_j - q_i^\top G_{e+1} G_{e+1}^\top q_j = V_{i,e}^\top V_{j,e} - V_{i,e+1}^\top V_{j,e+1} \tag{D.4}$$

For further simplification, we define $W_{i,e} = [V_{i,e}; V_{i,e+1}] \in \mathbb{R}^{2d_s}$, and $I^* = [I_{d_s}, \mathbf{0}; \mathbf{0}, -I_{d_s}]$, so

$$V_{i,e}^\top V_{j,e} - V_{i,e+1}^\top V_{j,e+1} = W_{i,e}^\top I^* W_{j,e} \tag{D.5}$$

We use the following lemma to decouple the correlations between $W_{i,e}^\top I^* W_{j,e}$ and $W_{i,e}^\top I^* W_{j',e}$ for $j' \neq j, j' \neq i, i \neq j$:

**Lemma D.1.3** (Theorem 1 of de la Peña and Montgomery-Smith [1995]). *Suppose $\{X_i\}$ ($i \in [k]$) are independent random variables, $X_i$ and $Y_i$ have the same distribution. There exists some absolute constant $c_4$ such that*

$$\Pr\left[\left|\sum_{i,j \in [k], i \neq j} f(X_i, X_j)\right| \geq t\right] \leq c_4 \Pr\left[\left|\sum_{i,j \in [k], i \neq j} f(X_i, Y_j)\right| \geq t/c_4\right]. \tag{D.6}$$

We apply Lemma D.1.3 with $X_i = W_{i,e}$ and $f(X_i, X_j) = Z_{ije} - \mathbb{E}[Z_{ije}]$ to get

$$\Pr\left[\left|\sum_{i,j \in [k], i \neq j} Z_{ije} - \mathbb{E}[Z_{ije}]\right| \geq t\right] \leq c_4 \Pr\left[\left|\sum_{i,j \in [k], i \neq j} Z'_{ije} - \mathbb{E}[Z'_{ije}]\right| \geq t/c_4\right]. \tag{D.7}$$

where $Y_{i,e}$ and $X_{i,e}$ are identically distributed and

$$Z'_{ije} = A_{ije}^2 + 2A_{ije} X_{i,e}^\top I^* Y_{j,e} + (X_{i,e}^\top I^* Y_{j,e})^2. \tag{D.8}$$

Note that $\{Z'_{ije}\}$ and $\{Z''_{ije}\}$ are identically distributed, where

$$Z''_{ije} = A_{ije}^2 + 2A_{ije} X_{i,e}^\top Y_{j,e} + (X_{i,e}^\top Y_{j,e})^2. \tag{D.9}$$

Below we first consider the randomness in $\{Y_{i,e}\}$, and prove that with high probability $\{Y_{i,e}\}$ satisfies some good properties; we then show the concentration of $\sum_{i,j,e} Z''_{ije}$ conditioned on the event that $\{Y_{i,e}\}$ satisfies these properties.

First, for fixed $Q$, since $Y_{i,e} = [G_e v_i; G_{e+1} v_i] \sim \mathcal{N}(0, I_{2d_s})$, if we write $Y_e = [Y_{1,e}; \dots; Y_{k,e}] \in \mathbb{R}^{k \times 2d_s}$, it is a random matrix with iid standard normal entries. We show that the $\|Y_e\|_F^2 = \Theta(kd_s)$ with high probability. The following lemma is a standard concentration bound for chi-squared variable:

**Lemma D.1.4** (Corollary of Lemma 1 in Laurent and Massart [2000]). *Suppose $Z_i \sim \mathcal{N}(0,1)$ for $i \in [n]$. For any $t > 0$,*

$$\Pr\left[\sum_{i=1}^{n} Z_i^2 \geq n + 2\sqrt{nt} + 2t\right] \leq \exp(-t), \tag{D.10}$$

$$\Pr\left[\sum_{i=1}^{n} Z_i^2 \leq n - 2\sqrt{nt}\right] \leq \exp(-t). \tag{D.11}$$

Applying Lemma D.1.4 to $n = Ekd_s$ entries of $\{Y_e\}_{e=1}^{E}$ and setting $t = Ekd_s/16$ we get with probability $1 - 2\exp(-Ekd_s/16)$,

$$\frac{Ekd_s}{2} \leq \sum_e \|Y_e\|_F^2 \leq \frac{13Ekd_s}{8}. \tag{D.12}$$

Second, we show that with high probability over the randomness of $G_e$, $\|Y_e\|_2$ viewed as a function of $Q$ satisfies $\|Y_e\|_2 = O(\sqrt{d_s})$ for all orthonormal $Q$. We use the following lemma to upper bound $\|G_e\|_2$:

**Lemma D.1.5** (Corollary 5.35 of Vershynin [2012]). *Suppose $G \in \mathbb{R}^{D \times d}$ and $[G]_{ij} \sim \mathcal{N}(0,1)$ for all $i \in [D], j \in [d]$. For every $t \geq 0$, with probability $1 - 2\exp(-t^2/2)$,*

$$\|G\|_2 \leq \sqrt{D} + \sqrt{d} + t \tag{D.13}$$

Applying Lemma D.1.5 with $G = [G_e; G_{e+1}]$, $D = 2d_s$, $d = d_s$, $t = \sqrt{d_s}$, we get with probability $1 - 2\exp(-d_s/2)$, $\|G\|_2 \leq (2 + \sqrt{2})\sqrt{d_s}$, and therefore for all orthonormal $Q \in \mathbb{R}^{k \times d_s}$,

$$\|Y_e\|_2 = \|QG^\top\|_2 \leq \|Q\|_2 \|G\|_2 \leq (2 + \sqrt{2})\sqrt{d_s}. \tag{D.14}$$

For any odd $e < E$, $i \in [k]$, and fixed $Y_e$, we prove $P_{ei} = \sum_{j \neq i} Z''_{ije}$ concentrates. Once we fix $Y_e$, the $Er/2$ random variables $\{P_{ei}\}$ are independent, so the concentration of their sum is immediate. Let $Y_{-i,e}$ be $Y_e$ without the $i$-th row,

$$P_{ei} = \sum_{j \neq i} Z''_{ije} = \sum_{j \neq i} A_{ije}^2 + 2X_{i,e}^\top \left(\sum_{j \neq i} A_{ije} Y_{j,e}\right) + X_{i,e}^\top Y_{-i,e} Y_{-i,e}^\top X_{i,e} \tag{D.15}$$

Define $B_{i,e} = Y_{-i,e}Y_{-i,e}^\top$. Let $a_{i,e} \in \mathbb{R}^{k-1}$ be the column vector consisting of $A_{ije}$ for $j \neq i$.

Since $X_{i,e} \sim \mathcal{N}(0, I_{2d_s})$, $X_{i,e}^\top \left(\sum_{j\neq i} A_{ije}Y_{j,e}\right)$ is a Gaussian variable with mean 0 and variance $a_{i,e}^\top B_{i,e}a_{i,e} \leq \|a_{i,e}\|_2^2\|B_{i,e}\|_2$, so by Hoeffding's inequality, for all $t \geq 0$,

$$\Pr\left[2X_{i,e}^\top\left(\sum_{j\neq i} A_{ije}Y_{j,e}\right) > t \mid Y_e\right] \leq \exp\left(-\frac{t^2}{8\|a_{i,e}\|^2\|B_{i,e}\|_2}\right). \tag{D.16}$$

By Hanson-Wright Inequality (e.g. Theorem 1.1 of Rudelson et al. [2013]), there exists constant $c_5$ such that

$$\begin{aligned}\Pr&\left[\mathbb{E}[X_{i,e}^\top B_{i,e}X_{i,e}] - X_{i,e}^\top B_{i,e}X_{i,e} > t \mid Y_e\right] \\ &\leq \exp\left(-c_5\min\left\{\frac{t^2}{\|B_{i,e}\|_F^2}, \frac{t}{\|B_{i,e}\|_2}\right\}\right).\end{aligned} \tag{D.17}$$

Combining Equations (D.15) to (D.17), we get

$$\begin{aligned}\Pr\left[\mathbb{E}[P_{ei}] - P_{ei} > t \mid Y_e\right] &\leq \exp\left(-\frac{t^2}{32\|a_{i,e}\|_2^2\|B_{i,e}\|_2}\right) \\ &+\exp\left(-c_5\min\{\frac{t^2}{4\|B_{i,e}\|_F^2}, \frac{t}{2\|B_{i,e}\|_2}\}\right).\end{aligned} \tag{D.18}$$

Summing over all $e \in [E]$ and $i \in [k]$ we get

$$\Pr\left[\mathbb{E}\left[\sum_{e,i} P_{ei}\right] - \sum_{e,i} P_{ei} > t \mid Y_1, \ldots, Y_E\right] \leq \exp\left(-\frac{t^2}{32\sum_{e,i}\|a_{i,e}\|_2^2\|B_{i,e}\|_2}\right) \tag{D.19}$$

$$+\exp\left(-c_5\min\left\{\frac{t^2}{4\sum_{e,i}\|B_{i,e}\|_F^2}, \frac{t}{2\max_{e,i}\|B_{i,e}\|_2}\right\}\right). \tag{D.20}$$

Note that $\mathbb{E}[X_{i,e}^\top BX_{i,e}] = \mathbb{E}\sum_{j\neq i}(X_{i,e}^\top Y_{j,e})^2 = \|Y_{-i,e}\|_F^2$ so

$$E\left[\sum_{e,i} P_{ei}\right] = \sum_{e,i}\|a_{i,e}\|_2^2 + \sum_{e,i}\|Y_{-i,e}\|_F^2 = A + (k-1)\sum_e \|Y_e\|_F^2. \tag{D.21}$$

Since $\|B_{i,e}\|_2 \le \|Y_e\|_2^2$, $\|B_{i,e}\|_F^2 \le \|Y_{-i,e}\|_F^2\|Y_e\|_2^2$, taking $t = \frac{1}{2}E[\sum_{e,i} P_{ei}]$,

$$\Pr\left[\sum_{e,i} P_{ei} < \frac{1}{2}\left(A + (k-1)\sum_e \|Y_e\|_F^2\right) \mid Y_1, \dots, Y_E\right]$$
$$\le \exp\left(-\frac{(A + (k-1)\sum_e \|Y_e\|_F^2)^2}{128\sum_{e,i}\|a_{i,e}\|_2^2\|Y_e\|_2^2}\right)$$
$$+ \exp\left(-c_5 \min\left\{\frac{(k-1)^2(\sum_e \|Y_e\|_F^2)^2}{16(k-1)\sum_e \|Y_e\|_F^2\|Y_e\|_2^2}, \frac{(k-1)\sum_e \|Y_e\|_F^2}{2\max_e \|Y_e\|_2^2}\right\}\right). \quad \text{(D.22)}$$

Let $\mathcal{E}_1$ denote the event that for all odd $e < E$, $[G_e; G_{e+1}] \in \mathbb{R}^{2d_s \times d_s}$ denote the matrix with $G_e, G_{e+1} \in \mathbb{R}^{d_s \times d_s}$ in its first and last $d_s$ rows, respectively, we have

$$\|[G_e; G_{e+1}]\|_2 \le (2+\sqrt{2})\sqrt{d_s}. \quad \text{(D.23)}$$

Due to Equation (D.14) and the union bound, $\Pr[\mathcal{E}_1] \ge 1 - E\exp(-d_s/2)$. Conditioned on $\mathcal{E}_1$, for all $Q \in \mathcal{Q}$ and odd $e < E$,

$$\|Y_e\|_2 \le (2+\sqrt{2})\sqrt{d_s}. \quad \text{(D.24)}$$

Let $\mathcal{E}_2$ denote the event that for all cover elements $Q \in \tilde{Q}$,

$$\frac{Ekd_s}{2} \le \sum_e \|Y_e\|_F^2 \le \frac{13Ekd_s}{8}. \quad \text{(D.25)}$$

Due to Equation (D.12) and the union bound, $\Pr[\mathcal{E}_2] \ge 1 - 2|\tilde{Q}|\exp(-Ekd_s/16)$. Conditioned on $\mathcal{E}_1$ and $\mathcal{E}_2$, for fixed $Q \in \tilde{Q}$, there exists constants $c_6, c_7$ such that

$$\Pr\left[\sum_{e,i} P_{ei} < \frac{1}{2}A + \frac{1}{4}Ek(k-1)d_s\right] \le \exp\left(-c_6\frac{(A + Ek(k-1)d_s)^2}{Ad_s}\right) \quad \text{(D.26)}$$
$$+ \exp\left(-c_7 \min\left\{\frac{(k-1)^2E^2k^2d_s^2}{Ek(k-1)d_s^2}, \frac{Ek(k-1)d_s}{d_s}\right\}\right), \quad \text{(D.27)}$$

which implies there exists constants $c_8$ such that

$$\Pr\left[\sum_{e,i} P_{ei} < \frac{1}{2}A + \frac{1}{4}Ek(k-1)d_s\right]$$
$$\le \exp\left(-c_8 \min\left\{\frac{(A + Ek(k-1)d_s)^2}{Ad_s}, Ek(k-1)\right\}\right). \quad \text{(D.28)}$$

Note that we always have $\frac{(A+Ek(k-1)d_s)^2}{Ad_s} \geq Ek(k-1)$. To see this, for $A > Ek(k-1)d_s$, $\frac{(A+Ek(k-1)d_s)^2}{Ad_s} > \frac{A}{d_s} > Ek(k-1)$. For $A \leq Ek(k-1)d_s$, $\frac{(A+Ek(k-1)d_s)^2}{Ad_s} \geq \frac{(Ek(k-1)d_s)^2}{Ek(k-1)d_s^2} = Ek(k-1)$.

In other words, with probability $1-\delta$, where

$$\delta = E\exp(-d_s/2) + 2|\tilde{Q}|\exp\left(-Ekd_s/16\right) + |\tilde{Q}|\exp\left(-c_8Ek(k-1)\right), \tag{D.29}$$

all $Q \in \tilde{Q}$ satisfies $\sum_{e,i} P_{ei} \geq \frac{1}{4}(A+Ek(k-1)d_s)$. Combined with Lemma D.1.3, with probability $1-c_9\delta$, all $Q \in \tilde{Q}$ satisfies $\sum_{e,i,j} Z_{ije} < c_{10}(A+Ek(k-1)d_s)$ for some constants $c_9, c_{10}$.

For any $Q^* \in \mathcal{Q}$, let $Q$ be the element in the cover closest to it, so that $\rho(Q,Q^*) = \|Q^\top Q - {Q^*}^\top Q^*\|_F \leq \epsilon$. Let $q_i^*$ be the $i$-th row of $Q^*$, and $Z_{ije}^* = ({q_i^*}^\top \Delta_2^e q_j^*)$. Then

$$\sum_{eij} Z_{ije}^* = \sum_e \|Q^*\Delta_2^e {Q^*}^\top\|_F^2 \tag{D.30}$$

$$= \sum_e \|\Delta_2^e {Q^*}^\top Q^*\|_F^2 \tag{D.31}$$

$$\geq \frac{1}{2}\sum_e \|\Delta_2^e Q^\top Q\|_F^2 - \|\Delta_2^e\left(Q^\top Q - {Q^*}^\top Q^*\right)\|_F^2 \tag{D.32}$$

$$\geq \frac{1}{2}\sum_{eij} Z_{ije} - \|\Delta_2^e\|_2^2\rho(Q,Q^*)^2. \tag{D.33}$$

Since $\|\Delta_2^e\|_2^2 \leq 2\|\overline{\Sigma_2^e}\|_2^2 + 2\|G_eG_e^\top\|_2^2$, and conditioned on $\mathcal{E}_1$, $\|G_eG_e^\top\|_2^2 \leq c_{11}d_s^2$ for all $e$, if $\max_e \|\overline{\Sigma_2^e}\|_2^2 \leq D$ for some constant $D$, we have with probability $1-\delta$,

$$\sum_{eij} Z_{ije}^* \geq \frac{c_{10}}{2}(A+Ek(k-1)d_s) - 2E(D+c_{11}d_s^2)\epsilon^2. \tag{D.34}$$

We choose $\epsilon^2 < \frac{c_{10}k(k-1)d_s}{8(D+c_{11}d_s^2)}$ so that $2E(D+c_{11}d_s^2)\epsilon^2 < \frac{c_{10}}{4}Ek(k-1)d_s$.

With this choice of $\epsilon$, by Equation (D.34) we have

$$\sum_{eij} Z_{ije}^* \geq \frac{c_{10}}{4}(A+Ek(k-1)d_s). \tag{D.35}$$

By Lemma D.1.1, $\log(|\tilde{Q}|) \leq k(d_s-k)\log(c_3\sqrt{k}/\epsilon) \leq c_{12}k(d_s-k)\log\left(\frac{D}{(k-1)d_s} + \frac{d_s}{k-1}\right)$.

Therefore there exists $b_1, b_2 > 0$ such that for $E$ satisfying

$$b_2 d_s > E > b_1 \frac{d_s - k}{k-1} \max\left\{1, \log\left(\frac{D}{(k-1)d_s}\right), \log\left(\frac{d_s}{k-1}\right)\right\}, \tag{D.36}$$

we have

$$\delta \leq \exp(-d_s/2 + \log(b_2 d_s)) \tag{D.37}$$

$$+ 2\exp\left(c_{12}k(d_s - k)\log\left(\frac{D}{(k-1)d_s} + \frac{d_s}{k-1}\right) - Ekd_s/16\right) \tag{D.38}$$

$$+ \exp\left(c_{12}k(d_s - k)\log\left(\frac{D}{(k-1)d_s} + \frac{d_s}{k-1}\right) - c_8 Ek(k-1)\right) \tag{D.39}$$

$$\leq c_1 \exp(-d_s) \tag{D.40}$$

for some constant $c_1$. Therefore with probability $1 - c_1 \exp(-d_s)$, for all $Q^* \in \mathcal{Q}$, and $c_2 = c_{10}/4$,

$$\sum_{eij} Z^*_{ije} \geq c_2(A + Ek(k-1)d_s). \tag{D.41}$$

□

**Corollary D.1.6** (Corollary of Lemma D.1.2)**.** *Suppose* $2 \leq k \leq r/2 \leq d_s/2$. *Let* $\mathcal{P} = \{P \in \mathbb{R}^{r \times d_s} : PP^\top = I_r\}$, $\mathcal{Q} = \{Q \in \mathbb{R}^{k \times r} : QQ^\top = I_k\}$. *For fixed* $P \in \mathcal{P}$, *there exists constants* $c_1, c_2, b_1, b_2 > 0$ *such that for all* $E$ *satisfying*

$$b_1 \frac{r-k}{k-1} \max\left\{1, \log\left(\frac{D}{(k-1)d_s}\right), \log\left(\frac{d_s}{k-1}\right)\right\} < E < b_2 d_s, \tag{D.42}$$

*where* $\max_e \|\overline{\Sigma_2^e}\|_2^2 \leq D$ *for some constant* $D$*, with probability* $1 - c_1 \exp(-d_s)$*, for all* $Q \in \mathcal{Q}$*,*

$$\sum_{\text{odd } e<E} \|QP\Delta_2^e P^\top Q^\top\|_F^2 > c_2 Ek(k-1)d_s. \tag{D.43}$$

*Proof.* The proof mostly follows that of Lemma D.1.2, with a few modifications below. We discretize over $\mathcal{Q}$ and get a $\epsilon$-covering $\tilde{\mathcal{Q}}$ of size $(c_3\sqrt{k}/\epsilon)^{r(r-k)}$.

For any $Q \in \mathcal{Q}$, let $v_i$ be the $i$-th row of $QP$ and define $Z_{ije}, A_{ije}$ accordingly. For any $Q^* \in \mathcal{Q}$, let $Q$ be its cover element, so $\rho(Q, Q^*) = \|Q^\top Q - Q^{*\top} Q^*\|_F \leq \epsilon$.

Let $q_i^*$ be the $i$-th row of $Q^*P$, and $Z_{ije}^* = (q_i^{*\top}\Delta_2^e q_j^*)$. Then

$$\begin{aligned}
\sum_{eij} Z_{ije}^* &= \sum_e \|Q^* P \Delta_2^e P^\top Q^{*\top}\|_F^2 &\text{(D.44)}\\
&= \sum_e \|P\Delta_2^e P^\top Q^{*\top} Q^*\|_F^2 &\text{(D.45)}\\
&\geq \frac{1}{2}\sum_e \|P\Delta_2^e P^\top Q^\top Q\|_F^2 - \|P\Delta_2^e P^\top \left(Q^\top Q - Q^{*\top}Q^*\right)\|_F^2 &\text{(D.46)}\\
&\geq \frac{1}{2}\sum_{eij} Z_{ije} - \|P\Delta_2^e P^\top\|_2^2 \rho(Q,Q^*)^2 &\text{(D.47)}\\
&\geq \frac{1}{2}\sum_{eij} Z_{ije} - \|\Delta_2^e\|_2^2 \rho(Q,Q^*)^2 &\text{(D.48)}
\end{aligned}$$

Thus with the same choice of $\epsilon$ as Lemma D.1.2, $\log(|\tilde{Q}|) \leq k(r-k)\log(c_3\sqrt{k}/\epsilon) \leq c_{12}k(r-k)\log\left(\frac{D}{(k-1)d_s} + \frac{d_s}{k-1}\right)$. The rest of the argument is identical. □

**Lemma D.1.7.** *Let $\mathcal{P} = \{P \in \mathbb{R}^{2\times d_s} : PP^\top = I_2\}$. Suppose $\Sigma_2 = \overline{\Sigma_2^1} - \overline{\Sigma_2^2} + G_1G_1^\top - G_2G_2^\top$ and $\Sigma_2' = \overline{\Sigma_2^1} - \overline{\Sigma_2^3} + G_1G_1^\top - G_3G_3^\top$, where $G_e \in \mathbb{R}^{d_s\times d_s}$ and $[G_e]_{ij} \sim \mathcal{N}(0,1)$ for all $e \in [3]$, $i,j \in [d_s]$. For fixed $P \in \mathcal{P}$, with probability 1, no vector $q \in \mathbb{R}^2$ satisfies $\|q\|_2 = 1$ and*

$$q^\top \Sigma_2 q = 0, \quad q^\top \Sigma_2' q = 0. \tag{D.49}$$

*Proof.* For any fixed $G_1, G_2$, consider the system of quadratic equations over two variables,

$$\{q^\top \Sigma_2 q = 0, \|q\|_2 = 1\}. \tag{D.50}$$

With probability 1, it has at most 4 real solutions. Conditioned on $G_1, G_2$, consider the third quadratic equation where the randomness is in $G_3$.

$$\{q^\top \Sigma_2' q = 0\}. \tag{D.51}$$

With probability 1, any fixed solution from the first system does not satisfy this. □

The following lemma is trivial so proof is omitted:

**Lemma D.1.8.** *Suppose $p \in \mathbb{R}^{d_s}$ and $\|p\|_2 = 1$. Suppose $\Sigma_2 = \overline{\Sigma_2^1} - \overline{\Sigma_2^2} + G_1 G_1^\top - G_2 G_2^\top$, where $G_e \in \mathbb{R}^{d_s \times d_s}$ and $[G_e]_{ij} \sim \mathcal{N}(0,1)$ for $e \in [2]$, $i, j \in [d_s]$. With probability 1, no scalar $q \neq 0$ satisfies*

$$q^2 p^\top \Sigma_2 p = 0. \tag{D.52}$$

## D.2 Proof of Theorem 5.4.2

*Proof.* Denote the unit-norm classifier $\beta$. For any environment with mean $(\mu_1, \mu_2^i)$ and covariance $\Sigma^e \succeq \sigma_{\min} I$, the accuracy of $\beta$ can be written

$$\begin{aligned}
& \mathbb{E}[\mathbf{1}(\mathrm{sign}(\beta^\top x) = y] && \text{(D.53)}\\
= \; & p(y=1)p(\beta^\top x \geq 0 \mid y = 1) + p(y=-1)p(\beta^\top x < 0 \mid y = -1) && \text{(D.54)}\\
= \; & \frac{1}{2}\left[1 - \Phi\left(-\frac{\beta_1^\top \mu_1 + \beta_2^\top \mu_2^i}{\sqrt{\beta^\top \Sigma^e \beta}}\right)\right] + \frac{1}{2}\Phi\left(\frac{\beta_1^\top \mu_1 + \beta_2^\top \mu_2^i}{\sqrt{\beta^\top \Sigma^e \beta}}\right) && \text{(D.55)}\\
= \; & \Phi\left(\frac{\beta_1^\top \mu_1 + \beta_2^\top \mu_2^i}{\sqrt{\beta^\top \Sigma^e \beta}}\right), && \text{(D.56)}
\end{aligned}$$

where $\Phi$ is the standard normal CDF. Observe that $\Phi$ is monotone, and therefore a training accuracy of at least $\gamma$ on each environment implies that for each environment,

$$\begin{aligned}
\gamma &\leq \Phi\left(\frac{\beta_1^\top \mu_1 + \beta_2^\top \mu_2^i}{\sqrt{\beta^\top \Sigma \beta}}\right) && \text{(D.57)}\\
&\leq \Phi\left(\frac{\beta_1^\top \mu_1 + \beta_2^\top \mu_2^i}{\sqrt{\sigma_{\min}}}\right). && \text{(D.58)}
\end{aligned}$$

Applying the inverse CDF (which is also monotone) and rearranging, we have

$$\beta_2^\top \mu_2^i \geq \sqrt{\sigma_{\min}} \Phi^{-1}(\gamma) - \beta_1^\top \mu_1, \tag{D.59}$$

which implies

$$\beta_1^\top \mu_1 - \beta_2^\top \mu_2^i \leq 2\beta_1^\top \mu_1 - \sqrt{\sigma_{\min}} \Phi^{-1}(\gamma). \tag{D.60}$$

If $\gamma \geq \Phi\left(\frac{2\|\mu_1\|}{\sqrt{\sigma_{\min}}}\right) \geq \Phi\left(\frac{2\beta_1^\top \mu_1}{\sqrt{\sigma_{\min}}}\right)$ then we have $\beta_1^\top \mu_1 - \beta_2^\top \mu_2^i \leq 0$ for all environments and therefore the classifier has accuracy $\leq \frac{1}{2}$ on all test environments. $\square$

## D.3 Proof of Theorem 5.4.3

**Definition D.3.1.** For a positive definite matrix $A \in Mat_{d\times d}(\mathbb{R})$ and vector $b \in \mathbb{R}^d$, the associated ellipsoid $E_{A,b} \subseteq \mathbb{R}^d$ is given by

$$E_{A,b} = \{x \in \mathbb{R}^d : x^\top Ax - b^\top x = 0\}.$$

Observe that the origin is contained in any such ellipsoid $E_{A,b}$. Therefore, any collection of ellipsoids $E_{A_i,b_i}$ has the origin as a trivial point in its intersection. Our main result ensures the existence of another (non-trivial) intersection of any $d$ such ellipses whenever the vectors $b_i$ are linearly independent.

**Theorem D.3.2.** *If $b_1, \dots, b_d \in \mathbb{R}^d$ are linearly independent and $A_1, \dots, A_d$ are positive-definite matrices, then*

$$\left|\bigcap_{i=1}^{d} E_{A_i,b_i}\right| \geq 2. \tag{D.61}$$

To prove this result we use technical tools from differential topology. The most central tool, Proposition D.3.6, ensures that the total number of intersection points between two manifolds of complementary dimensions $k, d-k$ is even when certain generic tranversality conditions hold. Using these techniques, we show that $\left|\bigcap_{i=1}^{d} E_{A_i,b_i}\right| \geq 2$ for almost all matrices $A_1, \dots, A_d$, as long as $b_1, \dots, b_d$ are linearly independent. Then we use a continuity argument to extend the result to all positive definite matrices $A_1, \dots, A_d$.

Throughout we say a function is *smooth* to mean it is infinitely differentiable, i.e. $C^\infty$. All manifolds considered are smooth, i.e. they have a smooth structure. When $F(x, y)$ has two arguments we denote by $F_x$ the function $F_x(y) = F(x, y)$ of $y$ given by fixing $x$, and similarly define $F_y$. If $x \in X$ is a point in the smooth manifold $X$, we denote by $T_x(X)$ its *tangent space*, which is intuitively the vector space of all tangent vectors to $X$ at $x$. The derivative of a smooth map $f : X \to Y$ at $x \in X$ is a linear map $df_x : T_x(X) \to T_{f(x)}(Y)$.

**Definition D.3.3.** [Guillemin and Pollack, 2010, Chapter 1.5]

Let $X, Y, Z$ be smooth manifolds (without boundary) such that $Z \subseteq Y$. The smooth map $f : X \to Y$ is *tranverse* to $Z$ if for each $x \in X$ with $f(x) \in Z$, it holds that

$$\text{Image}(df_x) + T_{f(x)}(Z) = T_{f(x)}(Y).$$

If $X, Z \subseteq Y$ are both submanifolds of $Y$, we say they are transverse if the inclusion $\iota_X : X \hookrightarrow Y$ is transverse to $Z$. Equivalently, this means that for any $x \in X \cap Z$,

$$T_x(X) + T_x(Z) = T_x(Y).$$

Roughly speaking, smooth two manifolds $X, Z$ are transversal if all intersection points are "typical". For example, if $\dim(X) + \dim(Z) < \dim(Y)$, then $X, Z$ being transverse is equivalent to their intersection being empty. This corresponds to the intuition that their total dimension is too small for them to generically intersect. If $\dim(X) + \dim(Z) = \dim(Y)$, transversality rules out "unstable" intersections such as a line tangent to a circle.

**Proposition D.3.4.** *[Guillemin and Pollack, 2010, Chapter 1.5]*

*The intersection $W = X \cap Z$ of two transversal submanifolds $X, Z \subseteq Y$ is itself a submanifold of $Y$, and $\dim(W) = \dim(X) + \dim(Z) - \dim(Y)$.*

**Proposition D.3.5.** *[Guillemin and Pollack, 2010, Chapter 2.3]*

*Suppose that $F : X \times S \to Y$ is a smooth map of manifolds, and let $Z$ be a sub-manifold of $Y$. If $F$ is transversal to $Z$, then for almost every $s \in S$, the map $f_s = F(\cdot, s) : X \to Y$ is also transversal to $Z$.*

**Proposition D.3.6.** *[Guillemin and Pollack, 2010, Chapter 2.4, Exercise 5]*

*Suppose the smooth, compact manifolds $X, Y \subseteq \mathbb{R}^d$ are transversal, and that $\dim(X) + \dim(Y) = d$. Then $|X \cap Y|$ is finite and even.*

*Remark* D.3.7. Proposition D.3.6 follows from the methods of [Guillemin and Pollack, 2010, Chapter 2.4], which shows that the parity of $|X \cap Y|$ is invariant under homotopy as long as transversality is enforced. One simply argues that by a homotopy $X \to X', Y \to Y'$, we can arrange that $|X' \cap Y'| = 0$ by translating $X$ far away and invoking compactness.

**Lemma D.3.8.** *The tangent space $T_0 E_{A,b}$ is exactly the orthogonal complement $b^{\perp}$.*

*Proof.* Since $E_{A,b}$ is an ellipsoid, it is a smooth manifold of dimension $d-1$. If $\gamma : [0,1] \to E_{A,b}$ is a smooth curve with $\gamma(0) = 0$, then we claim $\langle b, \gamma'(t) \rangle = 0$. This suffices to prove the desired result since $\gamma'(t)$ can be any vector in $T_0 E_{A,b}$. Indeed, differentiating the equation for $E_{A,b}$ gives

$$0 = 2\frac{d}{dt}\langle 0, A\gamma(t) \rangle \tag{D.62}$$
$$= \frac{d}{dt}\langle \gamma(t), A\gamma(t) \rangle|_{t=0} \tag{D.63}$$
$$= \frac{d}{dt}\langle b, \gamma(t) \rangle|_{t=0} \tag{D.64}$$
$$= \langle b, \gamma'(t) \rangle|_{t=0}. \tag{D.65}$$

□

Set $\mathcal{A}^\circ$ to be the set of all $d \times d$ strictly positive-definite matrices with distinct eigenvalues. Note that $\mathcal{A}^\circ$ is open in the space of all positive definite matrices, and its complement has Lebesgue measure $0$. Denote by $\mathbb{S}^{d-1} \subseteq \mathbb{R}^d$ the unit sphere so that $(c_1, \dots, c_d) \in \mathbb{S}^{d-1}$ if and only if $\sum_{i=1}^d c_i^2 = 1$.

**Proposition D.3.9.** *[Serre, 2010, Theorem 5.3]*

*For any $A_0 \in \mathcal{A}^\circ$, there is an open neighborhood $U_{A_0} \subseteq \mathcal{A}^\circ$ of $A_0$ such that the eigenvalues $\lambda_1(A) > \cdots > \lambda_d(A)$ and associated orthonormal eigenvectors $v_1, \dots, v_d$ can be chosen to depend smoothly on the entries of $A \in U_{A_0}$.*

We remark that is it impossible to make a *globally* smooth choice of the eigenvectors and eigenvalues as above. This is because of problems caused by higher multiplicity eigenvalues, and also by the need to choose a sign for the eigenvectors.

**Lemma D.3.10.** *For $A \in \mathcal{A}^\circ$ and non-zero $b \in \mathbb{R}^d$, let $\lambda_1 > \cdots > \lambda_d$ be the eigenvalues of $A$, with associated orthonormal eigenvectors $v_1, \dots, v_d$. Then $x \in E_{A,b}$ if and only if $x = x_0 + x_1$ where $x_0 = \frac{A^{-1}b}{2}$ and*

$$x_1 = \frac{\sqrt{b^\top A^{-1} b}}{2} \sum_{i=1}^d \frac{c_i v_i}{\sqrt{\lambda_i}}$$

*for $(c_1, \dots, c_d) \in \mathbb{S}^{d-1}$.*

*Proof.* Writing $x = x_0 + x_1$, we derive

$$\begin{aligned} x_1^\top A x_1 + x_1^\top b + \frac{b^\top A^{-1} b}{4} &= x_1^\top A x_1 + 2x_1^\top A x_0 + x_0 A x_0 \\ &= x^\top A x \\ &= b^\top (x_1 + x_0) \qquad \text{(D.66)} \\ &= b^\top x_1 + \frac{b^\top A^{-1} b}{2}. \end{aligned}$$

Since we used the condition $x \in E_{A,b}$ only in reaching Equation (D.66), the initial and final expressions are equal if and only if $x \in E_{A,b}$. It follows that $x = x_0 + x_1 \in E_{A,b}$ if and only if

$$x_1^\top A x_1 = \frac{b^\top A^{-1} b}{4}.$$

This easily leads to the parametrization given and concludes the proof. □

**Lemma D.3.11.** *Let $M^k \subseteq \mathbb{R}^d$ be a compact manifold of dimension $k \geq 1$ passing through the origin, and such that $T_0(M^k) \subsetneq b^\perp$. Then for all but a measure-zero set of positive-definite matrices $A$, the ellipsoid $E_{A,b}$ is transversal to $M^k$.*

*Proof of Lemma D.3.11.* Fixing $A_0 \in \mathcal{A}^\circ$, Proposition D.3.9 ensures the existence of an open neighborhood $U_{A_0} \subseteq \mathcal{A}^\circ$ of $A_0$ on which the eigenvalues $\lambda_1(A) > \lambda_2(A) > \cdots > \lambda_d(A)$ and associated orthonormal eigenvectors $v_1(A), \ldots, v_d(A)$ are defined smoothly on all $A \in U_{A_0}$. Define $F : U_A \times \mathbb{S}^{d-1} \to \mathbb{R}^d$ by:

$$F(A, (c_1, \ldots, c_d)) = \frac{A^{-1}b}{2} + \frac{\sqrt{b^\top A^{-1} b}}{2} \sum_{i=1}^{d} \frac{c_i v_i(A)}{\sqrt{\lambda_i(A)}}.$$

Lemma D.3.10 implies that for each fixed $A$ we obtain a diffeomorphism $F_A : \mathbb{S}^{n-1} \to E_{A,b}$. Moreover, $F$ is smooth by construction. We claim that $F$ and $M^k$ are transversal. To check this, we must verify that for any $z = F(A, c) \in M^k$, it holds that

$$\text{Image}\left(dF \circ T_{F^{-1}(z)}(U_{A_0} \times \mathbb{S}^{N-1})\right) + T_z(M^k) = \mathbb{R}^d.$$

First, recall that fixing $A = A_0$, the map $F_{A_0} : \mathbb{S}^{n-1} \to E_{A_0,b}$ is a diffeomorphism. Therefore

$$\text{Image}\left(dF \circ T_{F^{-1}(z)}(U_{A_0} \times \mathbb{S}^{N-1})\right)$$

contains the tangent space $T_z(E_{A,b}) = b^\perp$ of $E_{A,b}$ at $z$. When $z = 0$ is the origin, the assumption $T_0(M^k) \subsetneq b^\perp$ implies

$$\dim\left(\text{Image}\left(dF \circ T_{F^{-1}(z)}(U_{A_0} \times \mathbb{S}^{N-1})\right) + T_z(M^k)\right) \geq \dim(b^\perp) + 1 = d$$

and the claim follows. Supposing for the remainder of the proof that $z \neq 0$ is not the zero vector, we claim that in fact

$$\text{Image}\left(dF \circ T_{F^{-1}(z)}(U_{A_0} \times \mathbb{S}^{N-1})\right) + T_z(M^k) = \mathbb{R}^d,$$

i.e. the tangent space of $M^k$ is unnecessary. Indeed fixing $c \in \mathbb{S}^{N-1}$, we may vary $A \in U_A$ along the path $\gamma_A(t) = \frac{A}{t}$ for $t \in (1 - \varepsilon, 1 + \varepsilon)$. It is not difficult to see directly that

$$F(tA, c) = tF(A, c).$$

Therefore differentiating $F$ along $\gamma$ gives

$$\frac{d}{dt} F(\gamma_A(t), (c_1, \ldots, c_d))|_{t=1} = F(A, c).$$

This means $z \in \text{Image}\left(dF \circ T_{F^{-1}(z)}(U_{A_0} \times \mathbb{S}^{N-1})\right) + T_z(M^k)$. Because $E_{A,b}$ is strictly convex and passes through the origin, it follows that the tangent hyperplane to $E_{A,b}$ at $z$ does not pass through the origin, hence $z \notin T_z(E_{A,b})$. We have establish that Image $\left(dF \circ T_{F^{-1}(z)}(U_{A_0} \times \mathbb{S}^{N-1})\right) + T_z(M^k)$ contains both $T_z(E_{A,b})$ and $z \notin T_z(E_{A,b})$. Since $\dim\left(T_z(E_{A,b})\right) = d-1$ it follows that Image $\left(dF \circ T_{F^{-1}(z)}(U_{A_0} \times \mathbb{S}^{N-1})\right) + T_z(M^k) = \mathbb{R}^d$ for $z \neq 0$ as claimed. This shows the desired transversality for almost all $A \in U_{A_0}$.

To extend the transversality to all of $\mathcal{A}^{\circ}_{M^k}$, we use the fact that $\mathcal{A}^{\circ}_{M^k}$ is $\sigma$-compact, i.e. is the union of countably many compact sets. In fact, any open subset of $\mathbb{R}^d$ is $\sigma$-compact. As a consequence, $\mathcal{A}^{\circ}_{M^k}$ is contained the union of countably many of open neighborhoods $U_{A_0}$ as constructed above. Since the set of matrices $A$ inside each $U_{A_0}$ violating the transversality statement has measure $0$, we conclude by countable additivity that the set of $A \in \mathcal{A}^{\circ}_{M^k}$ violating transversality has measure $0$ as well. This concludes the proof. □

**Lemma D.3.12.** *Fix linearly independent vectors $b_1, \ldots, b_d \in \mathbb{R}^d$ and let $A_1, \ldots, A_d$ be positive-definite matrices sampled independently from probability distributions on $\mathbb{R}^{\binom{d+1}{2}}$ which are absolutely continuous with respect to Lebesgue measure (i.e. which have a density). Then*

$$\left|\bigcap_{i=1}^{d} E_{A_i,b_i}\right| \geq 2$$

*holds almost surely.*

*Proof.* We proceed iteratively. For $k = d-1, \ldots, 1$ set

$$M^k = E_{A_1,b_1} \cap \cdots \cap E_{A_{d-k},b_{d-k}}.$$

We show by induction that $M^k$ is almost surely a smooth compact manifold of dimension $k$. The base case $k = d-1$ is obvious, and for smaller $k$, we have

$$M^k = M^{k+1} \cap E_{A,b}.$$

Lemma D.3.11 combined with Proposition D.3.4 now implies that $M^k$ is a smooth compact manifold of dimension $k$ almost surely, completing the inductive step.

Finally Proposition D.3.6 implies that assuming $M^1$ and $E_{A_d,b_d}$ are transverse (which holds with probability 1), the number of intersection points $|M^1 \cap E_{A_d,b_d}|$ is finite and even. Of course $|M^1 \cap E_{A_d,b_d}| = \left|\cap_{i=1}^{d} E_{A_i,b_i}\right|$. Since $\cap_{i=1}^{d} E_{A_i,b_i}$ trivially contains the origin, it must also contain another point. This completes the proof. □

*Proof of Theorem D.3.2.* Given $A_1, \dots, A_d$, consider a sequence of $d$-tuples $\left(A_1^{(k)}, \dots, A_d^{(k)}\right)_{k \geq 1}$ converging to $(A_1, \dots, A_d)$, i.e. satisfying

$$\lim_{k \to \infty} A_i^{(k)} = A_i$$

for each $i \in [d]$. Moreover assume that $\left|\bigcap_{i \in [d]} E_{A_i^{(k)}, b_i}\right| \geq 2$ for each $k$; such a sequence certainly exists by Lemma D.3.12. We also assume that the estimates

$$\ell \leq \lambda_d(A_i^{(k)}) \leq \lambda_1(A_i^{(k)}) \leq L \tag{D.67}$$

hold for some positive constants $\ell, L$ where $\lambda_d, \lambda_1$ are the minimum and maximum eigenvalues. This last assumption is without loss of generality by restricting the values of $k$ to $k \geq k_0$ for suitably large $k_0$. For each $k$, choose a non-zero point

$$x_k \in \bigcap_{i \in [d]} E_{A_i^{(k)}, b_i} \backslash \{0\}.$$

Such points exist because $|\bigcap_{i \in [d]} E_{A_i^{(k)}, b_i}| \geq 2$. We claim the norms $|x_k|$ are bounded away from infinity, bounded away from zero, and that any sub-sequential limit $x_*$ satisfies

$$x_* \in \bigcap_{i \in [d]} E_{A_i, b_i}.$$

It follows from the above claims that at least one sub-sequential limit $x_*$ exists (using the Bolzano-Weierstrass theorem) and that $|x_*| \neq 0$. Therefore the above claims suffice to finish the proof, and we now turn to their individual proofs.

First, since $x_k^\top A_i^{(k)} x_k \geq \lambda_d(A_i^{(k)})|x_k|^2 \geq \ell |x_k|^2$ and $|b_i^\top x_k| \leq |b_i^\top| \cdot |x_k|$, it follows that $|x_k| \leq \frac{|b_1|}{\ell}$ for all $k$, so in particular these norms are bounded above. Next we show the values $|x_k|$ are bounded away from $0$. Suppose for sake of contradiction that $|x_{a_j}| \to 0$ along some subsequence $(a_j)_{j \geq 1}$. Then

$$\langle b_i, x_{a_j} \rangle = x_{a_j}^\top A_i^{(a_j)} x_{a_j} \leq L |x_{a_j}|^2 = o(|x_{a_j}|).$$

Defining the rescaled unit vectors $\widehat{x}_{a_j} = \frac{x_{a_j}}{|x_{a_j}|}$, it follows that

$$\lim_{j \to \infty} \langle b_i, \widehat{x}_{a_j} \rangle = 0$$

for each $i$. As the $\widehat{x}_{a_j}$ are unit vectors, the Bolzano-Weierstrass theorem guarantees existence of a subsequential limit $\widehat{x}_*$ which is also a unit vector. It follows $\langle b_i, \widehat{x}_* \rangle = 0$ for all $i \in [d]$. However because the vectors $b_i$ are linearly independent, this implies $|\widehat{x}_*| = 0$ which is a contradiction. We conclude that $|x_k|$ is bounded away from $0$.

Finally we show that any subsequential limit satisfies $x_* \in E_{A,b}$. With $b$ fixed, observe that the functions $g_{A,b}(x) = x^\top Ax - b^\top x$ are uniformly Lipschitz for $A$ obeying the eigenvalue bound (Equation (D.67)) and $|x| \leq \frac{|b_1|}{\ell}$. It follows that

$$\lim_{k\to\infty} g_{A_i^{(k)},b_i}(x_*) = \lim_{k\to\infty} g_{A_i^{(k)},b_i}(x_k) = 0.$$

Having established the three claims we conclude the proof of Theorem D.3.2.

□

## D.4 Proof of Theorem 5.4.4

*Proof.* Define $A, B$ as the left $r$ and right $d_s$ columns of $S$. The optimal output is defined by

$$\begin{bmatrix} A^\top \\ B^\top \end{bmatrix} w^* = \begin{bmatrix} \Sigma_1^{-1}\mu_1 \\ 0 \end{bmatrix} \iff A^\top w^* = \Sigma_1^{-1}\mu_1, B^\top w^* = 0 \tag{D.68}$$

$$\iff \Sigma_1 A^\top w^* = \mu_1, P_B w^* = 0. \tag{D.69}$$

The algorithmic output satisfies

$$(I - P_B)A\Sigma_1 A^\top (I - P_B)w' = (I - P_B)A\mu_1, P_B w' = 0 \tag{D.70}$$

$$\implies (I - P_B)A\Sigma_1 A^\top w' = (I - P_B)A\mu_1, P_B w' = 0 \tag{D.71}$$

Multiple the first equation on the RHS by its pseudo-inverse

$$(A^\top(I - P_B)A)^{-1}A^\top(I - P_B)A\Sigma_1 A^\top w' = (A^\top(I - P_B)A)^{-1}A^\top(I - P_B)A\mu_1 \tag{D.72}$$

$$\implies \Sigma_1 A^\top w' = \mu_1. \tag{D.73}$$

□

## D.5 Additional Experimental Details

For Noised MNIST dataset, for each class $c \in \{0, \dots, 9\}$, we first generative a class signature $x_c \in \mathbb{R}^{28} \sim N(0, 2.5 I_{28})$. For each of the $E = 12$ groups, we generate a training spurious covariance $\Sigma_2^e = G_e G_e^\top$ and a test spurious covariance $\Sigma_2^{e\prime} = G_e' {G_e'}^\top$. The noise code for digit $c$ in training environment $e$ is drawn from

$\mathcal{N}(x_c, \Sigma_2^e)$. In test environment, the noise is drawn from $\mathcal{N}(x_{c'}, \Sigma_2^{e}{}')$ for random label $c' \sim \text{unif}\{0, \ldots, 9\})$.

We use SGD optimizer for both datasets. The hyperparameters are the coefficients for coral penalty, orthonormal penalty, and irm penalty $\lambda_{coral}, \lambda_{on}, \lambda_{irm}$, and learning rate $lr$. For each algorithm in Figures 5.1 and 5.2, we select penalization strengths from $\{0.1, 1, 10, 100\}$ and $lr$ from $\{0.1, 0.01, 0.001, 0.0001\}$ that achieves highest average test accuracy within 500 epochs (for Gaussian dataset) and 400 epochs (for Noised MNIST). Gaussian dataset has batch size 100 and Noised MNIST has batch size 1000 from each training environment.

The average test accuracies for each algorithm with error bars are shown in Figures 5.1 and 5.2. We fix the datasets and use different random seeds for algorithmic randomness. Error bar indicates mean and standard deviation across 5 runs.

The MLP architecture in Figure 5.2 is in Table D.1:

Table D.1: MLP network architectures for Noised MNIST

| Layers | 1 | 2 | 3 | 4 | 6 |
|---|---|---|---|---|---|
| Widths | 24 | 96,24 | 128,50,24 | 192,96,48,24 | 400,300,200,100,50,24 |

Table D.2: Matching features at 3 layers with identical widths does not have significant advantage over matching only at the last layer (CORAL).

| Layer widths | 24 | 128, 50, 24 | 24, 24, 24 |
|---|---|---|---|
| ERM | $58.6 \pm 0.4$ | $56.0 \pm 0.6$ | $62.1 \pm 0.6$ |
| IRM | $59.0 \pm 0.2$ | $56.1 \pm 0.6$ | $62.3 \pm 1.0$ |
| CORAL (only match last layer) | $69.1 \pm 1.0$ | $65.2 \pm 1.0$ | $67.2 \pm 0.4$ |
| CORAL (match-disjoint) | $69.1 \pm 1.0$ | $75.5 \pm 1.0$ | $70.6 \pm 0.9$ |
| CORAL (match-all) | $69.1 \pm 1.0$ | $77.9 \pm 0.4$ | $70.4 \pm 0.9$ |

To answer (Q5), we compare performances of algorithms on a 3-layer MLP that does not shrink feature dimensions (right column) with those on a 3-layer MLP that does (middle column) and a 1-layer MLP (left column) in Table D.2. Results show that without shrinking feature dimensions, matching at multiple layers does not improve over naive CORAL on a smaller architecture.

No run in any of our experiments take more than 10 minutes on a single GPU. MNIST dataset [LeCun et al., 1998] is made available under the terms of the Creative Commons Attribution-Share Alike 3.0 license.

# Appendix E

# Appendix for Chapter 6

## E.1 Proof of Theorem 1

**Theorem 6.4.2.** *Suppose* $\sigma_{\max} \geq \sigma_{\min} > 0$ *such that* $\forall e \in \mathcal{E},\ \sigma_{\min} I \preceq \nabla^2 f_e \preceq \sigma_{\max} I$. *Define* $g$ *as the minimum gradient norm that is guaranteed to be forceable by the adversary:* $g := \min_{\beta \in B} \max_{\lambda \in \Delta_E} \|\nabla f(\beta)\|_2$. *Then for all* $t \in \mathbb{N}$ *it holds that* $V_t > \frac{g^2 \sigma_{\min}}{16 \sigma_{\max}^2} \log t$.

*Proof.* Define $F_t(z) = \sum_{s=1}^{t} f_s(z)$; since each $f$ is convex, this sum is convex as well. Let $\beta^*_{t-1}$ be the minimizer of $F_{t-1}$ (by Lemma E.3.1, this will lie in $B$), and let $z \in B$ be arbitrary. Finally, note that $\nabla^2 F_t \preceq t\sigma_{\max} I$. Then we have the following Taylor expansion:

$$
\begin{aligned}
F_t(z) &= F_{t-1}(z) + f_t(z) && \text{(E.1)}\\
&= F_{t-1}(\beta^*_{t-1} + (z - \beta^*_{t-1})) + f_t(z) && \text{(E.2)}\\
&\leq F_{t-1}(\beta^*_{t-1}) + \nabla F_{t-1}(\beta^*_{t-1})^T (z - \beta^*_{t-1})\\
&\qquad + \frac{(t-1)\sigma_{\max}}{2} \|z - \beta^*_{t-1}\|_2^2 + f_t(z) && \text{(E.3)}\\
&= F_{t-1}(\beta^*_{t-1}) + \frac{(t-1)\sigma_{\max}}{2} \|z - \beta^*_{t-1}\|_2^2 + f_t(z), && \text{(E.4)}
\end{aligned}
$$

where we have used the fact that $\nabla F_{t-1}(\beta^*_{t-1}) = 0$ by definition. Thus,

$$
\begin{aligned}
\sum_{s=1}^{t} f_s(\widehat{\beta}_s) - F_t(z) \geq &\left( \sum_{s=1}^{t-1} f_s(\widehat{\beta}_s) - F_{t-1}(\beta^*_{t-1}) \right)\\
&+ \left(f_t(\widehat{\beta}_t) - f_t(z) - \frac{(t-1)\sigma_{\max}}{2} \|z - \beta^*_{t-1}\|_2^2\right). && \text{(E.5)}
\end{aligned}
$$

Then we can write

$$V_t = \min_{\widehat{\beta}_1 \in B} \max_{\lambda_1} \dots \min_{\widehat{\beta}_t \in B} \max_{\lambda_t, z \in B} \left( \sum_{s=1}^{t} f_t(\widehat{\beta}_t) - F_t(z) \right) \tag{E.6}$$

$$\geq \min_{\widehat{\beta}_1 \in B} \max_{\lambda_1} \dots \min_{\widehat{\beta}_{t-1} \in B} \max_{\lambda_{t-1}} \Bigg[ \left( \sum_{s=1}^{t-1} f_s(\widehat{\beta}_s) - F_{t-1}(\beta^*_{t-1}) \right) \tag{E.7}$$

$$+ \min_{\widehat{\beta}_t \in B} \max_{\lambda_t, z \in B} \left( f_t(\widehat{\beta}_t) - f_t(z) - \frac{(t-1)\sigma_{\max}}{2} \|z - \beta^*_{t-1}\|_2^2 \right) \Bigg]. \tag{E.8}$$

Thus, by lower bounding the second term, we can unroll the recursion and lower bound the total regret. In particular, showing a bound of $\Omega(\frac{1}{t})$ will result in an overall regret lower bound of $\Omega(\log T)$, which would imply that ERM achieves minimax-optimal rates for OOD generalization (this is also how we prove Corollary 6.4.3).

We proceed by lower bounding the inner optimization term. We consider two possibilities for the choice of $\widehat{\beta}_t$. Suppose $\|\widehat{\beta}_t - \beta^*_{t-1}\|_2^2 \geq \frac{g^2}{8t\sigma^2_{\max}}$. Then by choosing $z = \beta^*_{t-1}$ the inner term can be lower bounded by $\min_{\widehat{\beta}_t \in B} \max_{\lambda_t} \bigg( f_t(\widehat{\beta}_t) - f_t(\beta^*_{t-1}) \bigg)$. Taylor expanding $f_t$ around $\beta^*_{t-1}$ gives

$$f_t(\widehat{\beta}_t) - f_t(\beta^*_{t-1}) \geq \nabla f_t(\beta^*_{t-1})^T (\widehat{\beta}_t - \beta^*_{t-1}) + \frac{\sigma_{\min}}{2} \|\widehat{\beta}_t - \beta^*_{t-1}\|_2^2. \tag{E.9}$$

By Lemma E.3.2, the adversary can always play $\lambda_t$ such that $\nabla f_t(\beta^*_{t-1}) = 0$. So plugging this in we get

$$\min_{\widehat{\beta}_t \in B} \max_{\lambda_t} \left( f_t(\widehat{\beta}_t) - f_t(\beta^*_{t-1}) \right) \geq \frac{\sigma_{\min}}{2} \|\widehat{\beta}_t - \beta^*_{t-1}\|_2^2 \tag{E.10}$$

$$\geq \frac{g^2 \sigma_{\min}}{16 t \sigma^2_{\max}}. \tag{E.11}$$

Now consider the case where $\|\widehat{\beta}_t - \beta^*_{t-1}\|_2^2 < \frac{g^2}{8t\sigma^2_{\max}}$. Suppose the adversary plays any $\lambda_t$ such that $\|\nabla f_t(\widehat{\beta}_t)\|_2 \geq g$ (by definition, such a choice is always possible). Here we again split on cases, considering the possible values of $\nabla f_t(\beta^*_{t-1})^T (\widehat{\beta}_t - \beta^*_{t-1})$:

**Case 1:** $\nabla f_t(\beta^*_{t-1})^T (\widehat{\beta}_t - \beta^*_{t-1}) \geq \frac{g^2 \sigma_{\min}}{16 t \sigma^2_{\max}}$

Following the same steps as previously, we find the lower bound

$$f_t(\widehat{\beta}_t) - f_t(\beta^*_{t-1}) \geq \nabla f_t(\beta^*_{t-1})^T(\widehat{\beta}_t - \beta^*_{t-1}) + \frac{\sigma_{\min}}{2}\|\widehat{\beta}_t - \beta^*_{t-1}\|_2^2 \tag{E.12}$$

$$\geq \nabla f_t(\beta^*_{t-1})^T(\widehat{\beta}_t - \beta^*_{t-1}) \tag{E.13}$$

$$\geq \frac{g^2\sigma_{\min}}{16t\sigma^2_{\max}}. \tag{E.14}$$

**Case 2:** $\nabla f_t(\beta^*_{t-1})^T(\widehat{\beta}_t - \beta^*_{t-1}) < \frac{g^2\sigma_{\min}}{16t\sigma^2_{\max}}$

In this case the lower bound follows directly from Lemma E.3.3.

Thus the lower bound is shown in all cases; it follows that

$$V_t \geq \min_{\widehat{\beta}_1 \in B} \max_{\lambda_1} \ldots \min_{\widehat{\beta}_{t-1} \in B} \max_{\lambda_{t-1}} \left[\left(\sum_{s=1}^{t-1} f_s(\widehat{\beta}_s) - F_{t-1}(\beta^*_{t-1})\right) + \frac{g^2\sigma_{\min}}{16t\sigma^2_{\max}}\right] \tag{E.15}$$

$$= \min_{\widehat{\beta}_1 \in B} \max_{\lambda_1} \ldots \min_{\widehat{\beta}_{t-1} \in B} \max_{\lambda_{t-1}} \left[\sum_{s=1}^{t-1} f_s(\widehat{\beta}_s) - F_{t-1}(\beta^*_{t-1})\right] + \frac{g^2\sigma_{\min}}{16t\sigma^2_{\max}} \tag{E.16}$$

$$= V_{t-1} + \frac{g^2\sigma_{\min}}{16t\sigma^2_{\max}}. \tag{E.17}$$

Expanding the recursion finishes the proof. □

## E.2 Proof of Existence for Theorem 6.4.4

We restate Theorem 6.4.4 for convenience:

**Theorem 6.4.4.** *No algorithm can guarantee sublinear regret against bounded affine combinations of a finite set of strongly convex losses.*

In the main body, we prove the primary claim. Here we include proof of the existence of a regression task over a set of distributions which induces the loss functions we construct in our proof.

*Proof.* Suppose we are regressing labels $y \in \mathbb{R}$ on observations $z \in \mathbb{R}^2$ with squared loss. We'll define our classifier with a parameter $\beta$ such that given an observation $(z_1, z_2)$ we predict $\beta^2 z_1 + \beta z_2$. This is of course an unusual regression setup, but we're just giving an existence proof for a simple lower bound.

The first environment will assign all its probability mass to a single example $(z_1, z_2, y) = (0, 1, 0)$. Thus, if we choose a parameter $\beta$, in this environment we will suffer risk $\mathbb{E}[(\beta z_1^2 + \beta z_2 - y)^2] = \beta^2$. This produces the first environment, loss $f_{e_1}(\beta) = \beta^2$.

We define the second environment as having two possible samples: one is $(z_1, z_2, y) = (0, \sqrt{\frac{2\alpha+1}{2\alpha}}, 0)$ and the other is $(z_1, z_2, y) = (\sqrt{\frac{2\alpha+1}{2\alpha}}, 0, 0)$. Thus, the first sample induces loss $\frac{2\alpha+1}{2\alpha}\beta^2$, and the second induces loss $\frac{2\alpha+1}{2\alpha}\beta^4$. Now for the probabilities: we assign probability $\frac{1}{2\alpha+1}$ to the first point and $\frac{2\alpha}{2\alpha+1}$ to the second point. Clearly these sum to 1, and taking the expectation over losses we see that the overall risk is $\beta^4 + \frac{1}{2\alpha}\beta^2$, as desired. □

## E.3 Lemmas

**Lemma 6.4.1.** *Recall $\mathcal{R}^e(\beta)$ is defined as the risk of $\beta$ on the distribution $p^e$. Then for all $\lambda \in \Delta_E$, it holds that $\mathcal{R}^\lambda(\beta) = \sum_{e\in\mathcal{E}} \lambda_e \mathcal{R}^e(\beta)$.*

*Proof.* Using Fubini's theorem, we have

$$\mathcal{R}^\lambda(\beta) = \int_{\mathcal{X}\times\mathcal{Y}} \left[\sum_{e\in\mathcal{E}} \lambda_e p^e(x,y)\right] \ell(\beta, (x,y))\, d(x,y) \tag{E.18}$$

$$= \sum_{e\in\mathcal{E}} \lambda_e \int_{\mathcal{X}\times\mathcal{Y}} p^e(x,y)\ell(\beta,(x,y))\, d(x,y) \tag{E.19}$$

$$= \sum_{e\in\mathcal{E}} \lambda_e \mathcal{R}^e(\beta). \qquad \square$$

**Lemma E.3.1.** *For any $F_t = \sum_{s=1}^t f_t$, there exist convex coefficients $\widehat{\lambda}$ such that*

$$F_t = t\sum_{e\in\mathcal{E}} \widehat{\lambda}_e f_e. \tag{E.20}$$

*Proof.* Every loss function $f_t$ can be written as a convex combination of the original environment losses:

$$f_t = \sum_{e\in\mathcal{E}} \lambda_{t,e} f_e. \tag{E.21}$$

So, write

$$F_t = \sum_{s=1}^t f_t = \sum_{s=1}^t \sum_{e\in\mathcal{E}} \lambda_{t,e} f_e = \sum_{e\in\mathcal{E}} \left(\sum_{s=1}^t \lambda_{t,e}\right) f_e. \tag{E.22}$$

Clearly, $\sum_{e\in\mathcal{E}} \left(\sum_{s=1}^t \lambda_{t,e}\right) = t$. So, defining $\widehat{\lambda}_e := \frac{1}{t}\left(\sum_{s=1}^t \lambda_{t,e}\right)$ gives the desired result. □

**Lemma E.3.2.** *For any solution $\beta^*_{t-1}$ which minimizes the sum of previously seen losses $F_{t-1}$, there exists a convex combination of losses $f_t$ playable by the adversary for which $\nabla f_t(\beta^*_{t-1}) = 0$.*

*Proof.* By Lemma E.3.1, we can write $F_{t-1} = (t-1)\sum_{e\in\mathcal{E}} \widehat{\lambda}_e f_e$ for some convex coefficients $\widehat{\lambda}$. Define $f_t = \sum_{e\in\mathcal{E}} \widehat{\lambda}_e f_e = \frac{1}{t-1}F_{t-1}$. Since $\beta^*_{t-1}$ minimizes $F_{t-1}$ it follows that

$$\nabla f_t(\beta^*_{t-1}) = \frac{1}{t-1}\nabla F_{t-1}(\beta^*_{t-1}) = 0. \tag{E.23}$$

□

**Lemma E.3.3.** *Let $\widehat{\beta}_t$, $\lambda_t$ be such that $\|\widehat{\beta}_t - \beta^*_{t-1}\|_2^2 < \frac{g^2}{8t\sigma^2_{\max}}$ and $\|\nabla f_t(\widehat{\beta}_t)\|_2 \geq g$. Define $z := \beta^*_{t-1} - c\nabla f_t(\widehat{\beta}_t)$, where $c := 1/2t\sigma_{\max}$. If $\nabla f_t(\beta^*_{t-1})^T(\widehat{\beta}_t - \beta^*_{t-1}) < \frac{g^2\sigma_{\min}}{16t\sigma^2_{\max}}$, then*

$$f_t(\widehat{\beta}_t) - f_t(z) - \frac{(t-1)\sigma_{\max}}{2}\|z - \beta^*_{t-1}\|_2^2 \geq \frac{g^2\sigma_{\min}}{16t\sigma^2_{\max}}. \tag{E.24}$$

*Proof.* Expanding $f_t$ around $\widehat{\beta}_t$,

$$f_t(\widehat{\beta}_t) - f_t(z) \geq -\nabla f_t(\widehat{\beta}_t)^T(z - \widehat{\beta}_t) - \frac{\sigma_{\max}}{2}\|z - \widehat{\beta}_t\|_2^2, \tag{E.25}$$

which gives

$$\begin{aligned}
&f_t(\widehat{\beta}_t) - f_t(z) - \frac{(t-1)\sigma_{\max}}{2}\|z - \beta^*_{t-1}\|_2^2 \\
&\geq \nabla f_t(\widehat{\beta}_t)^T(\widehat{\beta}_t - z) - \frac{\sigma_{\max}}{2}\left(\|z - \widehat{\beta}_t\|_2^2 + (t-1)\|z - \beta^*_{t-1}\|_2^2\right) \\
&= \nabla f_t(\widehat{\beta}_t)^T(\widehat{\beta}_t - \beta^*_{t-1} + c\nabla f_t(\widehat{\beta}_t)) \\
&\qquad - \frac{\sigma_{\max}}{2}\left(\|\beta^*_{t-1} - \widehat{\beta}_t - c\nabla f_t(\widehat{\beta}_t)\|_2^2 + (t-1)\|c\nabla f_t(\widehat{\beta}_t)\|_2^2\right).
\end{aligned} \tag{E.26}$$

By the triangle inequality,

$$\|\beta^*_{t-1} - \widehat{\beta}_t - c\nabla f_t(\widehat{\beta}_t)\|_2 \leq \|\beta^*_{t-1} - \widehat{\beta}_t\|_2 + c\|\nabla f_t(\widehat{\beta}_t)\|_2, \tag{E.27}$$

and therefore

$$\frac{1}{2}\|\beta^*_{t-1} - \widehat{\beta}_t - c\nabla f_t(\widehat{\beta}_t)\|_2^2 \leq \|\beta^*_{t-1} - \widehat{\beta}_t\|_2^2 + c^2\|\nabla f_t(\widehat{\beta}_t)\|_2^2. \tag{E.28}$$

Continuing with the lower bound in Equation (E.26),

$$\geq \nabla f_t(\widehat{\beta}_t)^T(\widehat{\beta}_t - \beta^*_{t-1}) + c\|\nabla f_t(\widehat{\beta}_t)\|_2^2 - \sigma_{\max}\left(\|\beta^*_{t-1} - \widehat{\beta}_t\|_2^2 + c^2\|\nabla f_t(\widehat{\beta}_t)\|_2^2\right) - \frac{(t-1)\sigma_{\max}c^2}{2}\|\nabla f_t(\widehat{\beta}_t)\|_2^2 \tag{E.29}$$

$$\geq \nabla f_t(\widehat{\beta}_t)^T(\widehat{\beta}_t - \beta^*_{t-1}) + \left(c - \frac{1}{8t\sigma_{\max}} - \frac{(t+1)c^2\sigma_{\max}}{2}\right)\|\nabla f_t(\widehat{\beta}_t)\|_2^2, \tag{E.30}$$

where we've used the upper bound on $\|\beta^*_{t-1} - \widehat{\beta}_t\|_2^2$ and simplified. Recalling that $c = \frac{1}{2t\sigma_{\max}}$ and noting that $\frac{t+1}{t^2} \leq \frac{2}{t}$,

$$= \nabla f_t(\widehat{\beta}_t)^T(\widehat{\beta}_t - \beta^*_{t-1}) + \left(\frac{1}{2t\sigma_{\max}} - \frac{1}{8t\sigma_{\max}} - \frac{(t+1)}{8t^2\sigma_{\max}}\right)\|\nabla f_t(\widehat{\beta}_t)\|_2^2 \tag{E.31}$$

$$\geq \nabla f_t(\widehat{\beta}_t)^T(\widehat{\beta}_t - \beta^*_{t-1}) + \frac{\|\nabla f_t(\widehat{\beta}_t)\|_2^2}{8t\sigma_{\max}} \tag{E.32}$$

$$\geq \nabla f_t(\widehat{\beta}_t)^T(\widehat{\beta}_t - \beta^*_{t-1}) + \frac{g^2}{8t\sigma_{\max}}. \tag{E.33}$$

By strong convexity,

$$(\nabla f_t(\beta^*_{t-1}) - \nabla f_t(\widehat{\beta}_t))^T(\beta^*_{t-1} - \widehat{\beta}_t) \geq \sigma_{\min}\|\beta^*_{t-1} - \widehat{\beta}_t\|_2^2, \tag{E.34}$$

and therefore

$$\nabla f_t(\widehat{\beta}_t)^T(\widehat{\beta}_t - \beta^*_{t-1}) \geq \sigma_{\min}\|\beta^*_{t-1} - \widehat{\beta}_t\|_2^2 - \nabla f_t(\beta^*_{t-1})^T(\widehat{\beta}_t - \beta^*_{t-1}) \tag{E.35}$$

$$> -\frac{g^2\sigma_{\min}}{16t\sigma_{\max}^2}, \tag{E.36}$$

where the second inequality is due to the assumption in the Lemma statement. Plugging this in above gives

$$\nabla f_t(\widehat{\beta}_t)^T(\widehat{\beta}_t - \beta^*_{t-1}) + \frac{g^2}{8t\sigma_{\max}} > -\frac{g^2\sigma_{\min}}{16t\sigma_{\max}^2} + \frac{g^2}{8t\sigma_{\max}} \tag{E.37}$$

$$\geq \frac{g^2\sigma_{\min}}{8t\sigma_{\max}^2} - \frac{g^2\sigma_{\min}}{16t\sigma_{\max}^2} \tag{E.38}$$

$$= \frac{g^2\sigma_{\min}}{16t\sigma_{\max}^2}, \tag{E.39}$$

completing the proof. □

# Appendix F

# Appendix for Chapter 7

## F.1 Additional Discussion

### F.1.1 Connection to Anchor Regression

The DARE objective for linear regression is written

$$\min_{\beta} \sum_{e \in \mathcal{E}} \mathbb{E}_{p^e}[(\beta^\top \Sigma_e^{-1/2}(x-\mu_e) - y)^2] \qquad \text{s.t.} \quad \beta^\top \Sigma_e^{-1/2}\mu_e = 0. \ \forall e \in \mathcal{E}. \tag{F.1}$$

The idea of adjusting for domain projections has similarities to Anchor Regression [Rothenhäusler et al., 2021], an objective which linearly regresses separately on the projection and rejection of the data onto the span of a set of *anchor variables*. These variables represent some (known) measure of variability across the data, and the resulting solution enjoys robustness to pointwise-additive shifts in the underlying SCM. If we define the anchor variable to be a one-hot vector indicating a sample's environment, the Anchor Regression objective minimizes

$$\frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \mathbb{E}_{p^e} \left[ \ell(\beta^\top (x-\mu_e),\ y-\mu_{y,e}) + \gamma \ell(\beta^\top \mu_e,\ \mu_{y,e})) \right], \tag{F.2}$$

where $\ell$ is the squared loss and $\mu_{y,e} = \mathbb{E}_{p^e}[y]$. Here we see that Anchor Regression is "adjusting" in a sense, by regressing on the residuals, though the objective also regresses the mean prediction onto the target mean. Unfortunately, this requires access to the target mean, which is unavailable in logistic regression due to the lack of a good estimator for $\mathbb{E}\left[\log \frac{p(y=1|x)}{p(y=-1|x)}\right]$. Nevertheless, if we (i) assume the feature covariance is fixed for all environments and (ii) assume the target mean is zero for all environments, we observe that the above objective becomes equivalent to the Lagrangian form of DARE for linear regression (F.1). Alternatively, we could imagine combining the two by keeping Anchor Regression's use of separate target means while adding DARE's feature covariance whitening which would give

$$\frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}}\mathbb{E}_{p^e}\left[\ell(\beta^\top\Sigma_e^{-1/2}(x-\mu_e),\ y-\mu_{y,e})+\gamma\ell(\beta^\top\Sigma_e^{-1/2}\mu_e,\ \mu_{y,e})\right], \quad \text{(F.3)}$$

However, this still requires us to estimate the target mean, so it is unclear if or how this objective could be applied to the task of classification.

### F.1.2 On Assumption 7.5.1 and the Conditions Assumed Without Loss of Generality

It may not be immediately clear why it is reasonable to assume both $V = I$ and $\Sigma_\epsilon = I$ WLOG simultaneously, so we clarify this point here. It is crucial to observe that $\beta^*$ and $A_e$ do not need to be directly identifiable, because we only care about the predictive distribution $\beta^{*T}\epsilon$. That is, we only need $A_e$ to be identifiable from $x$ up to equivalence of this distribution. So if for example we recover $A_e$ up to some invertible transformation: $\widehat{A}_e = A_e M$, this is not at all a problem because we also learn the corresponding $\widehat{\beta} = M^\top\beta^*$ such that $\widehat{\beta}^\top\widehat{A}_e^{-1}x = \beta^{*T}MM^{-1}A_e^{-1}x = \beta^{*T}\epsilon$.

In particular, suppose instead $V \neq I$ and $\mathbb{E}[\epsilon_0\epsilon_0^\top] = \Sigma_0$. Then we can simply reparameterize as $\epsilon_0 \to \Sigma_0^{-1/2}\epsilon_0$, $b_e \to \Sigma_0^{-1/2}b_e$, $\beta^* \to \Sigma_0^{1/2}\beta^*$, $A_e \to A_eV^{-1}\Sigma_0^{1/2}$. It is easy to see this results in the same observed distribution over $(x, y)$, and further that learning $\beta^*$ to predict on $\epsilon_0$ is the same as learning $\Sigma_0^{1/2}\beta^*$ to predict on $\Sigma_0^{-1/2}\epsilon_0$. So now we've reduced this to a setting where $\mathbb{E}[\epsilon_0\epsilon_0^\top] = I$ but perhaps $V \neq I$. However, when $V \neq I$ it represents precisely the unidentifiable transformation $M$ above, which does not pose a problem for prediction because it will not change in future environments. In other words, the DARE solution is technically $\widehat{\Pi}V^\top\beta^*$, where $V$ is the shared right singular vectors of all domain-specific transformations $A_e$. We assume $V = I$ WLOG so as to not carry around this additional term when it has no effect on the risk of the recovered predictor.

We emphasize that some assumption of the same flavor as Assumption 7.5.1 is *fundamentally necessary* when dealing with latent variable models of distribution shift. To see this, consider training on data from a linear regression model with latent variables $z \sim \mathcal{N}(0, I)$ where we observe $(x, y) = (Az, \beta^\top z)$. Now suppose we encounter a test distribution with $(x, y) = (\tilde{A}z, \beta^\top z)$, where $A$ has different right singular vectors from $\tilde{A}$. The resulting test prediction task would be *completely detached from the training task* and we could not possibly hope to generalize. In fact, it is encouraging that our method *only* requires $V$ to be constant, and not the entire matrix $A$, as is assumed in prior work [Rothenhäusler et al., 2021].

## F.2 Proofs of Main Results

### F.2.1 Notation

We use capital letters to denote matrices and lowercase to denote vectors or scalars, where the latter should be clear from context. $\|\cdot\|_2$ refers to the usual vector norm, or spectral norm for matrices. For a matrix $M$, we use $\lambda_{\max}(M)$ to mean its maximum eigenvalue—the minimum is defined analogously. We write the pseudo-inverse as $M^\dagger$. For a collection of samples $\{x_i\}_{i=1}^n$, we frequently make use of the sample mean, $\widehat{\mu} := \frac{1}{n}\sum x_i$, and the sample covariance, $\widehat{\Sigma} := \frac{1}{n}\sum(x_i - \bar{\mu})(x_i - \bar{\mu})^\top$. The notation $\lesssim$ means less than or equal to up to constant factors.

### F.2.2 Statement and Proof of Lemma F.2.1, and Discussion of Related Results

**Lemma F.2.1.** *Assume our data follows a logistic regression model with regression vector $\beta^*$ and covariates $z \sim \mathcal{N}(0, I)$: $\log \frac{p(y=1|z)}{p(y=-1|z)} = \beta^{*T} z$. Then the solution to the dimension-constrained logistic regression problem*

$$\arg\min_\beta \ \mathbb{E}_{z,y}[-\log \sigma(y\beta^\top z)] \qquad \text{s.t. } \beta_{S^c} = \mathbf{0}, \tag{F.4}$$

*where $S \subseteq [d]$ indexes a subset of the dimensions, is equal to $\alpha\beta^*_S$ for some $\alpha \in (0, 1]$.*

*Proof.* The logistic regression model can be rewritten:

$$y \mid z = \mathbf{1}\{\beta^{*T} z + \epsilon > 0\}, \tag{F.5}$$

where $\epsilon$ is drawn from a standard logistic distribution. If we are restricted to not use $z_{S^c}$, we can see that these can be modeled simply as an additional noise term. Thus, our new model is

$$y \mid z = \mathbf{1}\{\beta_S^{*T} z_S + \epsilon + \tau > 0\}, \tag{F.6}$$

where $\tau := \beta_{S^c}^{*T} z_{S^c} \sim p$ is symmetric zero-mean noise, independent of $z_S$. Because we are now modeling the other dimensions as noise, moving forward we will drop the $S$ subscript, writing simply $\beta^{*T}z$. Define $F, f$ as the CDF and PDF of the distribution of $\epsilon' := \epsilon + \tau$. Then the MLE population objective can be written

$$\begin{aligned}\mathcal{L}(\beta) = -\mathbb{E}_z\big[\mathbb{E}_{\epsilon'\sim f(\epsilon')}[\mathbf{1}\{\beta^{*T}z + \epsilon' > 0\}\log\sigma(\beta^\top z) \\ + \mathbf{1}\{-(\beta^{*T}z + \epsilon') > 0\}\log\sigma(-\beta^\top z)]\big].\end{aligned} \tag{F.7}$$

For a fixed $z$, note that $\mathbb{E}_{\epsilon'}[\mathbf{1}\{\beta^{*T}z + \epsilon' > 0\}] = \mathbb{P}(\epsilon' \geq -\beta^{*T}z) = F(\beta^{*T}z)$ (since $f$ is symmetric), and therefore taking the derivative of this objective we get

$$\begin{aligned}\nabla_\beta\mathcal{L}(\beta) &= -\nabla_\beta\mathbb{E}_z[F(\beta^{*T}z)\log\sigma(\beta^\top z) + F(-\beta^{*T}z)\log\sigma(-\beta^\top z)] && \text{(F.8)}\\ &= \mathbb{E}_z[z\cdot\big(F(-\beta^{*T}z)\sigma(\beta^\top z) - F(\beta^{*T}z)\sigma(-\beta^\top z)\big)] && \text{(F.9)}\end{aligned}$$

Because $f$ is symmetric, we have $F(z) = 1 - F(-z)$, giving

$$\nabla_\beta\mathcal{L}(\beta) = \mathbb{E}_z\left[z\cdot\Big(\sigma(\beta^\top z) - F(\beta^{*T}z)\Big)\right]. \tag{F.10}$$

Consider the directional derivative of the loss in the direction $\beta^*$, at the point $\beta = \alpha\beta^*$:

$$\beta^{*T}\nabla_\beta\mathcal{L}(\alpha\beta^*) = \mathbb{E}_z\left[\beta^{*T}z\cdot\big(\sigma(\alpha\beta^{*T}z) - F(\beta^{*T}z)\big)\right]. \tag{F.11}$$

Because $F$ is the CDF of a logistic distribution convolved with $p$, by Fubini's theorem we have

$$\begin{aligned}F(z) &= \int_{-\infty}^{z} f(z)\,dz && \text{(F.12)}\\ &= \int_{-\infty}^{z}\left[\int_{-\infty}^{\infty} p(\tau)\sigma'(z-\tau)\,d\tau\right]dz && \text{(F.13)}\\ &= \int_{-\infty}^{\infty} p(\tau)\left[\int_{-\infty}^{z-\tau}\sigma'(\omega)\,d\omega\right]d\tau && \text{(F.14)}\\ &= \int_{-\infty}^{\infty} p(\tau)\sigma(z-\tau) && \text{(F.15)}\\ &= \mathbb{E}_{\tau\sim p}[\sigma(z-\tau)]. && \text{(F.16)}\end{aligned}$$

Further, because $p$ is symmetric, this is equal to $\frac{1}{2}\left(\mathbb{E}_{\tau\sim p}[\sigma(z-\tau)] + \mathbb{E}_{\tau\sim p}[\sigma(z+\tau)]\right)$. Thus, we have

$$\begin{aligned}\beta^{*T}\nabla_\beta\mathcal{L}(\alpha\beta^*) = \mathbb{E}_z\Bigg[\beta^{*T}z\cdot \\ \mathbb{E}_{\tau\sim p}\Bigg[\sigma(\alpha\beta^{*T}z) - \frac{1}{2}\big(\sigma(\beta^{*T}z-\tau) + \sigma(\beta^{*T}z+\tau)\big)\Bigg]\Bigg].\end{aligned} \tag{F.17}$$

We first consider the case where $\alpha = 1$. When $\beta^{*T} z > 0$, the term inside the expectation is positive for all $\tau \neq 0$, and vice-versa when $\beta^{*T} z < 0$ (this can be verified by writing the difference as a function of $\beta^{*T} z, \tau$, and observing that all the terms are non-negative except for a factor of $e^{\beta^{*T} z} - 1$). It follows that at the point $\beta = \beta^*$, $-\beta^*$ is a descent direction. Furthermore, since the objective is continuous in $\alpha$, we can follow this direction by reducing $\alpha$ (that is, moving in the direction $-\beta^*$) until the directional derivative vanishes.

Next, consider $\alpha = 0$. Then the directional derivative is

$$\beta^{*T} \nabla_\beta \mathcal{L}(0) = \frac{1}{2} \mathbb{E}_z \left[ \beta^{*T} z \cdot \mathbb{E}_{\tau \sim p} \left[ 1 - (\sigma(\beta^{*T} z - \tau) + \sigma(\beta^{*T} z + \tau)) \right] \right]. \tag{F.18}$$

Here, when $\beta^{*T} z > 0$ the inner term is negative, and vice-versa for $\beta^{*T} z < 0$, implying that the directional derivative is now negative. Because the objective is convex, it follows that the optimal choice for $\alpha$ lies in $(0, 1]$, being equal to 1 when $\tau = 0$ almost surely.

It remains to show that the optimal vector has no other component orthogonal to $\beta^*$—in other words, that the solution is precisely $\alpha\beta^*$. For isotropic Gaussian $z$, we have for any $\delta$ perpendicular to $\beta^*$ that $\mathbb{E}[\delta^\top z | \beta^{*T} z] = 0$. Therefore, the gradient in the direction $\delta$ is

$$\begin{aligned}
&\mathbb{E}_z[\delta^\top z \cdot (\sigma(\alpha\beta^{*T} z) - F(\beta^{*T} z))] && \text{(F.19)}\\
&= \mathbb{E}_{\beta^{*T} z}[\mathbb{E}_{\delta^\top z | \beta^{*T} z}[\delta^\top z](\sigma(\alpha\beta^{*T} z) - F(\beta^{*T} z))] && \text{(F.20)}\\
&= 0. && \text{(F.21)}
\end{aligned}$$

Since $\beta^*$ and all orthogonal directions form a complete basis, it follows that $\nabla \mathcal{L}(\alpha\beta^*) = 0$ and therefore that $\alpha\beta^*$ is the optimal solution. □

Though we prove this lemma only for Gaussian $z$, we found empirically that the result approximately holds whenever $z$ is dimension-wise independent and symmetric about the origin. We believe this is a consequence of the Central Limit Theorem: our proof relies on the conditional expectation of inner products with $z$ which converge to Gaussian in distribution as the dimensionality of $z$ grows.

We observe that Li and Duan [1989] prove a much simpler result under a general "linear conditional expectation" condition which is similar to the property we exploit regarding zero-mean conditional expectation of orthogonal inner products with isotropic Gaussians. Their result is more general, but it allows for any value of $\alpha$, including negative. In this case, we would actually be recovering the *opposite* effects of the ground truth, which is clearly insufficient for test-time prediction. Heagerty and Zeger [2000] give an analytical closed-form for the solution under a probit model with Gaussian noise; since this is not a logistic model, the Gaussian noise represents significantly less model misspecification, which explains why the exact closed-form is recoverable.

### F.2.3 Proof of Theorem 7.5.2

**Theorem 7.5.2.** (Closed-form solution to the DARE population objective). *Under model (Equation (7.2)), the solution to the* DARE *population objective (Equation (F.1)) for linear regression is* $\widehat{\Pi}\beta^*$*. If* $\epsilon$ *is Gaussian, then the solution for logistic regression (Equation (7.1)) is* $\alpha\widehat{\Pi}\beta^*$ *for some* $\alpha \in (0, 1]$.

*Proof.* Observe that under the constraint, regressing on the centered observations is equivalent to regressing on the non-centered observations (since the mean must have no effect on the output), so the solutions to these two objectives must be the same and have the same minimizers. We therefore consider the solution to the DARE objective but on non-centered observations.

It is immediate that the unconstrained solution to the DARE population objective on non-centered data is $\beta^*$ for both linear and logistic regression. For linear regression, we observe that because the adjusted covariates in each environment have identity covariance, the excess training risk of a predictor $\beta$ is exactly $\|\beta - \beta^*\|_2^2$. Therefore, the solution can be rewritten

$$\begin{aligned} \min_{\beta} \quad & \|\beta - \beta^*\|_2^2 \\ \text{s.t.} \quad & \beta^\top \Sigma_e^{-1/2} \mu_e = 0. \;\; \forall e \in \mathcal{E}. \end{aligned}$$

Recalling that $\Sigma_e^{-1/2}\mu_e = b_e$, the constraint can be written in matrix form as $\mathbb{B}^\top \beta = \mathbf{0}$, and thus we see that the solution is the $\ell_2$-norm projection of $\beta^*$ onto the nullspace of $\mathbb{B}^\top$ (i.e., the intersection of the nullspaces of the $b_e$). By definition, this is given by $(I - \mathbb{B}\mathbb{B}^\dagger)\beta^* = \widehat{\Pi}\beta^*$.

To derive the closed-form for logistic regression, write the spectral decomposition $\mathbb{B} = UDV^\top$, and consider regressing on $U^\top \epsilon$ instead of $\epsilon$. As the predictor only affects the objective through its linear projection, the solution to this objective will be $U^\top$ times the solution to the original objective (that is, for all vectors $v$, $(U^\top \beta)^\top U^\top v = \beta^\top v$). We will denote parameters for the rotated objective with a tilde, e.g. $\tilde{v} := U^\top v$.

The constraint in Equation (7.1) is equivalent to requiring that the mean projection is a constant vector $c\mathbf{1}$ and, with the inclusion of a bias term, we can WLOG consider $c = 0$. Thus, the constraint can be written $\tilde{\mathbb{B}}^\top \tilde{\beta} = VD\tilde{\beta} = \mathbf{0} \iff D\tilde{\beta} = \mathbf{0}$. We can therefore see that this constraint is the same as requiring that the dimensions of $\tilde{\beta}$ corresponding to the non-zero dimensions of $D$ are 0.

Noting that $U^\top \epsilon \sim \mathcal{N}(0, I)$, we now apply Lemma F.2.1 to see that the solution will be $\tilde{\beta} = \alpha(I - DD^\dagger)\tilde{\beta}^*$ for some $\alpha \in (0, 1]$. Finally, as argued above we can recover the solution to the original objective by rotating back, giving the solution $\beta = U\tilde{\beta} = \alpha U(I - DD^\dagger)U^\top \beta^* = \alpha\widehat{\Pi}\beta^*$. □

### F.2.4 Proof of Theorem 7.5.6

**Theorem F.2.2** (Theorem 7.5.6, restated)**.** *For any $\rho \geq 0$, denote the set of possible test environments $\mathcal{A}_\rho$ which contains all parameters $(A_{e'}, b_{e'})$ subject to* **??** *7.5.3***??** *7.5.5 and a bound on the mean: $\|b_{e'}\|_2 \leq \rho$. For logistic or linear regression, let $\widehat{\beta}$ be the minimizer of the corresponding* DARE *objective (7.1) or (F.1). Then,*

$$\sup_{(A_{e'},b_{e'})\in\mathcal{A}_\rho} \mathcal{R}_{e'}(\widehat{\beta}) = (1+\rho^2)(\|\beta^*\|_2^2 + 2B\|\beta^*_{\widehat{\Pi}}\|_2\|\beta^*_{I-\widehat{\Pi}}\|_2). \tag{F.22}$$

*Furthermore, the* DARE *solution is* minimax*:*

$$\widehat{\beta} \in \arg\min_{\beta\in\mathbb{R}^d} \sup_{(A_{e'},b_{e'})\in\mathcal{A}_\rho} \mathcal{R}_{e'}(\beta). \tag{F.23}$$

*Proof.* Recall that in an environment $e$, $\mathbb{E}_e[y \mid x] = \beta^{*T}\Sigma_e^{-1/2}x$. So, for an environment $e'$ and predictor $\widehat{\beta}$, we have the following excess risk decomposition:

$$\mathcal{R}_{e'}(\widehat{\beta}) = \mathbb{E}_{e'}[(\widehat{\beta}^\top\bar{\Sigma}^{-1/2}x' - \beta^{*T}\Sigma_{e'}^{-1/2}x')^2] \tag{F.24}$$

$$= \overbrace{\mathbb{E}_{e'}[(\widehat{\beta}^\top\bar{\Sigma}^{-1/2}(x'-\mu_{e'}) - \beta^{*T}\Sigma_{e'}^{-1/2}(x'-\mu_{e'}))^2]}^{T_1}$$
$$+ \overbrace{\mathbb{E}_{e'}[(\widehat{\beta}^\top\bar{\Sigma}^{-1/2}\mu_{e'} - \beta^{*T}\Sigma_{e'}^{-1/2}\mu_{e'})^2]}^{T_2}. \tag{F.25}$$

Observe that term $T_1$ does not depend on the mean $b_{e'}$.

Term $T_2$ simplifies to

$$\mathbb{E}_{e'}[(\widehat{\beta}^\top\bar{\Sigma}^{-1/2}\mu_{e'} - \beta^{*T}\Sigma_{e'}^{-1/2}\mu_{e'})^2] = \left(\overbrace{(\Sigma_{e'}^{1/2}\bar{\Sigma}^{-1/2}\widehat{\beta} - \beta^*)}^{v}{}^\top b_{e'}\right)^2, \tag{F.26}$$

and so we can write a supremum over $T_2$ as

$$\sup_{\mathcal{A}_\rho} T_2 = \sup_{\mathcal{A}_\rho} (v^\top b_{e'})^2 \tag{F.27}$$

$$= \rho^2 \sup_{\mathcal{A}_\rho} \|v\|_2^2. \tag{F.28}$$

Next, observe that $T_1$ simplifies to

$$\widehat{\beta}^\top \bar{\Sigma}^{-1/2}\Sigma_{e'}\bar{\Sigma}^{-1/2}\widehat{\beta} + \|\beta^*\|_2^2 - 2\widehat{\beta}^\top\bar{\Sigma}^{-1/2}\Sigma_{e'}^{1/2}\beta^* = \|\Sigma_{e'}^{1/2}\bar{\Sigma}^{-1/2}\widehat{\beta} - \beta^*\|_2^2 \tag{F.29}$$

$$= \|v\|_2^2. \tag{F.30}$$

So, returning to the full loss and recalling that $T_1$ is independent of $b_{e'}$, we have

$$\sup_{\mathcal{A}_\rho} \mathcal{R}_{e'}(\widehat{\beta}) = \sup_{\mathcal{A}_\rho} T_1 + T_2 \tag{F.31}$$

$$= (1+\rho^2)\sup_{\mathcal{A}_\rho}\|v\|_2^2. \tag{F.32}$$

Of course, the ideal would be for a given environment $e'$ to set $\widehat{\beta} := \bar{\Sigma}^{1/2}\Sigma_{e'}^{-1/2}\beta^* \implies v = 0$, but we have to choose a single $\widehat{\beta}$ for all possible environments $e'$ parameterized by $(A_{e'}, b_{e'}) \in \mathcal{A}_\rho$. We will show that the choice of $\widehat{\beta} := \alpha\widehat{\Pi}\beta^*$ is minimax-optimal under this set for any $\alpha \in (0,1]$.

Leaving the supremum over adversary choices implicit, we can rewrite the squared norm of $v$ as

$$v^\top v = ((\Delta+I)\widehat{\beta} - \beta^*)^\top((\Delta+I)\widehat{\beta} - \beta^*) \tag{F.33}$$

$$= \widehat{\beta}^\top\Delta^\top\Delta\widehat{\beta} + \|\widehat{\beta} - \beta^*\|_2^2 + 2\widehat{\beta}^\top\Delta^\top(\widehat{\beta} - \beta^*). \tag{F.34}$$

By Lemma F.3.4, we can define an environment by defining the block terms of $U_{\widehat{\Pi}}\Delta U_{\widehat{\Pi}}^\top$ directly. Consider the choice of $\Delta_1 = \lambda\frac{\widehat{\beta}_{I-\widehat{\Pi}}\widehat{\beta}_{I-\widehat{\Pi}}^\top}{\|\widehat{\beta}_{I-\widehat{\Pi}}\|_2^2}$, $\Delta_2 = \Delta_{12} = \Delta_{21} = \mathbf{0}$. This choice is in $\mathcal{A}_\rho$ for any $\lambda \in \mathbb{R}$ since it is block-diagonal and $\|\Delta_2\|_2 = 0$. Now we can write

$$v^\top v = \lambda^2\|\widehat{\beta}_{I-\widehat{\Pi}}\|_2^2 + \|\widehat{\beta} - \beta^*\|_2^2 + 2\lambda\widehat{\beta}_{I-\widehat{\Pi}}^\top(\widehat{\beta}_{I-\widehat{\Pi}} - \beta^*_{I-\widehat{\Pi}}) \tag{F.35}$$

$$\geq \lambda^2\|\widehat{\beta}_{I-\widehat{\Pi}}\|_2^2 - 2\lambda\|\widehat{\beta}_{I-\widehat{\Pi}}\|_2\|\widehat{\beta} - \beta^*\|_2), \tag{F.36}$$

via Cauchy-Schwarz and the triangle inequality. So, taking $\lambda \to \infty$ means that the minimax risk is unbounded unless $\widehat{\beta}_{I-\widehat{\Pi}} = 0 \iff \widehat{\beta} = \widehat{\beta}_{\widehat{\Pi}}$. For the remainder of the proof we consider only this case.

With this restriction, we have

$$\|v\|_2^2 = \|(\Delta+I)\widehat{\beta}_{\widehat{\Pi}} - \beta^*\|_2^2 \tag{F.37}$$

$$= \|(\Delta+I)\widehat{\beta}_{\widehat{\Pi}} - \beta^*_{\widehat{\Pi}} - \beta^*_{I-\widehat{\Pi}}\|_2^2 \tag{F.38}$$

$$= \|(\Delta+I)\widehat{\beta}_{\widehat{\Pi}} - \beta^*_{\widehat{\Pi}}\|_2^2 + \|\beta^*_{I-\widehat{\Pi}}\|_2^2 - 2\beta^{*T}_{I-\widehat{\Pi}}\Delta\widehat{\beta}_{\widehat{\Pi}}. \tag{F.39}$$

Assumption 7.5.3 implies that

$$|\beta^{*T}_{I-\widehat{\Pi}}\Delta\widehat{\beta}_{\widehat{\Pi}}| = |\beta^{*T}U_{\widehat{\Pi}}(I - S_{\widehat{\Pi}})U_{\widehat{\Pi}}^\top \Delta U_{\widehat{\Pi}} S_{\widehat{\Pi}} U_{\widehat{\Pi}}^\top \widehat{\beta}| \tag{F.40}$$

$$= \left|\beta^{*T}U_{\widehat{\Pi}}(I - S_{\widehat{\Pi}}) \begin{bmatrix} \Delta_1 & \Delta_{12} \\ \Delta_{21} & \Delta_2 \end{bmatrix} S_{\widehat{\Pi}} U_{\widehat{\Pi}}^\top \widehat{\beta}\right| \tag{F.41}$$

$$\leq B\|\beta^*_{\widehat{\Pi}}\|_2 \|\beta^*_{I-\widehat{\Pi}}\|_2, \tag{F.42}$$

Consider the DARE solution $\widehat{\beta} = \alpha\beta^*_{\widehat{\Pi}}$ for $\alpha \in (0, 1]$. Then using the equivalent supremized set from Assumption 7.5.5 and Lemma F.3.3, the worst-case risk of this choice is

$$\sup_{\mathcal{A}_\rho} \mathcal{R}_{e'}(\widehat{\beta}) \tag{F.43}$$

$$= (1+\rho^2) \sup_{\|\Delta\beta^*_{\widehat{\Pi}}\|_2^2 < \|\beta^*_{\widehat{\Pi}}\|_2^2} (\|(\alpha\Delta + (\alpha-1)I)\beta^*_{\widehat{\Pi}}\|_2^2 + \|\beta^*_{I-\widehat{\Pi}}\|_2^2$$

$$+ 2B\|\beta^*_{\widehat{\Pi}}\|_2 \|\beta^*_{I-\widehat{\Pi}}\|_2) \tag{F.44}$$

$$= (1+\rho^2)(\|\beta^*_{\widehat{\Pi}}\|_2^2 + \|\beta^*_{I-\widehat{\Pi}}\|_2^2 + 2B\|\beta^*_{\widehat{\Pi}}\|_2 \|\beta^*_{I-\widehat{\Pi}}\|_2). \tag{F.45}$$

It remains to show that any other choice of $\widehat{\beta}$ results in greater risk. Observe that the second two terms of (F.43) are constant with respect to $\Delta$, so we focus on the first term. That is, we show that any other choice results in $\sup_{\|\Delta\beta^*_{\widehat{\Pi}}\|_2^2 < \|\beta^*_{\widehat{\Pi}}\|_2^2} \|(\Delta + I)\widehat{\beta}_{\widehat{\Pi}} - \beta^*_{\widehat{\Pi}}\|_2^2 > \|\beta^*_{\widehat{\Pi}}\|_2^2$ (except for $\widehat{\beta} = \mathbf{0}$, which we address at the end).

For any choice of $\widehat{\beta}_{\widehat{\Pi}}$, decompose the vector into its projection and rejection onto $\beta^*_{\widehat{\Pi}}$ as $\widehat{\beta}_{\widehat{\Pi}} = \alpha\beta^*_{\widehat{\Pi}} + \delta$, with $\delta^\top\beta^*_{\widehat{\Pi}} = 0$. The adversary can choose $\Delta_2 = \lambda\delta\delta^\top$, which lies in $\mathcal{A}_\rho$ for any $\lambda$. Then taking $\lambda \to \infty$ causes risk to grow without bound—it follows that we must have $\delta = 0$.

Next, consider any choice $\alpha \notin (0, 1]$. If $\alpha > 2$ or $\alpha < 0$, choosing $\Delta_2 = 0$ makes this term $(\alpha-1)^2\|\beta^*_{\widehat{\Pi}}\|_2^2 > \|\beta^*_{\widehat{\Pi}}\|_2^2$. If $2 \geq \alpha > 1$, choosing $\Delta_2 = \frac{1}{\alpha}I$ makes this term $\alpha^2\|\beta^*_{\widehat{\Pi}}\|_2^2 > \|\beta^*_{\widehat{\Pi}}\|_2^2$.

Finally, if $\alpha = 0$ then this term is equal to $\|\beta^*_{\widehat{\Pi}}\|_2^2$—however, this value is the *supremum* of the adversarial risk of the DARE solution and it cannot actually be attained. □

### F.2.5 Proof of Theorem 7.5.9

**Theorem F.2.3** (Theorem 7.5.9, restated)**.** *Fix test environment parameters $A_{e'}, b_{e'}$ and our guess $\tilde{\Sigma}$. Suppose we minimize the* DARE *regression objective (F.1) on environments whose means $b_e$ are Gaussian vectors with covariance $\Sigma_b$, with span$(\Sigma_b)$ = span$(I - \Pi)$. After seeing $E$ training domains:*

1. *If $E \geq \operatorname{rank}(\Sigma_b)$ then* DARE *recovers the minimax-optimal predictor almost surely: $\widehat{\beta} = \beta^*_\Pi$.*
2. *Otherwise, if $E \geq r(\Sigma_b)$ then with probability $\geq 1 - \delta$,*

$$\mathcal{R}_{e'}(\widehat{\beta}) \leq \mathcal{R}_{e'}(\beta^*_\Pi) + \mathcal{O}\left( \frac{\|\Sigma_b\|_2}{\xi(\Sigma_b)} \left( \sqrt{\frac{r(\Sigma_b)}{E}} + \max\left\{ \sqrt{\frac{\log 1/\delta}{E}}, \frac{\log 1/\delta}{E} \right\} \right) \right), \tag{F.46}$$

*where $\mathcal{O}(\cdot)$ hides dependence on $\|\Delta\|_2$.*

*Proof.* Define $\widehat{\Pi}_E$ as the projection onto the nullspace of $\mathbb{B}^\top$ after having seen $E$ environments. Item 1 is immediate, since as soon as we observe $E = \operatorname{rank}(\Sigma_b)$ linearly independent $b_e$ we have that span$(\mathbb{B})$ = span$(\Sigma_b)$ and therefore $\widehat{\Pi}_E = \Pi$ (this occurs almost surely for any absolutely continuous distribution). To prove item 2, we will write the solution learned after seeing $E$ environments as $\widehat{\beta}_E := \widehat{\Pi}_E \beta^*$. We can write the excess risk of the ground-truth minimax predictor $\beta^*_\Pi$ as

$$\mathcal{R}_{e'}(\beta^*_\Pi) = \mathbb{E}[(\beta^{*T}_\Pi \bar{\Sigma}^{-1/2} \Sigma_{e'}^{1/2} \epsilon - \beta^{*T}\epsilon)^2] \tag{F.47}$$

$$= \mathbb{E}[(\beta^{*T} \Pi \bar{\Sigma}^{-1/2} \Sigma_{e'}^{1/2} \epsilon - \beta^{*T}\epsilon)^2], \tag{F.48}$$

and likewise we have

$$\mathcal{R}_{e'}(\widehat{\beta}_E) = \mathbb{E}[(\widehat{\beta}_E^\top \bar{\Sigma}^{-1/2} \Sigma_{e'}^{1/2} \epsilon - \beta^{*T}\epsilon)^2] \tag{F.49}$$

$$= \mathbb{E}[(\beta^{*T} \widehat{\Pi}_E \bar{\Sigma}^{-1/2} \Sigma_{e'}^{1/2} \epsilon - \beta^{*T}\epsilon)^2]. \tag{F.50}$$

Taking the difference,

$$\mathcal{R}(\widehat{\beta}_E) - \mathcal{R}(\beta^*_\Pi) = \mathbb{E}[(\beta^{*T} \widehat{\Pi}_E \overbrace{\bar{\Sigma}^{-1/2} \Sigma_{e'}^{1/2} \epsilon}^{:=v} - \beta^{*T}\epsilon)^2 - (\beta^{*T} \Pi \bar{\Sigma}^{-1/2} \Sigma_{e'}^{1/2} \epsilon - \beta^{*T}\epsilon)^2] \tag{F.51}$$

$$\leq \|(\Pi - \widehat{\Pi}_E)\mathbb{E}[vv^\top](2I - \Pi - \widehat{\Pi}_E)\|_2 + 2\|(\Pi - \widehat{\Pi}_E)\mathbb{E}[v\epsilon^\top]\|_2 \tag{F.52}$$

$$\leq 2\left[ \|(\Pi - \widehat{\Pi}_E)\mathbb{E}[vv^\top]\|_2 + \|(\Pi - \widehat{\Pi}_E)\mathbb{E}[v\epsilon^\top]\|_2 \right]. \tag{F.53}$$

Now we note that

$$\mathbb{E}[vv^\top] = \mathbb{E}[(\bar{\Sigma}^{-1/2} \Sigma_{e'}^{1/2} \epsilon)(\bar{\Sigma}^{-1/2} \Sigma_{e'}^{1/2} \epsilon)^\top] \tag{F.54}$$

$$= \bar{\Sigma}^{-1/2} \Sigma_{e'} \bar{\Sigma}^{-1/2} \tag{F.55}$$

and

$$\mathbb{E}[v\epsilon^\top] = \bar{\Sigma}^{-1/2}\Sigma_{e'}^{1/2}. \tag{F.56}$$

These matrices are bounded by $\|\Delta + I\|_2^2, \|\Delta + I\|_2$ respectively and are constant with respect to the training environments we sample. It follows that we can bound the risk difference as

$$\mathcal{R}(\widehat{\beta}_E) - \mathcal{R}(\beta^*_\Pi) \lesssim \|\widehat{\Pi}_E - \Pi\|_2, \tag{F.57}$$

and all that remains is to bound the term $\|\widehat{\Pi}_E - \Pi\|_2$.

Combining Theorems 4 and 5 from Koltchinskii and Lounici [2017] with the triangle inequality, we have that when $r(\Sigma_b) \leq E$, with probability $\geq 1 - \delta$,

$$\|\bar{\Sigma} - \Sigma_b\|_2 \lesssim \|\Sigma_b\|_2 \left( \max\left\{\sqrt{\frac{\log 1/\delta}{E}}, \frac{\log 1/\delta}{E}\right\} + \max\left\{\sqrt{\frac{r(\Sigma_b)}{E}}, \frac{r(\Sigma_b)}{E}\right\}\right). \tag{F.58}$$

Since $r(\Sigma_b) \leq E$, the first term of the second max dominates. Further, Corollary 3 of Yu et al. [2014], a variant of the Davis-Kahan theorem, gives us

$$\|\widehat{\Pi}_E - \Pi\|_2 \lesssim \frac{\|\bar{\Sigma} - \Sigma_b\|_2}{\xi(\Sigma_b)}. \tag{F.59}$$

Combining these facts gives the result. □

### F.2.6 Proof of Theorem 7.5.11

**Theorem F.2.4** (Theorem 7.5.11, fully stated)**.** *Assume the data follows model (7.2) with $\epsilon \sim \mathcal{N}(0, I)$. Observing $n_S$ samples from a source distribution $S$ with covariance $\Sigma_S$, we use half to estimate $\widehat{\Sigma}_S$ and the other half to learn parameters $\beta$ which minimize the unconstrained ($\lambda = 0$),* uncentered DARE *regression objective. At test-time, given $n_T$ samples $\{x_i\}_{i=1}^{n_T}$ from the target distribution $T$ with mean and covariance $\mu_T, \Sigma_T$, we predict $f(x; \beta) = \beta^\top \widehat{\Sigma}_T^{-1/2} x$.*

*Define $m(\Sigma) := \frac{\lambda_{\max}(\Sigma)}{\lambda^3_{\min}(\Sigma)}$, and assume $n_S = \Omega(m(\Sigma_S)d^2)$, $n_T = \Omega(m(\Sigma_T)d^2)$. Then with probability at least $1 - 3d^{-1}$, the excess squared error of our predictor on the new environment is bounded as*

$$\mathcal{R}_T(f) = \mathcal{O}\left( (1 + \|\mu_T\|_2^2) \left( \frac{\mathbb{E}[\eta^2]}{1 - \mathcal{O}\left(\sqrt{\frac{d}{n_S}}\right)} \frac{d}{n_S} + d^2 \left( \frac{m(\Sigma_S)}{n_S} + \frac{m(\Sigma_T)}{n_T} \right) \right) \right). \tag{F.60}$$

*Proof.* We begin by bounding the error of our solution $\|\beta - \beta^*\|_2$. Observe that with our estimate $\widehat{\Sigma}_S$ of the source environment moments, we are solving linear regression with targets $\beta^{*T}\epsilon_i + \eta_i$ and covariates $\widehat{x}_i = \widehat{\Sigma}_S^{-1/2} x_i = \widehat{\Sigma}_S^{-1/2} {\Sigma_S}^{1/2} \epsilon_i$. Thus, if we had access to the true gradient of the modified least-squares objective (which is not the same as assuming $n_S \to \infty$, because in that case we would have $\widehat{\Sigma}_S \to \Sigma_S$), the solution would be

$$\mathbb{E}[\widehat{x}\widehat{x}^\top]^{-1}\mathbb{E}[\widehat{x}y] \tag{F.61}$$

$$= \left(\widehat{\Sigma}_S^{-1/2} {\Sigma_S}^{1/2}(I + \mu_T {\mu_T}^\top){\Sigma_S}^{1/2}\widehat{\Sigma}_S^{-1/2}\right)^{-1} \left(\widehat{\Sigma}_S^{-1/2}{\Sigma_S}^{1/2}(I + \mu_T{\mu_T}^\top)\beta^*\right) \tag{F.62}$$

$$= \widehat{\Sigma}_S^{1/2}{\Sigma_S}^{-1/2}\beta^*. \tag{F.63}$$

We denote this solution to the modified objective as $\widehat{\beta} := \widehat{\Sigma}_S^{1/2}{\Sigma_S}^{-1/2}\beta^*$, and further define $\Delta_S := {\Sigma_S}^{1/2}\widehat{\Sigma}_S^{-1/2}$, with $\Delta_T$ defined analogously. A classical result tells us that the OLS solution is distributed as $\mathcal{N}\left(\widehat{\beta}, \frac{\sigma_y^2}{n_S}M^{-1}\right)$, where $M$ is the modified covariance $\Delta_S^\top \Delta_S$ and $\sigma_y^2 := \mathbb{E}[\eta^2]$. To show a rate of convergence to the OLS solution, we need to solve for the minimum eigenvalue $\lambda_{\min}(M)$—this will suffice since the above fact implies finite-sample convergence of the OLS estimator to the population solution. The well-known bound for sub-Gaussian random vectors tells us that with probability $\geq 1 - \delta_1$,

$$\|\beta - \widehat{\beta}\|_2 \lesssim \sqrt{\lambda_{\max}(M^{-1})\sigma_y^2}\left(\sqrt{\frac{d}{n_S}} + \sqrt{\frac{\log 1/\delta_1}{n_S}}\right). \tag{F.64}$$

and moving forward we condition on this event. Now let $\gamma_S := \sqrt{d/n_S}$, with $\gamma_T$ defined analogously. Since $M \succeq 0$, it follows that

$$\lambda_{\max}(M^{-1}) = \lambda_{\min}(\Delta_S^\top \Delta_S)^{-1}, \tag{F.65}$$

and further,

$$\lambda_{\min}(\Delta_S^\top\Delta_S) \geq 1 - \|\Delta_S^\top\Delta_S - I\|_2. \tag{F.66}$$

By Lemma F.3.1, we have that with probability $\geq 1 - d^{-1}$

$$\|\Delta_S^\top\Delta_S - I\|_2 \lesssim \gamma_S. \tag{F.67}$$

This implies that our solution's error can be bounded as

$$\|\beta - \beta^*\|_2 = \|(\beta - \widehat{\beta}) + (\widehat{\beta} - \beta^*)\|_2 \tag{F.68}$$

$$\leq \sqrt{\frac{\sigma_y^2}{1 - O(\gamma_S)}} \left( \sqrt{\frac{d}{n_S}} + \sqrt{\frac{\log 1/\delta_1}{n_S}} \right) + \|\widehat{\Sigma}_S^{1/2} \Sigma_S{}^{-1/2} - I\|_2 \|\beta^*\|_2 \tag{F.69}$$

We assume $\|\beta^*\|_2 = 1$ so we can avoid carrying it throughout the rest of the proof. Bounding the second term with Lemma F.3.2 gives

$$\|\beta - \beta^*\|_2 \lesssim \sqrt{\frac{\sigma_y^2}{1 - O(\gamma_S))}} \left( \sqrt{\frac{d}{n_S}} + \sqrt{\frac{\log 1/\delta_1}{n_S}} \right) + \gamma_S \sqrt{d\, m(\Sigma_S)}. \tag{F.70}$$

On the target distribution, our excess risk with a predictor $\beta$ is

$$\mathcal{R}_{e'}(\beta) = \mathbb{E}[(\beta^\top \widehat{\Sigma}_T^{-1/2} x - \beta^{*T} \Sigma_T{}^{-1/2} x)^2] \tag{F.71}$$

$$= \mathbb{E}\left[ \left( (\overbrace{\Sigma_T{}^{1/2} \widehat{\Sigma}_T^{-1/2}}^{\Delta_T} \beta - \beta^*)^\top \epsilon \right)^2 \right] \tag{F.72}$$

$$= (\Delta_T \beta - \beta^*)^\top (I + \mu_T \mu_T{}^\top)(\Delta_T \beta - \beta^*) \tag{F.73}$$

$$\leq (1 + \|\mu_T\|_2^2) \|\Delta_T \beta - \beta^*\|_2^2 \tag{F.74}$$

Now, we have

$$\|\Delta_T \beta - \beta^*\|_2 = \|(\Delta_T \beta - \Delta_T \beta^*) + (\Delta_T \beta^* - \beta^*)\|_2 \tag{F.75}$$

$$\leq \|\Delta_T(\beta - \beta^*)\|_2 + \|(\Delta_T - I)\beta^*\|_2 \tag{F.76}$$

$$\leq (1 + \|\Delta_T - I\|_2)\|\beta - \beta^*\|_2 + \|\Delta_T - I\|_2. \tag{F.77}$$

Once again invoking Lemma F.3.2, with probability $\geq 1 - d^{-1}$,

$$\|\Delta_T - I\|_2 \lesssim \gamma_T \sqrt{d\, m(\Sigma_T)}, \tag{F.78}$$

and using this plus the previous result, the triangle inequality, and $(a+b)^2 \leq 2(a^2 + b^2)$ gives

$$\|\Delta_T \beta - \beta^*\|_2^2 \lesssim (1 + \|\Delta_T - I\|_2)^2 \|\beta - \beta^*\|_2^2 + \|\Delta_T - I\|_2^2 \tag{F.79}$$

$$\lesssim (1 + \gamma_T \sqrt{d\, m(\Sigma_T)})^2 \|\beta - \beta^*\|_2^2 + (\|\Sigma^{1/2}\|_2 \|\Sigma^{-1}\|_2^{3/2} \sqrt{d} \gamma_T)^2 \tag{F.80}$$

$$\lesssim \|\beta - \beta^*\|_2^2 + \gamma_T^2 d\, m(\Sigma_T), \tag{F.81}$$

where the lower bound on $n_T$ enforces $\gamma_T\sqrt{d\, m(\Sigma_T)} \leq 1$. It follows that the excess risk can be bounded as

$$\mathcal{R}_{e'} \lesssim (1+\|\mu_T\|_2^2)\|\Delta_T\beta-\beta^*\|_2^2 \tag{F.82}$$
$$\lesssim (1+\|\mu_T\|_2^2)\left(\|\beta-\beta^*\|_2^2+\gamma_T^2 d\, m(\Sigma_T)\right). \tag{F.83}$$

Letting $\delta_1 = 1/d$, combining all of the above via union bound, and eliminating lower-order terms, we get

$$\mathcal{R}_{e'}(\beta) \lesssim (1+\|\mu_T\|_2^2)\left(\frac{\sigma_y^2}{1-\mathcal{O}(\gamma_S)}\frac{d}{n_S}+\gamma_S^2 d\, m(\Sigma_S)+\gamma_T^2 d\, m(\Sigma_T)\right) \tag{F.84}$$
$$= (1+\|\mu_T\|_2^2)\left(\frac{\sigma_y^2}{1-\mathcal{O}\left(\sqrt{\frac{d}{n_S}}\right)}\frac{d}{n_S}+d^2\left(\frac{m(\Sigma_S)}{n_S}+\frac{m(\Sigma_T)}{n_T}\right)\right) \tag{F.85}$$

with probability $\geq 1-3d^{-1}$. □

## F.3 Technical Lemmas

**Lemma F.3.1.** *Suppose we observe $n \in \Omega(d+\log 1/\delta)$ samples $X \sim \mathcal{N}(\mu,\Sigma)$ with $\Sigma \succeq 0$ and estimate the inverse covariance matrix $\Sigma^{-1}$ with the inverse of the sample covariance matrix $\bar{\Sigma}^{-1}$. Then with probability $\geq 1-\delta$, it holds that*

$$\|\bar{\Sigma}^{-1/2}\Sigma\bar{\Sigma}^{-1/2}-I\|_2 \lesssim \sqrt{\frac{d+\log 1/\delta}{n}}. \tag{F.86}$$

*Proof.* As $\bar{\Sigma}^{-1/2}\Sigma\bar{\Sigma}^{-1/2}$ and $\Sigma^{1/2}\bar{\Sigma}^{-1}\Sigma^{1/2}$ have the same spectrum, it suffices to bound the latter. Observe that

$$\|\Sigma^{1/2}\bar{\Sigma}^{-1}\Sigma^{1/2}-I\|_2 = \|\Sigma^{1/2}(\bar{\Sigma}^{-1}-\Sigma^{-1})\Sigma^{1/2}\|_2. \tag{F.87}$$

Now applying Theorem 10 of Kereta and Klock [2021] with $A = B = \Sigma^{1/2}$ yields the result. □

**Lemma F.3.2.** *Assume the conditions of Lemma F.3.1. Then under the same event as Lemma F.3.1, it holds that*

$$\|\Sigma^{1/2}\bar{\Sigma}^{-1/2}-I\|_2 \lesssim \|\Sigma^{1/2}\|_2\left(\|\Sigma^{-1}\|_2^{3/2}\sqrt{d}\gamma+\|\Sigma^{-1}\|_2^2\gamma^2\right), \tag{F.88}$$

*where $\gamma=\sqrt{\frac{d+\log 1/\delta}{n}}$. In particular, if $\delta=d^{-1}$, then*

$$\|\Sigma^{1/2}\bar{\Sigma}^{-1/2}-I\|_2 \lesssim \sqrt{\|\Sigma\|_2\|\Sigma^{-1}\|_2^3\frac{d^2}{n}} \tag{F.89}$$

*Proof.* We begin by deriving a bound for $\|\bar{\Sigma}^{-1/2} - \Sigma^{-1/2}\|_2$. Define $E = \bar{\Sigma}^{-1} - \Sigma^{-1}$. Observe that

$$\begin{aligned}\|E\|_2 &= \|\Sigma^{-1/2}\Sigma^{1/2}E\Sigma^{1/2}\Sigma^{-1/2}\|_2 && \text{(F.90)}\\ &= \|\Sigma^{-1/2}(\Sigma^{1/2}\bar{\Sigma}^{-1}\Sigma^{1/2} - I)\Sigma^{-1/2}\|_2 && \text{(F.91)}\\ &\le \|\Sigma^{-1}\|_2\|\Sigma^{1/2}\bar{\Sigma}^{-1}\Sigma^{1/2} - I\|_2, && \text{(F.92)}\end{aligned}$$

and now apply Lemma F.3.1 (since $\bar{\Sigma}^{-1/2}\Sigma\bar{\Sigma}^{-1/2}$ and $\Sigma^{1/2}\bar{\Sigma}^{-1}\Sigma^{1/2}$ have the same spectrum), giving

$$\|E\|_2 \lesssim \|\Sigma^{-1}\|_2\gamma. \tag{F.93}$$

Let $UDU^\top$ be the eigendecomposition of $\Sigma$, and define the matrix $[\sqrt{\cdot}, \alpha]_{i,j} = \frac{1}{\sqrt{D_{ii}}+\sqrt{D_{jj}}}$ as in Carlsson [2018]. The Daleckii-Krein theorem [Daleckii and Krein, 1965] tells us that

$$\|\bar{\Sigma}^{-1/2} - \Sigma^{-1/2}\|_2 = \|U([\sqrt{\cdot}, \alpha] \circ E)U^\top\|_2 + O(\|E\|_2^2). \tag{F.94}$$

Note that $\max_{i,j} |[\sqrt{\cdot}, \alpha]_{i,j}| = 1/2\sqrt{\lambda_{\min}(\Sigma)} \implies \|[\sqrt{\cdot}, \alpha]\|_2 \le \sqrt{d\|\Sigma^{-1}\|_2}$, and therefore by sub-multiplicativity of spectral norm under Hadamard product,

$$\begin{aligned}\|\bar{\Sigma}^{-1/2} - \Sigma^{-1/2}\|_2 &\lesssim \sqrt{d\|\Sigma^{-1}\|_2}\|E\|_2 + \|E\|_2^2 && \text{(F.95)}\\ &\lesssim \|\Sigma^{-1}\|_2^{3/2}\sqrt{d}\gamma + \|\Sigma^{-1}\|_2^2\gamma^2. && \text{(F.96)}\end{aligned}$$

Finally, noting that

$$\begin{aligned}\|\Sigma^{1/2}\bar{\Sigma}^{-1/2} - I\|_2 &= \|\Sigma^{1/2}(\bar{\Sigma}^{-1/2} - \Sigma^{-1/2})\|_2 && \text{(F.97)}\\ &\le \|\Sigma^{1/2}\|_2\|\bar{\Sigma}^{-1/2} - \Sigma^{-1/2}\|_2 && \text{(F.98)}\end{aligned}$$

completes the main proof. To see the second claim, note that $n \ge 2\|\Sigma^{-1}\|_2$ implies $\|\Sigma^{-1}\|_2^{3/2}\sqrt{d} \ge \|\Sigma^{-1}\|_2^2\gamma$, meaning the first of the two terms dominates. □

**Lemma F.3.3.** *Let $\bar{\Sigma}$ be fixed. Then Assumption 7.5.5 is satisfied if and only if $\|\Delta\beta^*_{\widehat{\Pi}}\|_2^2 < \|\beta^*_{\widehat{\Pi}}\|_2^2$.*

*Proof.* The claim follows by rewriting the risk terms. Recall that $\Delta = \Sigma_{e'}^{1/2}\bar{\Sigma}^{-1/2} - I$. Writing out the excess risk of the ground truth $\beta^*$ in the subspace $\widehat{\Pi}$,

$$\begin{aligned}\mathcal{R}^{\widehat{\Pi}}_{p_{e'}}(\beta^*) &= \mathbb{E}_{p_{e'}}\left[(\beta^{*T}_{\widehat{\Pi}}\bar{\Sigma}^{-1/2}x - \beta^{*T}_{\widehat{\Pi}}\Sigma_{e'}^{-1/2}x)^2\right] && \text{(F.99)}\\ &= \mathbb{E}_{p_{e'}}\left[(\beta^{*T}\widehat{\Pi}\bar{\Sigma}^{-1/2}\Sigma_{e'}^{1/2}\epsilon - \beta^{*T}\widehat{\Pi}\epsilon)^2\right] && \text{(F.100)}\\ &= \mathbb{E}_{p_{e'}}\left[(\beta^{*T}\widehat{\Pi}\Delta^\top\epsilon)^2\right] && \text{(F.101)}\\ &= \|\Delta\widehat{\Pi}\beta^*\|_2^2. && \text{(F.102)}\end{aligned}$$

Next, the excess risk of the null vector $\widehat{\beta} = \mathbf{0}$ in the same subspace is

$$\begin{aligned}\mathcal{R}^{\widehat{\Pi}}_{p_{e'}}(\mathbf{0}) &= \mathbb{E}_{p_{e'}}[(\beta^{*T}_{\widehat{\Pi}} \Sigma_{e'}^{-1/2} x)^2] && \text{(F.103)}\\ &= \mathbb{E}_{p_{e'}}[(\beta^{*T} \widehat{\Pi} \epsilon)^2] && \text{(F.104)}\\ &= \|\widehat{\Pi}\beta^*\|_2^2. && \square\end{aligned}$$

**Lemma F.3.4.** *For a fixed $\bar{\Sigma}$, choosing an environmental covariance $\Sigma_{e'}$ is equivalent to directly choosing the error terms $\Delta_1, \Delta_2, \Delta_{12}, \Delta_{21}$.*

*Proof.* For a fixed $\bar{\Sigma}$, due to the unique definition of square root as a result of Assumption 7.5.1, the map $\Sigma_{e'} \to \Sigma_{e'}^{1/2} \bar{\Sigma}^{-1/2} - I$ is one-to-one. Recall that we can write

$$U_{\widehat{\Pi}}^\top \Delta U_{\widehat{\Pi}} = \begin{bmatrix} \Delta_1 & \Delta_{12} \\ \Delta_{21} & \Delta_2 \end{bmatrix} \tag{F.105}$$

which defines a one-to-one map from $\Delta$ to the error terms. Since the composition of bijective functions is bijective, the claim follows. $\square$

## F.4 Implementation Details

### F.4.1 Evaluation Pipeline

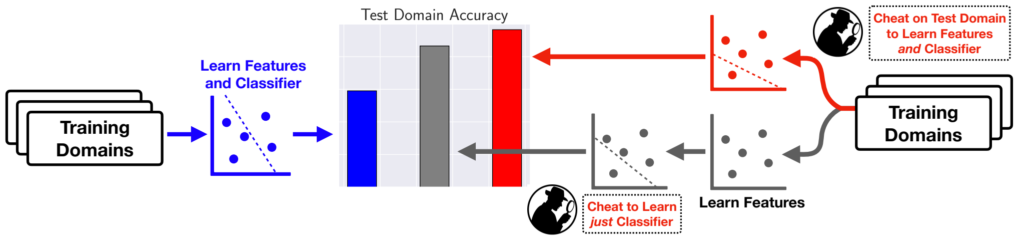

Test Domain Accuracy
Training Domains
Learn Features and Classifier
Cheat on Test Domain to Learn Features *and* Classifier
Training Domains
Learn Features
Cheat to Learn *just* Classifier

Figure F.1: Depiction of evaluation pipeline. Standard training on train domains leads to poor performance. Cheating while training the full network (features and classifier) does substantially better. However, cheating on *just* the linear classifier does almost as well, implying that the features learned without cheating are already good enough for massive improvements over SOTA.

Figure F.1 depicts the overall pipeline for evaluation of the different approaches we describe in Section 7.2. Our baselines are three distinct pipelines, each slightly different:

1. The first pipeline (dark blue in Figure F.1) is the standard method of evaluating ERM on a new domain. We train the entire network (the combination of a feature embedder comprising all layers except the last linear layer, and the last layer linear classifier) end-to-end on the training domains, simultaneously learning the features and the linear classifier on the training domains via backpropagation. When evaluated on a new domain, this achieves state-of-the-art accuracy [Gulrajani and Lopez-Paz, 2021], but it still performs quite poorly.
2. The second pipeline (red) is very similar to the first, but we include the test domain among the domains on which the network is trained end-to-end. Because this means that the test distribution is one of the training domains, this simulates an "ideal" setting where no distribution shift occurs from train-time to test-time. As such, it is unsurprising that this approach performs substantially better (though it still leaves a bit to be desired—this raises a separate question about failures of in-distribution learning with multiple domains). As this approach requires cheating, it is completely unrealistic to expect current methods to even begin to approach this baseline, but it gives a good sense of what would be the best performance to hope for.
3. The final pipeline (grey) is our main experimental contribution. Here, the *features* (all but the last layer) are learned without cheating, as in the first pipeline. Next, we freeze the features and cheat only while learning a linear classifier on top of these features. Not only does this method do significantly better than the first baseline, it actually performs almost as well as—and sometimes better than—the second pipeline. Thus, we find that standard features learned via ERM without cheating are *already good enough for generalization* and that the main bottleneck to reaching the accuracy of the second baseline is in learning a good linear classifier *only*. This has several important advantages to current methods which attempt to change the entire end-to-end process, which we explicate in Section 7.2 in the main body.

### F.4.2 Code details

All features were learned by finetuning a ResNet-50 using the default settings and hyperparameter sweeps of the DomainBed benchmark [Gulrajani and Lopez-Paz, 2021]. We extracted features from 3 trials, with 5 random hyperparameter choices per trial, picking the one with the best training domain validation accuracy. We used the default random splits of 80% train / 20% test for each domain.

Using frozen features, the cheating linear classifier was trained by minimizing the multinomial logistic loss with full-batch SGD with momentum for 3000 steps.

We did not use a validation set.

For the main experiments, all algorithms were trained using full-batch L-BFGS [Liu and Nocedal, 1989]. We used the exact same optimization hyperparameters for all methods; the default learning rate of 1 was unstable when optimizing IRM, frequently diverging, so we lowered the learning rate until IRM consistently converged (which occurred at a learning rate of 0.2). Since the IRM penalty only makes sense with both a feature embedder *and* a classification vector, we used an additional linear layer for IRM, making the objective non-convex. Presumably due to their convexity for linear classifiers, all other methods were unaffected by this change. For all methods, we halted optimization once 20 epochs occurred with no increase in training domain validation accuracy; the maximum validation accuracy typically occurred within the first 5 epochs.

For stability when whitening (and because the number of samples per domain was often less than the feature dimension), in estimating $\widehat{\Sigma}_e$ for each environment we shrank the sample covariance towards the identity. Specifically, we define $\widehat{\Sigma}_e = (1-\rho)\frac{1}{n}\sum_{i=1}^n x_i x_i^\top + \rho I$, with $\rho = 0.1$, and $\rho = 0.01$ for DomainNet due to its much larger size. We found that increasing $\lambda$ beyond $\sim$1 had little-to-no effect on the accuracy, loss, *or* penalty of the DARE solution (see Appendix F.5 for ablations). However, we did observe that choosing a very large value for $\lambda$ (e.g., $10^5$ or higher) could result in poor conditioning of the objective, with the result being that the L-BFGS optimizer took several epochs for the loss to begin going down.

## F.5 Additional Experiments

Here we present additional experimental results. All reported accuracies represent the mean of three trials, and all error bars (where present) display 90% confidence intervals calculated as $\pm 1.645\frac{\widehat{\sigma}}{\sqrt{n}}$. Note though that **the results are not independent across methods, so simply checking if the error bars overlap is overly conservative in identifying methods which perform better.**

### F.5.1 Accuracy of the Test Covariance Whitener

As we do not have access to the true test-time covariance, we estimate the adjustment with the average of the train-time adjustments. Table F.1 reports the normalized squared Frobenius norm of our estimate's error to the sample covariance adjustment, defined as $\frac{\|\widehat{W}-W\|_F^2}{\|W\|_F^2}$, where $W := \widehat{\Sigma}_{e'}^{-1/2}$ is the sample covariance adjustment and $\widehat{W} := \frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}} \Sigma_e^{-1/2}$ is our averaging estimate. We find that this normalized error is consistently small, meaning our estimated adjustment is reasonably close to the "true" adjustment, relative to its spectrum (we put "true" in quotation marks because the best we can do is estimate it via the sample covariance). This helps explain why our averaging adjustment performs so well in practice, though we expect future methods could improve on this estimate (particularly on the last domain of PACS).

| **Dataset** | **Normalized Error** |
|---|---|
| **Office-Home** | |
| A | 0.033 |
| C | 0.040 |
| P | 0.022 |
| R | 0.019 |
| **PACS** | |
| A | 0.052 |
| C | 0.016 |
| P | 0.025 |
| S | 0.217 |
| **VLCS** | |
| C | 0.094 |
| L | 0.039 |
| S | 0.051 |
| V | 0.029 |

Table F.1: Mean normalized adjustment estimation error for each dataset and each train-domain/test-domain split.

### F.5.2 Alignment of Domain-Specific Optimal Classifiers

As discussed in Section 7.3, DARE does not project out varying subspaces but rather aligns them such that the adjusted domains have similar optimal classifiers simultaneously. To verify that this is actually happening, we learn the optimal linear classifier for each domain individually and evaluate the inter-domain cosine similarity for these vectors for each class. We see that without adjustment, different domains' optimal linear decision boundaries have normals with small alignment on average, which explains why trying to learn a single linear classifier which does well on all domains simultaneously performs poorly in most cases. After alignment, the individually optimal classifiers are more aligned, which allows a single classifier to perform better on all domains. Furthermore, this is done without throwing away the varying component (as would be done by invariant prediction [Peters et al., 2016]), allowing DARE to use more information and resulting in higher test accuracy.

| Dataset | With Adjustement | Without Adjustement |
|---|---|---|
| **Office-Home** | | |
| A | 0.693 | 0.596 |
| C | 0.710 | 0.691 |
| P | 0.695 | 0.636 |
| R | 0.642 | 0.590 |
| **PACS** | | |
| A | 0.903 | 0.863 |
| C | 0.896 | 0.776 |
| P | 0.905 | 0.827 |
| S | 0.903 | 0.791 |
| **VLCS** | | |
| C | 0.598 | 0.200 |
| L | 0.723 | 0.436 |
| S | 0.674 | 0.231 |
| V | 0.591 | 0.210 |

Table F.2: Mean cosine similarity between linear classification vectors which are individually optimal for their respective domains (learned via logistic regression). For each dataset and each train-domain/test-domain split we report the average similarity across all class normal vectors and all domain pairs.

### F.5.3 Ablations

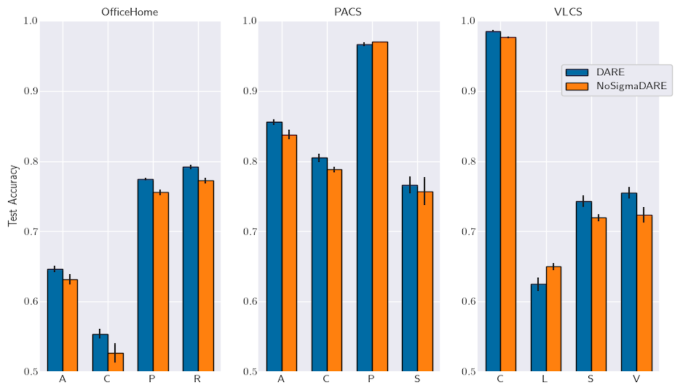


Figure F.2: Demonstration of the effect of whitening. NoSigmaDARE is the exact same algorithm as DARE but with no covariance whitening. In almost all cases, covariance whitening + guessing at test-time results in better performance. We expect under much larger distribution shift that this pattern may reverse.

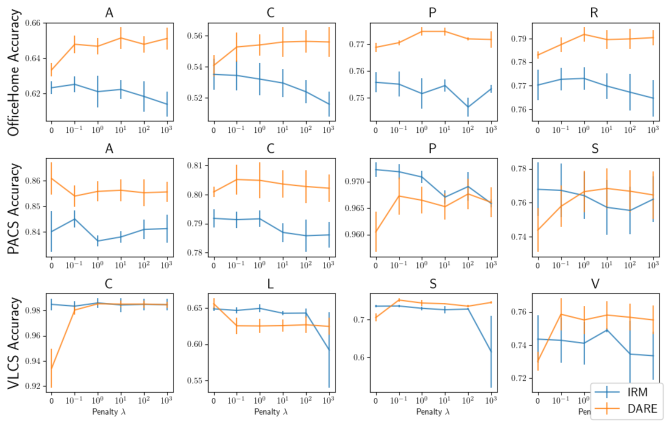


Figure F.3: Effect of penalty term $\lambda$ on the two algorithms which use it. $\lambda = 0$ corresponds to no constraint, and the lower performance demonstrates that this invariance requirement is essential to the quality of the learned classifier. For $\lambda \neq 0$, DARE accuracy is extremely robust, effectively constant for all $\lambda \geq 1$; in practice we also found the penalty term itself to always be $\sim 0$. In contrast, IRM accuracy appears to *decrease* with increasing $\lambda$, implying that the observed benefit of IRM primarily comes from the domain reweighting as in our Reweighted ERM method.

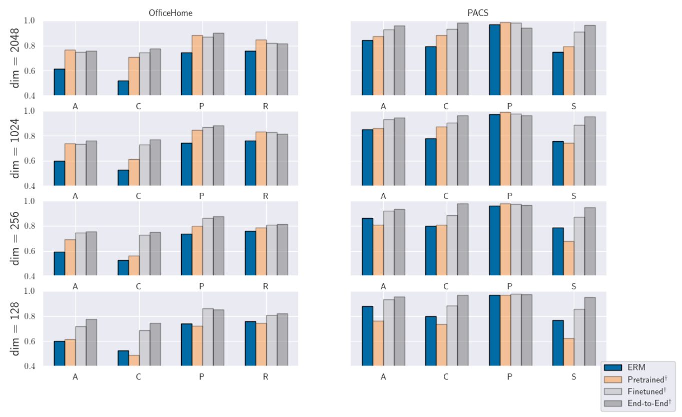


Figure F.4: Effect of final feature bottleneck dimensionality on cheating accuracy. Reducing the dimensionality reduces accuracy of all methods to varying degrees, though in some cases it actually *increases* test accuracy. We observe that the main pattern persists, though the gap between cheating on finetuned features and traditional ERM shrinks as the dimensionality is reduced substantially. To reduce dimensionality of the pretrained features we use a random projection; unsurprisingly, the quality dramatically falls as the number of dimensions is reduced.

# Appendix G

# Appendix for Chapter 8

## G.1 Experimental Details

### G.1.1 Description of Baselines

*Average Thresholded Confidence (ATC).* ATC first estimates a threshold $t$ on the confidence of softmax prediction (or on negative entropy) such that the number of source labeled points that get a confidence greater than $t$ match the fraction of correct examples, and then estimates the test error on on the target domain $\mathcal{D}_{\text{test}}$ as the expected number of target points that obtain a score less than $t$, i.e.,

$$\text{ATC}_{\mathcal{D}_{\text{test}}}(s) = \sum_{i=1}^{n} \mathbb{I}\left[s(f(x_i')) < t\right] \,, \tag{G.1}$$

where $t$ satisfies: $\sum_{i=1}^{j} \mathbb{I}\left[\max_{j\in\mathcal{Y}}(f_j(x_i)) < t\right] = \sum_{i=1}^{m} \mathbb{I}\left[\arg\max_{j\in\mathcal{Y}} f_j(x_i) \neq y_i\right]$

*Average Confidence (AC).* Error is estimated as the average value of the maximum softmax confidence on the target data, i.e, $\text{AC}_{\mathcal{D}_{\text{test}}} = \sum_{i=1}^{n} \max_{j\in\mathcal{Y}} f_j(x_i')$.

*Difference Of Confidence (DOC).* We estimate error on the target by subtracting the difference of confidences on source and target (as a surrogate to distributional distance [Guillory et al., 2021]) from the error on source distribution, i.e., $\text{DOC}_{\mathcal{D}_{\text{test}}} = \sum_{i=1}^{n} \max_{j\in\mathcal{Y}} f_j(x_i') + \sum_{i=1}^{m} \mathbb{I}\left[\arg\max_{j\in\mathcal{Y}} f_j(x_i) \neq y_i\right] - \sum_{i=1}^{m} \max_{j\in\mathcal{Y}} f_j(x_i)$. This is referred to as DOC-Feat in [Guillory et al., 2021].

*Confidence Optimal Transport (COT).* COT uses the empirical estimator of the Earth Mover's Distance between labels from the source domain and softmax outputs of samples from the target domain to provide accuracy estimates:

$$\mathrm{COT}_{\mathcal{D}_{\text{test}}}(s) = \frac{1}{2} \min_{\pi \in \Pi(S^n, Y^m)} \sum_{i,j=1}^{n,m} \|s_i - e_{y_j}\| 2\pi_{ij} \,, \tag{G.2}$$

where $S^n = \{f(x_i')\}_{i=1}^n$ are the softmax outputs on the unlabeled target data and $Y^m = \{y_j\}_{j=1}^m$ are the labels on holdout source examples.

For all of the methods described above, we assume that $\{(x_i')\}_{i=1}^n$ are the unlabeled target samples and $\{(x_i, y_i)\}_{i=1}^m$ are hold-out labeled source samples.

### G.1.2 Dataset Details

In this section, we provide additional details about the datasets used in our benchmark study.

- **CIFAR10** We use the original CIFAR10 dataset [Krizhevsky and Hinton, 2009] as the source dataset. For target domains, we consider (i) synthetic shifts (CIFAR10-C) due to common corruptions [Hendrycks and Dietterich, 2019]; and (ii) natural distribution shift, i.e., CIFAR10v2 [Recht et al., 2018a, Torralba et al., 2008] due to differences in data collection strategy. We randomly sample 3 set of CIFAR-10-C datasets. Overall, we obtain 5 datasets (i.e., CIFAR10v1, CIFAR10v2, CIFAR10C-Frost (severity 4), CIFAR10C-Pixelate (severity 5), CIFAR10-C Saturate (severity 5)).
- **CIFAR100** Similar to CIFAR10, we use the original CIFAR100 set as the source dataset. For target domains we consider synthetic shifts (CIFAR100-C) due to common corruptions. We sample 4 CIFAR100-C datasets, overall obtaining 5 domains (i.e., CIFAR100, CIFAR100C-Fog (severity 4), CIFAR100C-Motion Blur (severity 2), CIFAR100C-Contrast (severity 4), CIFAR100C-spatter (severity 2) ).
- **FMoW** In order to consider distribution shifts faced in the wild, we consider FMoW-WILDs [Koh et al., 2021, Christie et al., 2018] from WILDS benchmark, which contains satellite images taken in different geographical regions and at different times. We use the original train as source and OOD val and OOD test splits as target domains as they are collected over different time-period. Overall, we obtain 3 different domains.
- **Camelyon17** Similar to FMoW, we consider tumor identification dataset from the wilds benchmark [Bandi et al., 2018]. We use the default train as source and OOD val and OOD test splits as target domains as they are collected across different hospitals. Overall, we obtain 3 different domains.

- **BREEDs** We also consider BREEDs benchmark [Santurkar et al., 2020] in our setup to assess robustness to subpopulation shifts. BREEDs leverage class hierarchy in ImageNet to re-purpose original classes to be the subpopulations and defines a classification task on superclasses. We consider distribution shift due to subpopulation shift which is induced by directly making the subpopulations present in the training and test distributions disjoint. BREEDs benchmark contains 4 datasets **Entity-13**, **Entity-30**, **Living-17**, and **Non-living-26**, each focusing on different subtrees and levels in the hierarchy. We also consider natural shifts due to differences in the data collection process of ImageNet [Russakovsky et al., 2015], e.g, ImageNetv2 [Recht et al., 2019] and a combination of both. Overall, for each of the 4 BREEDs datasets (i.e., Entity-13, Entity-30, Living-17, and Non-living-26), we obtain four different domains. We refer to them as follows: BREEDsv1 sub-population 1 (sampled from ImageNetv1), BREEDsv1 sub-population 2 (sampled from ImageNetv1), BREEDsv2 sub-population 1 (sampled from ImageNetv2), BREEDsv2 sub-population 2 (sampled from ImageNetv2). For each BREEDs dataset, we use BREEDsv1 sub-population A as source and the other three as target domains.
- **OfficeHome** We use four domains (art, clipart, product and real) from OfficeHome dataset [Venkateswara et al., 2017b]. We use the product domain as source and the other domains as target.
- **DomainNet** We use four domains (clipart, painting, real, sketch) from the Domainnet dataset [Peng et al., 2019b]. We use real domain as the source and the other domains as target.
- **Visda** We use three domains (train, val and test) from the Visda dataset [Peng et al., 2018]. While 'train' domain contains synthetic renditions of the objects, 'val' and 'test' domains contain real world images. To avoid confusing, the domain names with their roles as splits, we rename them as 'synthetic', 'Real-1' and 'Real-2'. We use the synthetic (original train set) as the source domain and use the other domains as target.

### G.1.3 Setup and Protocols

**Architecture Details** For all datasets, we used the same architecture across different algorithms:

- CIFAR-10: Resnet-18 [He et al., 2016] pretrained on Imagenet
- CIFAR-100: Resnet-18 [He et al., 2016] pretrained on Imagenet
- Camelyon: Densenet-121 [Huang et al., 2017a] *not* pretrained on Imagenet as per the suggestion made in [Koh et al., 2021]

- FMoW: Densenet-121 [Huang et al., 2017a] pretrained on Imagenet
- BREEDs (Entity13, Entity30, Living17, Nonliving26): Resnet-18 [He et al., 2016] *not* pretrained on Imagenet as per the suggestion in [Santurkar et al., 2020]. The main rationale is to avoid pre-training on the superset dataset where we are simulating sub-population shift.
- Officehome: Resnet-50 [He et al., 2016] pretrained on Imagenet
- Domainnet: Resnet-50 [He et al., 2016] pretrained on Imagenet
- Visda: Resnet-50 [He et al., 2016] pretrained on Imagenet

Except for Resnets on CIFAR datasets, we used the standard pytorch implementation [Gardner et al., 2018]. For Resnet on cifar, we refer to the implementation here: `https://github.com/kuangliu/pytorch-cifar`. For all the architectures, whenever applicable, we add antialiasing [Zhang, 2019]. We use the official library released with the paper.

For imagenet-pretrained models with standard architectures, we use the publicly available models here: `https://pytorch.org/vision/stable/models.html`. For imagenet-pretrained models on the reduced input size images (e.g. CIFAR-10), we train a model on Imagenet on reduced input size from scratch. We include the model with our publicly available repository.

**Hyperparameter details** First, we tune learning rate and $\ell_2$ regularization parameter by fixing batch size for each dataset that correspond to maximum we can fit to 15GB GPU memory. We set the number of epochs for training as per the suggestions of the authors of respective benchmarks. Note that we define the number of epochs as a full pass over the labeled training source data. We summarize learning rate, batch size, number of epochs, and $\ell_2$ regularization parameter used in our study in Table G.2.

For each algorithm, we use the hyperparameters reported in the initial papers. For domain-adversarial methods (DANN and CDANN), we refer to the suggestions made in Transfer Learning Library [Jiang et al., 2022a]. We tabulate hyperparameters for each algorithm next:

- **DANN, CDANN,** As per Transfer Learning Library suggestion, we use a learning rate multiplier of $0.1$ for the featurizer when initializing with a pre-trained network and $1.0$ otherwise. We default to a penalty weight of $1.0$ for all datasets with pre-trained initialization.
- **FixMatch** We use the lambda is $1.0$ and use threshold $\tau$ as $0.9$.

**Compute Infrastructure** Our experiments were performed across a combination of Nvidia T4, A6000, and V100 GPUs.

## G.2 Comparing Disagreement Losses

We define the alternate losses for maximizing disagreement:

1. Chuang et al. [2020] minimize the negative cross-entropy loss, which is concave in the model logits. That is, they add the term $\log \operatorname{softmax}(h(x)_y)$ to the objective they are minimizing. This loss results in substantially lower disagreement discrepancy than the other two.
2. Pagliardini et al. [2023] use a loss which is not too different from ours. They define the disagreement objective for a point $(x, y)$ as

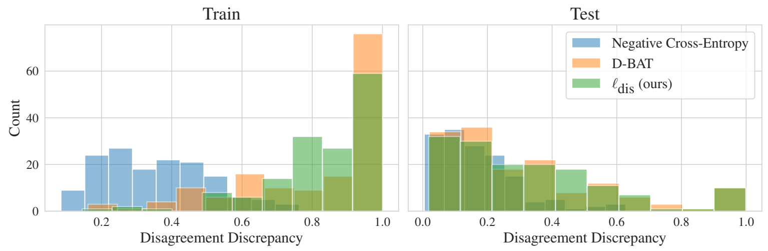
$$\log\left(1 + \frac{\exp(h(x)_y)}{\sum_{\widehat{y}\neq y}\exp(h(x)_{\widehat{y}})}\right). \tag{G.3}$$

For comparison, $\ell_{\text{dis}}$ can be rewritten as

$$\log\left(1 + \frac{\exp(h(x)_y)}{\exp\left(\frac{1}{|\mathcal{Y}|-1}\sum_{\widehat{y}\neq y} h(x)_{\widehat{y}}\right)}\right), \tag{G.4}$$

where the incorrect logits are averaged and the exponential is pushed outside the sum. This modification results in (G.4) being convex in the logits and an upper bound to the disagreement 0-1 loss, whereas (G.3) is neither.

Figure G.1 displays histograms of the achieved disagreement discrepancy across all distributions for each of the disagreement losses (all hyperparameters and random seeds are the same for all three losses). The table below it reports the mean disagreement discrepancy on the train and test sets. We find that the negative cross-entropy, being a concave function, results in very low discrepancy. The loss (G.3) is reasonably competitive with our loss (G.4) on average, seemingly because it gets very high discrepancy on a subset of shifts. This suggests that it may be particularly suited for a specific type of distribution shift, though it is less good overall. Though the averages are reasonably close, the samples are not independent, so we run a paired t-test and we find that the increases to average train and test discrepancies achieved by $\ell_{\text{dis}}$ are significant at levels $p = 0.024$ and $p = 0.009$, respectively. However, with enough holdout data, a reasonable approach would be to split the data in two: one subset to validate critics trained on either of the two losses, and another to evaluate the discrepancy of whichever one is ultimately selected.



| Loss | Mean Discr. (Train) | Mean Discr. (Test) |
|---|---|---|
| Neg. X-Ent [Chuang et al., 2020] | $0.3555 \pm .0124$ | $0.1694 \pm .0105$ |
| D-BAT [Pagliardini et al., 2023] | $0.8145 \pm .0177$ | $0.3224 \pm .0212$ |
| $\ell_{\text{dis}}$ (Ours) | $\underline{0.8333 \pm .0132}$ | $\mathbf{0.3322 \pm .0205}$ |

Figure G.1 & Table G.2: Histogram of disagreement discrepancies for each of the three losses, and the average values across all datasets. **Bold** (resp. Underline) indicates the method has higher average discrepancy under a paired t-test at significance $p = .01$ (resp. $p = .05$).

## G.3 Exploration of the Validity Score

To experiment with reducing the complexity of the class $\mathcal{H}$, we evaluate $\textsc{Dis}^2$ on progressively fewer top principal components (PCs) of the features. Precisely, for features of dimension $d$, we evaluate $\textsc{Dis}^2$ on the same features projected onto their top $d/k$ components, for $k \in [1, 4, 16, 32, 64, 128]$ (Figure G.2). We see that while projecting to fewer and fewer PCs does reduce the error bound value, unlike the logits it is a rather crude way to reduce complexity of $\mathcal{H}$, meaning at some point it goes too far and results in invalid error bounds.

However, during the optimization process we observe that around when this violation occurs, the task of training a critic to both agree on $\mathcal{S}$ and disagree on $\mathcal{T}$ goes from "easy" to "hard". Figure G.3 shows that on the full features, the critic rapidly ascends to maximum agreement on $\mathcal{S}$, followed by slow decay (due to both overfitting and learning to simultaneously disagree on $\mathcal{T}$). As we drop more and more components, this optimization becomes slower.

We therefore design a "validity score" intended to capture this phenomenon which we refer to as the *cumulative* $\ell_1$ *ratio*. This is defined as the maximum agreement achieved, divided by the cumulative sum of absolute differences in agreement across all epochs up until the maximum was achieved. Formally, let $\{a_i\}_{i=1}^T$ represent the agreement between $h'$ and $\widehat{h}$ after epoch $i$, i.e. $1 - \epsilon_{\widehat{\mathcal{S}}}(\widehat{h}, h'_i)$, and define $m := \arg\max_{i \in [T]} a_i$. The cumulative $\ell_1$ ratio is then $\frac{a_m}{a_1 + \sum_{i=2}^m |a_i - a_{i-1}|}$.

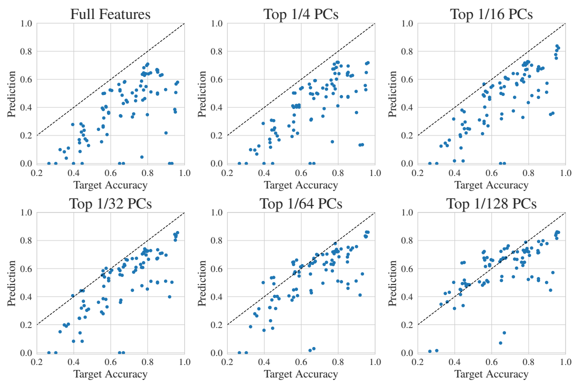


Figure G.2: **DIS$^2$ bound as fewer principal components are kept.** Reducing the number of top principal components crudely reduces complexity of $\mathcal{H}$—this leads to lower error estimates, but at some point the bounds become invalid for a large fraction of shifts.

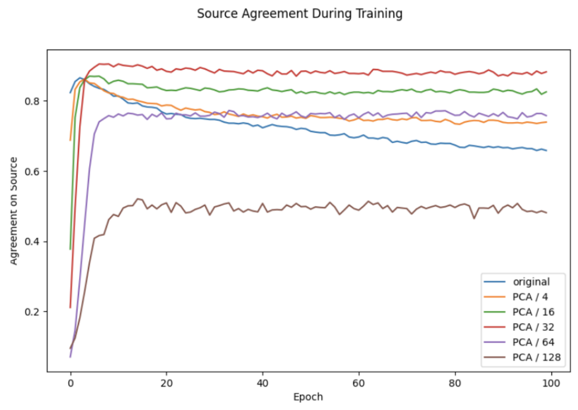


Figure G.3: **Agreement on one shift between $\widehat{h}$ and $h'$ on $\widehat{\mathcal{S}}$ during optimization.** We observe that as the number of top PCs retained drops, the optimization occurs more slowly and less monotonically. For this particular shift, the bound becomes invalid when keeping only the top $1/128$ components, depicted by the brown line.

Thus, if the agreement rapidly ascends to its maximum without ever going down over the course of an epoch, this ratio will be equal to 1, and if it non-monotonically ascends then the ratio will be significantly less. This definition was simply the first metric we considered which approximately captures the behavior we observed; we expect it could be greatly improved.

Figure G.4 displays a scatter plot of the cumulative $\ell_1$ ratio versus the difference

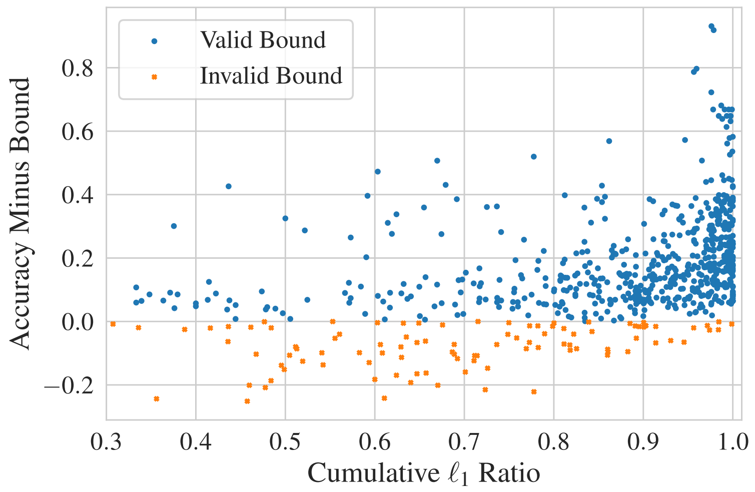


Figure G.4: **Cumulative $\ell_1$ ratio versus error prediction gap.** Despite its simplicity, the ratio captures the information encoded in the optimization trajectory, roughly linearly correlating with the tightness and validity of a given prediction. It is thus a useful metric for identifying the ideal number of top PCs to use.

in estimated and true error for DIS$^2$ evaluated on the full range of top PCs. A negative value implies that we have underestimated the error (i.e., the bound is not valid). We see that even this very simply metric roughly linearly correlates with the tightness of the bound, which suggests that evaluating over a range of top PC counts and only keeping predictions whose $\ell_1$ ratio is above a certain threshold can improve raw predictive accuracy without reducing coverage by too much. Figure G.5 shows that this is indeed the case: compared to DIS$^2$ evaluated on the logits, keeping all predictions above a score threshold can produce more accurate error estimates, without *too* severely underestimating error in the worst case.

## G.4 Making Baselines More Conservative with LOOCV

To more thoroughly compare DIS$^2$ to prior estimation techniques, we consider a strengthening of the baselines which may give them higher coverage without too much cost to prediction accuracy. Specifically, for each desired coverage level $\alpha \in [0.9, 0.95, 0.99]$, we use all but one of the datasets to learn a parameter to either scale or shift a method's predictions enough to achieve coverage $\alpha$. We then evaluate this scaled or shifted prediction on the distribution shifts of the remaining dataset, and we repeat this for each one.

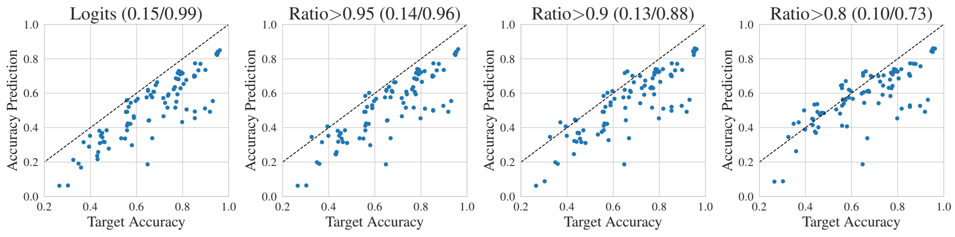


Figure G.5: **$\text{DIS}^2$ bounds and MAE / coverage as the cumulative $\ell_1$ ratio threshold is lowered.** Values in parenthesis are (MAE / coverage). By only keeping predictions with ratio above a varying threshold, we can smoothly interpolate between bound validity and raw error prediction accuracy.

The results, found in Table G.3, demonstrate that prior methods can indeed be made to have much higher coverage, although as expected their MAE suffers. Furthermore, they still underestimate error on the tail distribution shifts by quite a bit, and they rarely achieve the desired coverage on the heldout dataset—though they usually come reasonably close. In particular, ATC [Garg et al., 2022] and COT [Lu et al., 2023] do well with a shift parameter, e.g. at the desired coverage $\alpha = 0.95$ ATC matches $\text{DIS}^2$ in MAE and gets 94.4% coverage (compared to 98.9% by $\text{DIS}^2$). However, its conditional average overestimation is quite high, almost 9%. COT gets much lower overestimation (particularly for higher coverage levels), and it also appears to suffer less on the tail distribution shifts in the sense that $\alpha = 0.99$ does not induce nearly as high MAE as it does for ATC. However, at that level it only achieves 95.6% coverage, and it averages almost 5% accuracy overestimation on the shifts it does not correctly bound (compared to 0.1% by $\text{DIS}^2$). Also, its MAE is still substantially higher than $\text{DIS}^2$, despite getting lower coverage. Finally, we evaluate the scale/shift approach on our $\text{DIS}^2$ bound without the lower order term, but based on the metrics we report there appears to be little reason to prefer it over the untransformed version, one of the baselines, or the original $\text{DIS}^2$ bound.

Taken together, these results imply that if one's goal is predictive accuracy and tail behavior is not important (worst ˜10%), ATC or COT will likely get reasonable coverage with a shift parameter—though they still significantly underestimate error on a non-negligible fraction of shifts. If one cares about the long tail of distribution shifts, or prioritizes being conservative at a slight cost to average accuracy, $\text{DIS}^2$ is clearly preferable. Finally, we observe that the randomness which determines which shifts are not correctly bounded by $\text{DIS}^2$ is "decoupled" from the distributions themselves under Theorem 8.3.6, in the sense that it is an artifact of the random samples, rather than a property of the distribution (recall Figure 8.4b). This is in contrast with the shift/scale approach which would produce almost identical results under larger sample sizes because it does not account for finite sample effects. This implies that some distribution shifts are simply “unsuitable” for prior methods because they do not satisfy whatever condition these methods rely on, and observing more samples will not remedy this problem. It is clear that working to understand these conditions is crucial for reliability and interpretability, since we are not currently able to identify which distributions are suitable a priori.

## G.5 Proving that Assumption 8.3.5 Holds

Here we describe how the equivalence of Assumption 8.3.5 and the bound in Theorem 8.3.6 allow us to prove that the assumption holds with high probability. By repeating essentially the same proof as Theorem 8.3.6 in the other direction, we get the following corollary:

**Corollary G.5.1.** *If Assumption 8.3.5 does* not *hold, then with probability* $\geq 1-\delta$*,*

$$\epsilon_{\widehat{\mathcal{T}}}(\widehat{h}) > \epsilon_{\widehat{\mathcal{S}}}(\widehat{h}) + \widehat{\Delta}(\widehat{h}, h') - \sqrt{\frac{2(n_S+n_T)\log 1/\delta}{n_S n_T}}. \tag{G.5}$$

Note that the last term here is different from Theorem 8.3.6 because we are bounding the empirical target error, rather than the true target error. The reason for this change is that now we can make direct use of its contrapositive:

**Corollary G.5.2.** *If it is the case that*

$$\epsilon_{\widehat{\mathcal{T}}}(\widehat{h}) \leq \epsilon_{\widehat{\mathcal{S}}}(\widehat{h}) + \widehat{\Delta}(\widehat{h}, h') - \sqrt{\frac{2(n_S+n_T)\log 1/\delta}{n_S n_T}}, \tag{G.6}$$

*then either Assumption 8.3.5 holds, or an event has occurred which had probability* $\leq \delta$ *over the randomness of the samples* $\widehat{\mathcal{S}}, \widehat{\mathcal{T}}$*.*

We evaluate this bound on non-domain-adversarial shifts with $\delta = 10^{-6}$. As some of the BREEDS shifts have as few as 68 test samples, we restrict ourselves to shifts with $n_T \geq 500$ to ignore those where the finite-sample term heavily dominates; this removes a little over 20% of all shifts. Among the remainder, we find that the bound in Corollary G.5.2 holds 55.7% of the time when using full features and 25.7% of the time when using logits. This means that for these shifts, we can be essentially certain that Assumption 8.3.5—and therefore also Assumption 8.3.3—is true.

Note that the fact that the bound is *not* violated for a given shift does not at all imply that the assumption is not true. In general, the only rigorous way to prove that Assumption 8.3.5 does not hold would be to show that for a fixed $\delta$, the fraction of shifts for which the bound in Theorem 8.3.6 does not hold is larger than $\delta$ (in a manner that is statistically significant under the appropriate hypothesis test). Because this never occurs in our experiments, we cannot conclude that the assumption is ever false. At the same time, the fact that the bound *does* hold at least $1 - \delta$ of the time does not prove that the assumption is true—it merely suggests that it is reasonable and that the bound should continue to hold in the future. This is why it is important for Assumption 8.3.5 to be simple and intuitive, so that we can trust that it will persist and anticipate when it will not.

However, Corollary G.5.2 allows us to make a substantially stronger statement. In fact, it says that for *any* distribution shift, with enough samples, we can prove a posteriori whether or not Assumption 8.3.5 holds, because the gap between these two bounds will shrink with increasing sample size.

| Dataset | Source | Target |
|---|---|---|
| CIFAR10 | CIFAR10v1 | **CIFAR10v1, CIFAR10v2, CIFAR10C-Frost (severity 4), CIFAR10C-Pixelate (severity 5), CIFAR10-C Saturate (severity 5)** |
| CIFAR100 | CIFAR100 | **CIFAR100, CIFAR100C-Fog (severity 4), CIFAR100C-Motion Blur (severity 2), CIFAR100C-Contrast (severity 4), CIFAR100C-spatter (severity 2)** |
| Camelyon | **Camelyon (Hospital 1–3)** | Camelyon (Hospital 1–3), Camelyon (Hospital 4), Camelyon (Hospital 5) |
| FMoW | FMoW (2002–'13) | FMoW (2002–'13), FMoW (2013–'16), FMoW (2016–'18) |
| Entity13 | **Entity13 (ImageNetv1 sub-population 1)** | **Entity13 (ImageNetv1 sub-population 1), Entity13 (ImageNetv1 sub-population 2), Entity13 (ImageNetv2 sub-population 1), Entity13 (ImageNetv2 sub-population 2)** |
| Entity30 | **Entity30 (ImageNetv1 sub-population 1)** | **Entity30 (ImageNetv1 sub-population 1), Entity30 (ImageNetv1 sub-population 2), Entity30 (ImageNetv2 sub-population 1), Entity30 (ImageNetv2 sub-population 2)** |
| Living17 | **Living17 (ImageNetv1 sub-population 1)** | **Living17 (ImageNetv1 sub-population 1), Living17 (ImageNetv1 sub-population 2), Living17 (ImageNetv2 sub-population 1), Living17 (ImageNetv2 sub-population 2)** |
| Nonliving26 | **Nonliving26 (ImageNetv1 sub-population 1)** | **Nonliving26 (ImageNetv1 sub-population 1), Nonliving26 (ImageNetv1 sub-population 2), Nonliving26 (ImageNetv2 sub-population 1), Nonliving26 (ImageNetv2 sub-population 2)** |
| Officehome | Product | Product, Art, ClipArt, Real |
| DomainNet | Real | Real, Painiting, Sketch, ClipArt |
| Visda | **Synthetic (originally referred to as train)** | **Synthetic, Real-1 (originally referred to as val), Real-2 (originally referred to as test)** |

Table G.1: Details of the source and target datasets in our testbed.

| Dataset | Epoch | Batch size | $\ell_2$ regularization | Learning rate |
|---|---|---|---|---|
| CIFAR10 | 50 | 200 | 0.0001 (chosen from $\{0.0001, 0.001,$1e-5$, 0.0\}$) | 0.01 (chosen from $\{0.001, 0.01, 0.0001\}$) |
| CIFAR100 | 50 | 200 | 0.0001 (chosen from $\{0.0001, 0.001,$1e-5$, 0.0\}$) | 0.01 (chosen from $\{0.001, 0.01, 0.0001\}$) |
| Camelyon | 10 | 96 | 0.01 (chosen from $\{0.01, 0.001, 0.0001, 0.0\}$) | 0.03 (chosen from $\{0.003, 0.3, 0.0003, 0.03\}$) |
| FMoW | 30 | 64 | 0.0 (chosen from $\{0.0001, 0.001,$1e-5,0.0$\}$) | 0.0001 (chosen from $\{0.001, 0.01, 0.0001\}$) |
| Entity13 | 40 | 256 | 5e-5 (chosen from $\{$5e-5, 5e-4, 1e-4, 1e-5$\}$) | 0.2 (chosen from $\{0.1, 0.5, 0.2, 0.01, 0.0\}$) |
| Entity30 | 40 | 256 | 5e-5 (chosen from $\{$5e-5, 5e-4, 1e-4, 1e-5$\}$) | 0.2 (chosen from $\{0.1, 0.5, 0.2, 0.01, 0.0\}$) |
| Living17 | 40 | 256 | 5e-5 (chosen from $\{$5e-5, 5e-4, 1e-4, 1e-5$\}$) | 0.2 (chosen from $\{0.1, 0.5, 0.2, 0.01, 0.0\}$) |
| Nonliving26 | 40 | 256 | 0 5e-5 (chosen from $\{$5e-5, 5e-4, 1e-4, 1e-5$\}$) | 0.2 (chosen from $\{0.1, 0.5, 0.2, 0.01, 0.0\}$) |
| Officehome | 50 | 96 | 0.0001 (chosen from $\{0.0001, 0.001,$1e-5$, 0.0\}$) | 0.01 (chosen from $\{0.001, 0.01, 0.0001\}$) |
| DomainNet | 15 | 96 | 0.0001 (chosen from $\{0.0001, 0.001,$1e-5$, 0.0\}$) | 0.01 (chosen from $\{0.001, 0.01, 0.0001\}$) |
| Visda | 10 | 96 | 0.0001 (chosen from $\{0.0001, 0.001,$1e-5$, 0.0\}$) | 0.01 (chosen from $\{0.001, 0.01, 0.0001\}$) |

Table G.2: Details of the learning rate and batch size considered in our testbed

| | | MAE (↓) | | | Coverage (↑) | | | Overest. (↓) | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $\alpha \rightarrow$ | 0.9 | 0.95 | 0.99 | 0.9 | 0.95 | 0.99 | 0.9 | 0.95 | 0.99 |
| Method | Adjustment | | | | | | | | | |
| AC | none | | 0.106 | | | 0.122 | | | 0.118 | |
| | shift | 0.153 | 0.201 | 0.465 | 0.878 | 0.922 | 0.956 | 0.119 | 0.138 | 0.149 |
| | scale | 0.195 | 0.221 | 0.416 | 0.911 | 0.922 | 0.967 | 0.135 | 0.097 | 0.145 |
| DoC | none | | 0.105 | | | 0.167 | | | 0.122 | |
| | shift | 0.158 | 0.200 | 0.467 | 0.878 | 0.911 | 0.956 | 0.116 | 0.125 | 0.154 |
| | scale | 0.195 | 0.223 | 0.417 | 0.900 | 0.944 | 0.967 | 0.123 | 0.139 | 0.139 |
| ATC NE | none | | 0.067 | | | 0.289 | | | 0.083 | |
| | shift | 0.117 | 0.150 | 0.309 | 0.900 | 0.944 | 0.978 | 0.072 | 0.088 | 0.127 |
| | scale | 0.128 | 0.153 | 0.357 | 0.889 | 0.933 | 0.978 | 0.062 | 0.074 | 0.144 |
| COT | none | | 0.069 | | | 0.256 | | | 0.085 | |
| | shift | 0.115 | 0.140 | 0.232 | 0.878 | 0.944 | 0.956 | 0.049 | 0.065 | 0.048 |
| | scale | 0.150 | 0.193 | 0.248 | 0.889 | 0.944 | 0.956 | 0.074 | 0.066 | 0.044 |
| DIS$^2$ (w/o $\delta$) | none | | 0.083 | | | 0.756 | | | 0.072 | |
| | shift | 0.159 | 0.169 | 0.197 | 0.889 | 0.933 | 0.989 | 0.021 | 0.010 | 0.017 |
| | scale | 0.149 | 0.168 | 0.197 | 0.889 | 0.933 | 0.989 | 0.023 | 0.021 | 0.004 |
| DIS$^2$ ($\delta = 10^{-2}$) | none | | 0.150 | | | 0.989 | | | 0.001 | |
| DIS$^2$ ($\delta = 10^{-3}$) | none | | 0.174 | | | 1.000 | | | 0.000 | |

Table G.3: MAE, coverage, and conditional average overestimation for the strengthened baselines with a shift or scale parameter on non-domain-adversarial representations. Because a desired coverage $\alpha$ is only used when an adjustment is learned, "none"—representing no adjustment—does not vary with $\alpha$.

# Appendix H

# Appendix for Chapter 9

## H.1 Proofs of Hardness Results

We restate the theorems for convenience.

### H.1.1 Proof of Theorem 9.3.1

**Theorem H.1.1.** *Consider the task of learning a norm-bounded linear classifier. Fix any two costs $c_1, c_2$ with non-equal PD matrices, and let $0 \leq \epsilon \leq \frac{1}{2}$. There exists a distribution $p$ over $\mathcal{X} \times \mathcal{Y}$ such that:*

1. *For each of $c_1$ and $c_2$, there is a (different) classifier which achieves 0 error on $p$ when facing strategic response under that cost; and*
2. *Any classifier which achieves 0 error on $p$ under cost $c_1$ suffers error $\epsilon$ under cost $c_2$, and any classifier which achieves 0 error on $p$ under cost $c_2$ suffers error $1 - \epsilon$ under cost $c_2$.*

*Proof.* We will construct two distributions, one over the conditional $q(x \mid y = 1)$ and the other over $q(x \mid y = -1)$, and combine them via the mixture $q(y = 1) = \epsilon$, $q(y = -1) = 1 - \epsilon$. Let $\Sigma_1, \Sigma_2$ denote the cost matrices for $c_1, c_2$ respectively. Let $B$ denote the upper bound on the classifier norm. Pick any $\beta^*$ such that $\|\beta^*\|_{\Sigma_1} \neq \|\beta^*\|_{\Sigma_2}$, and define $r := \frac{u_* \|\beta^*\|}{3B}(\|\beta^*\|_{\Sigma_1} - \|\beta^*\|_{\Sigma_2})$. WLOG, suppose $r > 0$. Focusing on the negatively labeled portion, consider the $(d-1)$-dimensional plane in $\mathcal{X}$ defined by $\beta^{*\top} x = -r$. We let $q(x \mid y = -1)$ be any distribution with full support over that plane. Similarly, define the conditional distribution $q(x \mid y = 1)$ as a full-support distribution over the plane defined by $\beta^{*\top} x = r$. Thus the only classifier which achieves 0 error must have $\widehat{\beta} = \alpha \beta^*$ for some scaling factor $B/\|\beta^*\| \geq \alpha > 0$. The only remaining degree of freedom is the bias term $\widehat{\beta}_0$.

Consider the case where the true cost is $c_1$. If we wish to correctly classify the negative points under strategic response, we must classify the negatively labeled plane with margin greater than $u_* \|\beta^*\|_{\Sigma_1}$. However, if we wish to do the same with the positive points, the margin for that plane must be less than or equal to this same value. Formally, we must have

$$\beta^{*\top} x = -r \implies \widehat{\beta}^\top x + \widehat{\beta}_0 < -u_* \|\beta^*\|_{\Sigma_1}, \tag{H.1}$$

$$\beta^{*\top} x = r \implies \widehat{\beta}^\top x + \widehat{\beta}_0 \geq -u_* \|\beta^*\|_{\Sigma_1}, \tag{H.2}$$

which, remembering that $\widehat{\beta} = \alpha\beta^*$, immediately implies

$$\widehat{\beta}_0 - \alpha r < -u_* \|\beta^*\|_{\Sigma_1}, \tag{H.3}$$

$$\widehat{\beta}_0 + \alpha r \geq -u_* \|\beta^*\|_{\Sigma_1}. \tag{H.4}$$

Thus, we have

$$-u_* \|\beta^*\|_{\Sigma_1} - \alpha r \leq \widehat{\beta}_0 < -u_* \|\beta^*\|_{\Sigma_1} + \alpha r, \tag{H.5}$$

and since $\alpha r > 0$, this describes the non-empty set of all classifiers which achieve 0 error under cost $c_1$. By an analogous argument we can construct the set of classifiers which achieve 0 error under cost $c_2$.

However, observe that

$$2\alpha r = \frac{2}{3} \frac{\|\beta^*\|}{B} u_* (\|\beta^*\|_{\Sigma_1} - \|\beta^*\|_{\Sigma_2}) \tag{H.6}$$

$$< u_* (\|\beta^*\|_{\Sigma_1} - \|\beta^*\|_{\Sigma_2}), \tag{H.7}$$

and therefore

$$-u_* \|\beta^*\|_{\Sigma_1} + \alpha r < -u_* \|\beta^*\|_{\Sigma_2} - \alpha r. \tag{H.8}$$

This means that the upper bound for any $\widehat{\beta}_0$ which achieves 0 error under cost $c_1$ cannot satisfy the lower bound for cost $c_2$, which means the positively labeled plane will be classified negatively with a margin that is too large for strategic response, causing error $\epsilon$. By a symmetric argument, any $\widehat{\beta}_0$ which achieves 0 error under cost $c_2$ cannot satisfy the upper bound for cost $c_1$, implying the negatively labeled plane will strategically shift and cause error $1 - \epsilon$. Finally, we remark that while the distribution defined here is not absolutely continuous on $\mathcal{X}$, this can be remedied by simply making the distribution a product of the constructed planar distribution and a uniform distribution along the direction $\beta^*$ with width sufficiently small (e.g, width $c \cdot \alpha r$ for $c \ll 1$). $\square$

### H.1.2 Proof of Theorem 9.3.2

**Theorem H.1.2.** *Define the distribution $q$ over $\mathcal{X} \times \mathcal{Y}$ as $q(y=1) = q(y=-1) = 1/2$, $q(x \mid y) \overset{d}{=} \mathcal{N}(y \cdot \mu_0, \sigma^2 I)$. Denote by $\Phi$ the standard Normal CDF. Let the true cost be defined as $\|x - x'\|_\Sigma$ with unknown cost matrix $\Sigma$, and let $\beta^*$ be the classifier which minimizes the strategic 0-1 risk under this cost.*

*Suppose one instead learns a classifier $\widehat{\beta}$ by assuming an incorrect cost $\widehat{\Sigma}$ and minimizing the population strategic 0-1 risk under that cost: $\widehat{\beta} := \arg\min_\beta \mathbb{E}_q[\ell_{0-1}^{\widehat{c}}(\beta)]$. Then the excess 0-1 risk suffered by $\widehat{\beta}$ is*

$$\Phi\left(\frac{\|\mu_0\|}{\sigma}\right) - \frac{1}{2}\left(\Phi\left(\frac{\|\mu_0\| - \epsilon}{\sigma}\right) + \Phi\left(\frac{\|\mu_0\| + \epsilon}{\sigma}\right)\right), \tag{H.9}$$

*where $\epsilon := \frac{u_* | \|\mu_0\|_{*,\widehat{\Sigma}^{-1}} - \|\mu_0\|_{*,\Sigma^{-1}} |}{\|\mu_0\|}$.*

*Proof.* By the Neyman-Pearson Lemma, the classifier which minimizes non-strategic 0-1 risk will be the one which predicts $\operatorname{sign}(\ln \frac{p(y=1|x)}{p(y=0|x)})$, which gives $\beta^* = 2\mu_0$. To account for strategic response, we observe that as proven in Lemma 9.2.3, each user will have $\delta(x)$ in the direction of $\beta^*$ which induces a change in their predicted value by at most $u_* \|\beta^*\|_* = 2u_* \|\mu_0\|_{*,\Sigma^{-1}}$; since we are considering the 0-1 loss, this is equivalent to *every* user shifting in this way.

It is immediate that to find the corresponding minimizer of the population 0-1 *strategic* risk, we can simply add a negative bias equal to the change induced by this shift, because this maintains the same labels as before on all points, *after* they strategically shift. Therefore, the strategic risk minimizer will be $2\mu_0^\top x - 2u_* \|\mu_0\|_{*,\Sigma^{-1}}$. By the same argument, the minimizer of the 0-1 strategic risk under the incorrect cost matrix $\widehat{\Sigma}$ is $2\mu_0^\top x - 2u_* \|\mu_0\|_{*,\widehat{\Sigma}^{-1}}$. It remains to derive a lower bound on their difference in risk.

First note that as argued above, the 0-1 strategic risk of the minimizer for the correct cost is the same as the non-strategic risk of the non-strategic solution. For the both positively and negatively labeled points, this is equal to $\Phi\left(-\frac{\|\mu_0\|}{\sigma}\right)$ due to rotational symmetry. To determine the risk of the incorrect solution, we can identify the regions whose label will differ under that classifier and bound the measure of those regions. Define $\gamma = 2u_*(\|\mu_0\|_{*,\widehat{\Sigma}^{-1}} - \|\mu_0\|_{*,\Sigma^{-1}})$ as the difference in the two solutions' predictions on all $x$. Due to the symmetry of the positive and negative conditional distributions, we can assume WLOG that $\gamma > 0$, i.e., the incorrect classifier assigns a smaller prediction to all $x$. This means it will have less risk on a region of negative points and more on positive. Specifically, the two classifiers will differ on all points for which the true strategic-optimal classifier assigns a value *before strategic response* which lies in $(-2u_* \|\mu_0\|_{*,\Sigma^{-1}}, -2u_* \|\mu_0\|_{*,\Sigma^{-1}} + \gamma)$;

those assigned a value less than $-2u_*\|\mu_0\|_{*,\Sigma^{-1}}$ will receive a negative prediction from both classifiers, and those assigned a value greater than $-2u_*\|\mu_0\|_{*,\Sigma^{-1}} + \gamma$ will still be close enough to the decision boundary that they can shift to achieve a positive label prediction from $\widehat{\beta}$. Formally, this region is

$$\{x \mid -2u_*\|\mu_0\|_{*,\Sigma^{-1}} < \beta^{*\top} x < -2u_*\|\mu_0\|_{*,\Sigma^{-1}} + \gamma\} = \{x \mid 0 < \mu_0^\top x < \gamma/2\}. \tag{H.10}$$

This region depends only on the value $\mu_0^\top x$. Since the negative points are distributed as $\mathcal{N}(-\mu_0, \sigma^2 I)$, this term has the distribution $\mu_0^\top x \sim \mathcal{N}(-\|\mu_0\|^2, \sigma^2\|\mu_0\|^2)$. Therefore, the measure of this region under $q(x \mid y = -1)$ is

$$\Phi\left(\frac{\gamma/2 + \|\mu_0\|^2}{\sigma\|\mu_0\|}\right) - \Phi\left(\frac{\|\mu_0\|}{\sigma}\right) \tag{H.11}$$

$$= \int_0^{\frac{\gamma}{2\|\mu_0\|}} \rho\left(\frac{\|\mu_0\| + z}{\sigma}\right)\, dz, \tag{H.12}$$

This is the amount by which the risk of the incorrect solution will *decrease* on the negative points. Likewise, the risk will *increase* on positive points in this region, which under $q(x \mid y = 1)$ has measure

$$\Phi\left(\frac{\gamma/2 - \|\mu_0\|^2}{\sigma\|\mu_0\|}\right) - \Phi\left(\frac{-\|\mu_0\|}{\sigma}\right) \tag{H.13}$$

$$= \int_0^{\frac{\gamma}{2\|\mu_0\|}} \rho\left(\frac{-\|\mu_0\| + z}{\sigma}\right)\, dz, \tag{H.14}$$

Therefore, the overall increase to risk will be

$$\frac{1}{2}\int_0^{\epsilon} \rho\left(\frac{-\|\mu_0\| + z}{\sigma}\right) - \rho\left(\frac{\|\mu_0\| + z}{\sigma}\right)\, dz, \tag{H.15}$$

where $\epsilon := \frac{u_*|\|\mu_0\|_{*,\widehat{\Sigma}^{-1}} - \|\mu_0\|_{*,\Sigma^{-1}}|}{\|\mu_0\|}$. By applying the fundamental theorem of calculus and the fact that $\Phi(x) = 1 - \Phi(-x)$ we arrive at the stated equality. $\square$

## H.2 Proof of Theorem 9.4.2 and Proposition 9.4.3

**Theorem H.2.1.** *Let $\|\cdot\|$ be a $p$-norm and fix a cost matrix $\Sigma$. For any distribution $q$ on $\mathcal{X} \times \mathcal{Y}$ with $q(y = 1) =: \tau^+$, the dual-regularized loss $R^c_{\text{s-hinge}}(\beta) + \lambda u_*\|\beta\|_*$ is guaranteed to be convex for $\lambda \geq \tau^+$. In contrast, the $\ell_2$-regularized loss $R^c_{\text{s-hinge}}(\beta) + \lambda u_*\|\beta\|_2$ is non-convex unless $\lambda \geq \tau^+\|\Sigma^{-1/2}\|_2$.*

*Proof.* Writing out the dual-regularized loss with the full definition of the strategic hinge,

$$\begin{aligned}R^c_{\text{s-hinge}}(\beta) + \lambda u_* \|\beta\|_* &= \mathbb{E}_{(x,y)\sim q}[\max\{0, 1 - y(\beta^\top x + u_* \|\beta\|_*)\}] + \lambda u_* \|\beta\|_* \\ &= \tau^+ \mathbb{E}_{q(x|y=1)}[\max\{0, 1 - \beta^\top x - u_* \|\beta\|_*\} + {\lambda}/{\tau^+} u_* \|\beta\|_*] + \\ &\qquad (1-\tau^+)\mathbb{E}_{q(x|y=-1)}[\max\{0, 1 + \beta^\top x + u_* \|\beta\|_*\}].\end{aligned}$$

The last term is already convex in $\beta$. Rewriting the first term, we get

$$\tau^+ \mathbb{E}_{q(x|y=1)}[\max\{{\lambda}/{\tau^+} u_* \|\beta\|_*, 1 - \beta^\top x + u_* \left({\lambda}/{\tau^+} - 1\right) \|\beta\|_*\}] \tag{H.16}$$

For $\lambda \geq \tau^+$ this is the expectation of the maximum of two convex functions, and thus the full loss is convex.

To see why the $\ell_2$ norm requires much stronger regularization, consider again the above term with regularization $\lambda u_* \|\beta\|_2$:

$$\tau^+ \mathbb{E}_{x|y=1}\left[\max\left\{\frac{\lambda}{\tau^+} u_* \|\beta\|_2, 1 - \beta^\top x + u_* \left(\frac{\lambda}{\tau^+}\|\beta\|_2 - \|\beta\|_*\right)\right\}\right]. \tag{H.17}$$

The first term in the max is convex—so to show non-convexity, we will consider values where the second term is larger. Recalling that $\|v\|_* := \|\Sigma^{-1/2} v\|_q$, write the second term as a function $f(\beta) := 1 - \beta^\top x + u_* \left(\frac{\lambda}{\tau^+}\|\beta\|_2 - \|\Sigma^{-1/2}\beta\|_q\right)$, and thus

$$\nabla f(\beta) = -x + u_* \left(\frac{\lambda}{\tau^+} \frac{\beta}{\|\beta\|_2} - \Sigma^{-1/2}\left(\frac{\partial}{\partial v}\|v\|_q\big|_{v=\Sigma^{-1/2}\beta_2}\right)\right). \tag{H.18}$$

Recall that a function is convex if and only if for all $x, y$ in its domain,

$$f(x) - f(y) \geq \nabla f(y)^\top (x - y). \tag{H.19}$$

This means that $f$ is convex only if for *all* vectors $\beta_1, \beta_2 \in \mathcal{B}$,

$$\frac{\lambda}{\tau^+}(\|\beta_1\|_2 - \|\beta_2\|_2) - \|\Sigma^{-1/2}\beta_1\|_q + \|\Sigma^{-1/2}\beta_2\|_q \tag{H.20}$$

$$\geq \left(\frac{\lambda}{\tau^+}\frac{\beta_2}{\|\beta_2\|_2} - \Sigma^{-1/2}\left(\frac{\partial}{\partial v}\|v\|_q\big|_{v=\Sigma^{-1/2}\beta_2}\right)\right)^\top (\beta_1 - \beta_2). \tag{H.21}$$

Without loss of generality, suppose $\Sigma$ is diagonal. Let $v_i,\ i \in [d]$ denote the eigenvectors of $\Sigma$ with *decreasing* eigenvalues $\sigma_1^2, \sigma_2^2, \ldots$. Choose any $\beta_2 =$

$\sum_i \lambda_i v_i$ with non-negative $\lambda_i$ and $\lambda_d = 0$. Thus $\Sigma^{-1/2}\beta_2 = \sum_i (\lambda_i/\sigma_i)\, v_i$. Then

$$\beta_2^\top \Sigma^{-1/2} \frac{\partial}{\partial v}\|v\|_q\Big|_{v=\Sigma^{-1/2}\beta_2} = \sum_i (\lambda_i/\sigma_i) \cdot \left(\frac{|\lambda_i/\sigma_i|}{\|\Sigma^{-1/2}\beta_2\|_q}\right)^{q-1} \tag{H.22}$$

$$= \frac{\|\Sigma^{-1/2}\beta_2\|_q^q}{\|\Sigma^{-1/2}\beta_2\|_q^{q-1}} \tag{H.23}$$

$$= \|\Sigma^{-1/2}\beta_2\|_q. \tag{H.24}$$

This allows us to simplify the above inequality and arrive at the condition

$$\frac{\lambda}{\tau^+}\|\beta_1\|_2 - \|\Sigma^{-1/2}\beta_1\|_q \geq \beta_1^\top \left(\frac{\lambda}{\tau^+}\frac{\beta_2}{\|\beta_2\|_2} - \Sigma^{-1/2}\frac{\partial}{\partial v}\|v\|_q\Big|_{v=\Sigma^{-1/2}\beta_2}\right). \tag{H.25}$$

Now choose $\beta_1 = cv_d$ for some scalar $c \neq 0$. Then the RHS vanishes and $f(\beta)$ is convex only if

$$\frac{\lambda c}{\tau^+} - \frac{c}{\sigma_d} \geq 0 \tag{H.26}$$

$$\iff \lambda \geq \frac{\tau^+}{\sigma_d} = \tau^+\|\Sigma^{-1/2}\|_2. \tag{H.27}$$

Thus we've proven the required lower bound on $\lambda$. What remains is to show that this condition applies at a location in parameter space where the negatively labeled samples do not contribute to the gradient, and where the losses on the positively labeled samples are dominated by the second term of the maximum.

To do this, we scale down $\beta_2 \to 0$ and choose the bias as $\beta_0 = -(1 + u_*\|\beta_2\|_* + \epsilon(X+1))$ for some very small positive $\epsilon$. It follows that for all $x$,

$$\beta_2^\top x + \beta_0 + u_*\|\beta_2\|_* \leq \epsilon X - (1 + u_*\|\beta_2\|_* + \epsilon(X+1)) + u_*\|\beta_2\|_* \tag{H.28}$$

$$\leq -(1+\epsilon). \tag{H.29}$$

This accomplishes both desiderata: first, it ensures that the loss on all negatively labeled points is $\max\{0, 1 + \beta^\top x + \beta_0 + u_*\|\beta\|_*\} \leq \max\{0, -\epsilon\} = 0$, with gradient equal to 0. Second, it ensures that on the positive examples, the second term in the loss dominates. We can see this by observing that the second term being larger is equivalent to

$$0 < 1 - \left(\beta_2^\top x + \beta_0 + u_*\|\beta_2\|_*\right), \tag{H.30}$$

and by construction the RHS is at least $2 + \epsilon$ for all $x$. $\square$

## H.3 Proofs of Lemmas in Main Body

### H.3.1 Proofs of Lemmas 9.2.3 and 9.2.4

**Lemma H.3.1.** *For any cost $c(x, x') = \phi(\|x - x'\|_\Sigma)$, the maximum change to a user's prediction score that can result from strategic behavior is given by*

$$\beta^\top x(\beta) - \beta^\top x \leq u_* \|\beta\|_* \tag{H.31}$$

*where $\|\beta\|_* := \|\beta\|_{*,\Sigma^{-1}} = \sup_{\|v\|_\Sigma = 1} \beta^\top v$ is the $\Sigma$-transformed dual norm of $\beta$ and $u_* := \sup r \in \mathbb{R}_{\geq 0}$ s.t. $\phi(r) \leq u$.*

*Proof.* A user at $x$ will move so as to maximize the inner product $\beta^\top x(\beta)$ so long as the cost of this move does not exceed the additional utility $u$ (and only up until the point that they achieve a positive classification). In other words, the maximum logit they will feasibly achieve is given by the optimization problem

$$\sup_{x'} \beta^\top x' \quad \text{s.t. } c(x, x') \leq u. \tag{H.32}$$

We can reparameterize $x' = x + \delta$ to rewrite the objective as

$$\beta^\top x + \sup_{\{\delta \,:\, \phi(\|\delta\|_\Sigma) \leq u\}} \beta^\top \delta, \tag{H.33}$$

which, recalling the definition of $u_*$ and monotonicity of $\phi$, is equal to

$$\beta^\top x + \sup_{\{\delta \,:\, \|\delta\|_\Sigma \leq u_*\}} \beta^\top \delta \tag{H.34}$$

Here we recognize the variational formula for the dual norm, giving the solution $\beta^\top x + u_* \|\beta\|_*$. □

**Lemma H.3.2.** *For any cost $c \in \mathcal{C}$, $R^c_{0-1}(\beta) \leq R^c_{\text{s-hinge}}(\beta)$.*

*Proof.* For a fixed sample $(x, y)$, recall the loss definitions:

$$\ell^c_{0-1}(\beta) := \mathbf{1}\{\text{sign}(\beta^\top (x + \delta)) \neq y\} \tag{H.35}$$

$$\ell^c_{\text{hinge}}(\beta) := \max\left(0, 1 - y\beta^\top (x + \delta)\right) \tag{H.36}$$

$$\ell^c_{\text{s-hinge}}(\beta) := \max\left(0, 1 - y(\beta^\top x + u_* \|\beta\|_*)\right). \tag{H.37}$$

Since strategic response is agnostic to the loss used (i.e., $\delta$ does not change) and the hinge loss upper bounds the 0-1 loss, it is immediate that $\ell^c_{0-1} \leq \ell^c_{\text{hinge}}$. Consider any point with true label $y = 1$. If the point is positively classified (whether it moves or not) then $\ell^c_{0-1} = 0 \leq \ell^c_{\text{s-hinge}}$. If the point is negatively classified and does not move, this means $\beta^\top x < -u_* \|\beta\|_* \implies \beta^\top x + u_* \|\beta\|_* < 0$, and therefore $\ell^c_{0-1} = 1 < \ell^c_{\text{s-hinge}}$. So the claim holds for any point with $y = 1$.

Next, if a point with true label $y = -1$ does not move, then neither loss changes as a result of strategic response, which means the strategic hinge loss is no less than the regular hinge loss. It remains to prove the inequality for points with $y = -1$ which move in response to the classifier. By Lemma 9.2.3 the classifier's output after strategic response will increase by no more than $u_* \|\beta\|_*$. We have

$$\begin{aligned}
\ell^c_{\text{hinge}}(\beta) &= \ell_{\text{hinge}}(\beta^\top (x + \delta), y = -1) && \text{(H.38)}\\
&= \max\{0, 1 + \beta^\top (x + \delta)\} && \text{(H.39)}\\
&\leq \max\{0, 1 + \beta^\top x + u_* \|\beta\|_*\} && \text{(H.40)}\\
&= \ell_{\text{s-hinge}}(\beta; u_* \|\beta\|_*) && \text{(H.41)}\\
&= \ell^c_{\text{s-hinge}}(\beta). && \square
\end{aligned}$$

### H.3.2 Proof of Lemma 9.4.1

**Lemma H.3.3.** *Fix some classifier $\beta$. Then for any dataset $(\mathcal{X} \times \mathcal{Y})^n$ and uncertainty set $\mathcal{C}$,* MaxLossCost *runs in $\mathcal{O}(nd + n \ln n)$ time and returns the value $k^* \in \mathbb{R}_{\geq 0}$ which maximizes the k-shifted strategic hinge loss $\widehat{R}_{\text{s-hinge}}(\beta; k)$ subject to $k^* = u_* \|\beta\|_*$ for some cost $c \in \mathcal{C}$.*

*Proof.* Recall the regularized strategic hinge loss $\widehat{R}_{\text{s-hinge}}(\widehat{\beta}; u_* \|\widehat{\beta}\|_*) = \frac{1}{n} \sum_{i=1}^n \max\{0, 1 - y_i(\widehat{\beta}^\top x + u_* \|\widehat{\beta}\|_*)\} + \lambda u_* \|\widehat{\beta}\|_*$. As this function depends on $c$ only through the dual norm, and since $\mathcal{C}$ is a convex set and the norm is continuous, the worst-case cost *scalar* can be reparameterized as the argmax over $k \in [\|\widehat{\beta}\|_*^{\min}, \|\widehat{\beta}\|_*^{\max}]$ of $\widehat{R}_{\text{s-hinge}}(\widehat{\beta}; u_* k)$. This function is one-dimensional and piecewise linear in $k$, and therefore the maximum must occur either at an endpoint or at the boundary between two linear segments.

By sorting the $v_i := y_i(1 - y_i \widehat{\beta}^\top x_i)$, we get the values $1 - y_i \widehat{\beta}^\top x_i$ with $y = +1$ in increasing order and those with $y = -1$ in decreasing order. At each step, we maintain the condition that for all $j' < j$, $v_{j'} - u_* k \leq 0$. It follows that by increasing $k$ to the boundary of the next linear segment at $k'$, there are exactly $c_{+1}$ points for which the loss will decrease by $u_*(k' - k)$ and $c_{-1}$ points for which the loss will increase by that same amount, while the regularization term increases by $\lambda u_*(k' - k)$. Thus $r$ tracks the induced risk for the current $k$, and we keep track of the $k$ which so far induces the maximum risk. Finally, since we have moved to the next linear segment: either an example with $y = +1$ now has 0 loss and will not change for the remainder; or an example with $y = -1$ has $> 0$ loss and will contribute linearly to the risk for the remainder. We therefore update the appropriate count and iterate. The algorithm is complete when we reach the boundary $k = \|\beta\|_*^{\max}$. If before this point we reach the end of the sorted $v_j$, then we know that for the remaining possible increase $\|\beta\|_*^{\max} - k$, only the loss on the negative examples will change, growing linearly until the boundary. So we do one last evaluation and return the maximum. $\square$

## H.4 Proof of Rademacher Generalization Bound

**Theorem H.4.1** (Strategic Hinge Generalization Bound). *Fix a norm $\|\cdot\|$. Assume $\max_{x\in\mathcal{D}} \|x\| \leq X$ and $\|\beta\|_2, \|\beta\|_* \leq B,\ \forall \beta \in \mathcal{B}, c \in \mathcal{C}$. Then with probability $\geq 1-\delta$, for all $\beta \in \mathcal{B}$ and all cost functions $c \in \mathcal{C}$,*

$$R^c_{0-1}(\beta) \leq \widehat{R}^c_{\textit{s-hinge}}(\beta) + \frac{B(4X + u_*) + 3\sqrt{\ln 1/\delta}}{\sqrt{n}} \tag{H.42}$$

Lemma 9.2.4 shows that $R^c_{0-1}(\beta) \leq R^c_{\text{s-hinge}}(\beta)$. The result then follows from standard Rademacher bounds, requiring only the following additional Lemma:
**Lemma H.4.2.** *For any set of $n$ samples,*

$$\widehat{\mathcal{R}}_n(\ell_{\textit{s-hinge}} \circ \mathcal{B}) \leq \frac{B(4X + u_*)}{2\sqrt{n}}. \tag{H.43}$$

*Proof.* Define the function class $\mathcal{H} := \{x \mapsto \beta^\top x + z(y)u_*\|\beta\|_*\}$ ($z(y)$ follows notation from Levanon and Rosenfeld [2022]—in our case it is always equal to 1 but more generally we let it be a map from $\{\pm 1\} \mapsto \{\pm 1\}$). With the definition of Rademacher complexity,

$$\widehat{\mathcal{R}}_n(\mathcal{H}) = \mathbb{E}_\sigma\left[\sup_{\beta\in\mathcal{B},c\in\mathcal{C}} \frac{1}{n}\sum_{i=1}^n \sigma_i \cdot (\beta^\top x + z(y)u_*\|\beta\|_*)\right] \tag{H.44}$$

$$\leq \mathbb{E}_\sigma\left[\sup_{\beta\in\mathcal{B}} \frac{1}{n}\sum_{i=1}^n \sigma_i\beta^\top x\right] + \mathbb{E}_\sigma\left[\sup_{\beta\in\mathcal{B},c\in\mathcal{C}} \frac{1}{n}\sum_{i=1}^n \sigma_i z(y)u_*\|\beta\|_*\right]. \tag{H.45}$$

The first term is the empirical Rademacher complexity of norm-bounded linear functions which is well known to be upper bounded by $\frac{2BX}{\sqrt{n}}$. An error in the proof by Levanon and Rosenfeld [2022] dropped the second term from the calculation of the Rademacher complexity. We observe that it is not zero, but we can bound it as follows:

Since $z(y) = \pm 1$ it can be dropped due to the symmetry of the Rademacher variables (since the $y$ are fixed). Also, if the sum of the Rademacher variables is negative, the supremizing $\beta$ will have dual norm 0, and if the sum is positive, it will have dual norm $B$. Thus,

$$\mathbb{E}_\sigma\left[\sup_{\beta\in\mathcal{B},c\in\mathcal{C}} \frac{1}{n}\sum_{i=1}^{n}\sigma_i u_*\|\beta\|_*\right] = \mathbb{P}\left(\sum\sigma_i > 0\right)\frac{u_*B}{n}\mathbb{E}_\sigma\left[\sum\sigma_i \middle| \sum\sigma_i > 0\right] \tag{H.46}$$

$$= \frac{u_*B}{2n}\mathbb{E}_\sigma\left[\left|\sum\sigma_i\right|\right] \tag{H.47}$$

$$\leq \frac{u_*B}{2n}\sqrt{\mathbb{E}_\sigma\left[\left(\sum\sigma_i\right)^2\right]} \tag{H.48}$$

$$= \frac{u_*B}{2\sqrt{n}}, \tag{H.49}$$

where the second equality is due to the symmetry of the distribution over $\sigma$ and the inequality is by Jensen's. Since the function class $\ell_{\text{s-hinge}} \circ \mathcal{B}$ is generated by a 1-Lipschitz function applied to $\mathcal{H}$, the claim follows by Talagrand's contraction lemma. □

## H.5 Proof for Full-Batch Subgradient Method

**Algorithm 9** Subgradient method on k-shifted strategic hinge loss

```
input: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, Iteration number $T$, Cost uncertainty set $\mathcal{C}$.
Upper bound $u_*$, Regularization parameter $\lambda$.
define: $\beta^{(1)} \leftarrow \mathbf{0}$.
        $\eta \leftarrow \frac{B}{L\sqrt{T}}$.
        $\Sigma_{\min} \leftarrow \text{diag}([\sigma_{1\ell}, \ldots, \sigma_{d\ell}])$.
for $t = 1, \ldots, T$ do
  1. $c_t \leftarrow \textsc{MaxLossCost}(\mathcal{D}, \beta^{(t)}, \mathcal{C}, u_*, \lambda)$.
  2. Choose $g_t \in \partial_\beta[\widehat{R}^{c_t}_{\text{s-hinge}}(\beta^{(t)}) + \lambda u_*\|\beta\|_*]$.
  3. $\beta^{(t+1)} \leftarrow \beta^{(t)} - \eta\Sigma_{\min} g_t$
end for
return $\beta^{(t^*)}$, where $t^* := \arg\min_t \widehat{R}^{c_t}_{\text{s-hinge}}(\beta^{(t)}) + \lambda u_*\|\beta^{(t)}\|_*$
```

**Algorithm 10** MAXLOSSCOST subprocedure for Algorithm 9

```
input: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, parameters $\beta$, Cost uncertainty set $\mathcal{C}$, Upper bound $u_*$, Regularization parameter $\lambda$.
define: $\|\beta\|_*^{\min} := \min_{c\in\mathcal{C}} \|\beta\|_*$, $\|\beta\|_*^{\max} := \max_{c\in\mathcal{C}} \|\beta\|_*$.
initialize: $k \leftarrow \|\beta\|_*^{\min}$, $k_{\max} \leftarrow k$.
initialize: $r \leftarrow \widehat{R}_{\text{s-hinge}}(\beta; u_* k) + \lambda u_* k$, $r_{\max} \leftarrow r$.
1. Evaluate $v_i = y_i(1 - y_i\beta^\top x_i)$ for all $x_i$.
2. Sort $v_i$ in increasing order to get sorted indices $j$.
3. Initialize index $j \leftarrow \min j$ s.t. $v_j - u_* k > 0$.
4. Initialize counts $c_{+1} \leftarrow |\{j' \geq j : y_{j'} = +1\}|, c_{-1} \leftarrow |\{j' < j : y_{j'} = -1\}|$.
while $k < \|\beta\|_*^{\max}$ && $j < n$ do
  $k' \leftarrow \min\{{}^{v_j}/_{u_*}, \|\beta\|_*^{\max}\}$.
  $r \leftarrow r + u_*(k' - k)\left(\lambda + \frac{c_{-1} - c_{+1}}{n}\right)$.
  $k \leftarrow k'$.
  if $r > r_{\max}$ then
    $r_{\max} \leftarrow r$, $k_{\max} \leftarrow k$.
  end if
  $c_{y_j} \leftarrow c_{y_j} - y_j$.
  $j \leftarrow j + 1$.
end while

# Found maximizing norm scalar, now need matrix $\Sigma$ which induces it.
if $j == n$ && $r + u_*\left[\|\beta\|_*^{\max} - k\right]\left(\lambda + \frac{n^-}{n}\right) > r_{\max}$ then
  return $\arg\max_{c\in\mathcal{C}} \|\beta\|_* = \text{diag}([\sigma_{1\ell}, \ldots, \sigma_{d\ell}])$.
else
  initialize: $\widehat{\sigma} \leftarrow [\sigma_{1u}, \ldots, \sigma_{du}]$.
  for $i = 1, \ldots, d$ do
    Let $\widehat{\Sigma} := \text{diag}([\widehat{\sigma}_1, \ldots, \sigma_{i\ell}, \ldots, \widehat{\sigma}_d])$.
    if $\|\beta\|_{*,\widehat{\Sigma}^{-1}} < k_{\max}$ then
      $\widehat{\sigma}_i \leftarrow \sigma_{i\ell}$.
      continue.
    end if
    Let $\widehat{\Sigma}_{i=0} := \text{diag}([\widehat{\sigma}_1, \ldots, 0_i, \ldots, \widehat{\sigma}_d])$.
    $\widehat{\sigma}_i \leftarrow \frac{|\beta_i|}{\left(k_{\max}^p - \|\widehat{\Sigma}_{i=0}^{-1/2}\beta\|_*^p\right)^{1/p}}$.
    return $\text{diag}(\widehat{\sigma})$.
  end for
end if
```

**Theorem H.5.1.** *Suppose we run the subgradient method on the regularized k-shifted strategic hinge loss as described in Algorithm 9 for $T$ iterations and get classifier $\widehat{\beta}$. Then with probability $\geq 1-\delta$, the worst-case 0-1 strategic loss under costs in $\mathcal{C}$ can be bounded by*

$$\max_{c\in\mathcal{C}} R^c_{0-1}(\widehat{\beta}) \leq \max_{c\in\mathcal{C}} \widehat{R}^c_{\textit{s-hinge}}(\widehat{\beta}) + \frac{B(4X+u_*)+3\sqrt{\ln {}^2\!/\!_\delta}}{\sqrt{n}}. \tag{H.50}$$

*Furthermore, the sub-optimality of $\widehat{\beta}$ with respect to the population minimax solution is bounded by*

$$\max_{c\in\mathcal{C}} \widehat{R}^c_{\textit{s-hinge}}(\widehat{\beta}) \leq \min_{\beta}\max_{c\in\mathcal{C}} R^c_{\textit{s-hinge}}(\beta) + B\left(\frac{L}{\sqrt{T}} + (X+u_*)\sqrt{\frac{\ln {}^2\!/\!_\delta}{2n}}\right). \tag{H.51}$$

*Proof.* The first statement follows immediately with probability $\geq 1 - {}^\delta\!/\!_2$ from Theorem H.4.1 since the bound holds uniformly for all $c \in \mathcal{C}$. Now we prove the second statement also holds with probability $\geq 1-{}^\delta\!/\!_2$, which we then combine via union bound. By Danskin's theorem, the function $r(\beta) := \max_{c\in\mathcal{C}} \widehat{R}^c_{\text{s-hinge}}(\beta)$ is convex in $\beta$, and its subgradient is defined by $\partial_\beta \widehat{R}^{c^*}_{\text{s-hinge}}(\beta)$ where $c^* := \arg\max_{c\in\mathcal{C}} \widehat{R}^c_{\text{s-hinge}}(\beta)$.

A standard result says that if we run the subgradient method on $r(\beta)$ with step size $\eta = \frac{\epsilon}{L^2}$ for $T \geq \frac{B^2L^2}{\epsilon^2}$ steps, we will have $r(\beta^{t^*}) - \min_\beta r(\beta) \leq \epsilon$, which matches the hyperparameters of Algorithm 9 with $\epsilon = \frac{LB}{\sqrt{T}}$. The descent lemma in our setting is a bit different: we apply it to the $\Sigma_{\min}$-transformed gradient and show convergence in the norm $\|\cdot\|_{\Sigma^{-1}_{\min}}$. That is, our update is $\beta^{(t+1)} = \beta^{(t)} - \eta\Sigma_{\min} g_t$, and therefore

$$\|\beta^{(t+1)} - \beta^*\|^2_{\Sigma^{-1}_{\min}} = \|\beta^{(t)} - \beta^*\|^2_{\Sigma^{-1}_{\min}} - 2\eta g_t^\top(\beta^{(t)}-\beta^*) + \eta^2\|g_t\|^2_{\Sigma_{\min}} \tag{H.52}$$

$$\leq \|\beta^{(t)} - \beta^*\|^2_{\Sigma^{-1}_{\min}} \tag{H.53}$$

$$- 2\eta\left(\widehat{R}_{\text{s-hinge}}(\beta^{(t)}) - \widehat{R}_{\text{s-hinge}}(\beta^*)\right) + \eta^2\|\Sigma^{1/2}_{\min} g_t\|^2_2.$$

Unrolling the chain of inequalities as in the typical convergence proof yields the same result, with the difference being that the Lipschitz constant will now be an upper bound on $\|\Sigma^{1/2}_{\min} g_t\|_2$. Here we observe that for all $\Sigma \in \mathcal{C}$ (and therefore for every max player cost choice $\Sigma_t$ at each iteration),

$$\|\Sigma_{\min}^{1/2} g_t\|_2 \le \left\| \Sigma_{\min}^{1/2} \left( x + u_*(1+\lambda) \frac{\partial \|\beta\|_{*,\Sigma_t^{-1}}}{\partial \beta} \right) \right\|_2 \tag{H.54}$$

$$\le \|\Sigma_{\min}^{1/2} x\|_2 + u_*(1+\lambda) \left\| \Sigma_{\min}^{1/2} \frac{\partial \|\Sigma_t^{-1/2}\beta\|_*}{\partial \beta} \right\|_2 \tag{H.55}$$

$$\le X + u_*(1+\lambda) \left\| \Sigma_{\min}^{1/2} \Sigma_t^{-1/2} \frac{\partial \|\beta\|_*}{\partial \beta} \Big|_{\beta = \Sigma_t^{-1/2}\beta} \right\|_2 \tag{H.56}$$

$$\le X + u_*(1+\lambda) \overbrace{\|\Sigma_{\min}^{1/2} \Sigma_t^{-1/2}\|_2}^{\le 1} \left\| \frac{\partial \|\beta\|_*}{\partial \beta} \Big|_{\beta = \Sigma_t^{-1/2}\beta} \right\|_2 \tag{H.57}$$

$$\le X + u_*(1+\lambda) L_* =: L, \tag{H.58}$$

where $L_*$ is the Lipschitz constant of the gradient of the dual norm (recall for $p$-norms this is equal to $\max\left(1, d^{(p-2)/2p}\right)$). Now continuing with the standard proof of convergence of the subgradient method gives us the desired result.

It remains to show that Algorithm 9 successfully identifies the subgradient of $r$. We do so by invoking Lemma 9.4.1, which says that we can efficiently solve for $c^*$ at each iteration. Thus we have that

$$\max_{c \in \mathcal{C}} \widehat{R}^c_{\text{s-hinge}}(\widehat{\beta}) \le \min_\beta \max_{c \in \mathcal{C}} \widehat{R}^c_{\text{s-hinge}}(\beta) + \frac{LB}{\sqrt{T}}, \tag{H.59}$$

noting that this bound is with respect to the minimax *empirical* risk. To complete the bound with respect to the minimax population risk, define $\beta^* := \arg\min_\beta \max_{c \in \mathcal{C}} R^c_{\text{s-hinge}}(\beta)$ as the population minimax solution. Then we have with probability $\ge 1 - \delta/2$

$$\min_\beta \max_{c \in \mathcal{C}} \widehat{R}^c_{\text{s-hinge}}(\beta) \le \max_{c \in \mathcal{C}} \widehat{R}^c_{\text{s-hinge}}(\beta^*) \tag{H.60}$$

$$\le \max_{c \in \mathcal{C}} R^c_{\text{s-hinge}}(\beta^*) + B(X + u_*)\sqrt{\frac{\ln 2/\delta}{2n}} \tag{H.61}$$

$$= \min_\beta \max_{c \in \mathcal{C}} R^c_{\text{s-hinge}}(\beta) + B(X + u_*)\sqrt{\frac{\ln 2/\delta}{2n}}, \tag{H.62}$$

where in the second inequality we've applied Hoeffding's between the empirical and population adversarial risk of $\beta^*$ (which is bounded in $[0, B(X + u_*)]$) because that classifier does not depend on the training data. □

## H.6 Proof for Stochastic Mirror Descent-Ascent

We let $0 < \epsilon \leq 1$ denote the discretization parameter which tunes the size of the set $|\mathcal{S}|$ (for simplicity, assume $1/\epsilon$ is an integer). Specifically, we choose $1/\epsilon$ equally spaced points in each dimension's range of *inverse* eigenvalues and then define the elements of $\mathcal{S}$ to be the collection of smallest values in each dimension, then all the second-smallest values, etc. In this way we discretize the "diagonal" of the cost uncertainty set $\mathcal{C}$ to avoid an exponential dependence on the dimension.

**Theorem H.6.1.** *Suppose we run SMDA on the regularized strategic hinge loss as described in Algorithm 7 for $T$ iterations and get averaged classifier iterates $\tilde{\beta}$. Define the convergence error*

$$\varepsilon_T := \max_{c \in \mathcal{C}} R^c_{\textit{s-hinge}}(\tilde{\beta}) - \min_{\beta} \max_{c \in \mathcal{C}} R^c_{\textit{s-hinge}}(\beta). \tag{H.63}$$

*Then over the randomness of the optimization procedure it holds that*

$$\mathbb{E}[\varepsilon_T] \lesssim B \left[ u_* |1 - \lambda| \max_i \sqrt{\epsilon \left( \sigma_{i\ell}^{-2} - \sigma_{iu}^{-2} \right)} + \frac{L + (B^{-1} + X + u_*) \sqrt{\ln 1/\epsilon}}{\sqrt{T}} \right]. \tag{H.64}$$

*Proof.* From Proposition 9.4.3, we have that the regularized loss is convex in $\beta$. However, the loss is *not* concave in any parameterization of the cost function $c$. To resolve this, we discretize the space of cost functions. We parameterize the eigenvalues of the inverse cost matrix $\Sigma^{-1}$ as a choice of eigenvalue for each eigenvector $v_i$ from the compact set $[\sigma_{iu}^{-2}, \sigma_{i\ell}^{-2}]$. We do so by discretizing the space of choices linearly as $\sigma_k^{-2} = \sigma_{iu}^{-2} + k\epsilon(\sigma_{i\ell}^{-2} - \sigma_{iu}^{-2})$ for $k \in [1, 1/\epsilon]$. Now we can instead optimize

$$\min_{\beta} \max_{k \in [1/\epsilon]} R^{c(k)}_{\text{s-hinge}}(\beta), \tag{H.65}$$

where $R^{c(k)}_{\text{s-hinge}}$ denotes the loss under the cost defined by the eigenvalues induced by $k$. As this is a discrete set of $1/\epsilon$ choices, this objective is exactly equivalent to

$$\min_{\beta} \max_{\delta \in \Delta^{1/\epsilon}} \sum_{i=1}^{1/\epsilon} \delta_i R^{c(i)}_{\text{s-hinge}}(\beta), \tag{H.66}$$

where $\Delta^{1/\epsilon}$ is the simplex over $1/\epsilon$ values such that $\delta_i \geq 0 \; \forall i, \; \sum_i \delta_i = 1$. This objective *is* concave in $\delta$, which means it can be solved via SMDA as described by

Nemirovski et al. [2009]. Let $\tilde{\beta}, \tilde{\delta}$ denote the two players' averaged iterates over choices $\beta, \delta$. Define

$$\widehat{\varepsilon}_T := \max_{k \in [1/\epsilon]} R^{c(k)}_{\text{s-hinge}}(\tilde{\beta}) - \min_{\beta} \sum_{i=1}^{1/\epsilon} \tilde{\delta}_i R^{c(i)}_{\text{s-hinge}}(\beta). \tag{H.67}$$

Note that this is the expected sub-optimality gap for a new optimization problem, whose solution is not the same as the one in the theorem statement. Nemirovski et al. [2009] prove that after $T$ iterations of SMDA with the appropriate step size we have (in their original notation)

$$\mathbb{E}[\widehat{\varepsilon}_T] \leq 2\sqrt{\frac{10[R_x^2 M_{*,x}^2 + M_{*,y}^2 \ln m]}{N}}. \tag{H.68}$$

Translating these terms into our notation,

$$\begin{aligned}
m &= 1/\epsilon, && \text{(H.69)}\\
N &= T, && \text{(H.70)}\\
M_{*,x}^2 &= \max_{1 \leq i \leq 1/\epsilon} \mathbb{E}[\|\nabla \ell^{c(i)}_{\text{s-hinge}}(\tilde{\beta})\|^2] \leq L^2, && \text{(H.71)}\\
M_{*,y}^2 &= \mathbb{E}\left[\max_{1 \leq i \leq 1/\epsilon} |\ell^{c(i)}_{\text{s-hinge}}(\tilde{\beta})|^2\right] \leq (1 + B(X + u_*))^2, && \text{(H.72)}\\
R_x^2 &= \frac{1}{2} \max_{b_1, b_2 \in \mathcal{B}} \|b_1 - b_2\|_2^2 \lesssim B^2, && \text{(H.73)}
\end{aligned}$$

where the last inequality follows from the triangle inequality and the upper bound on $\|\beta\|_2$. Plugging these in gives the bound

$$\begin{aligned}
\mathbb{E}[\widehat{\varepsilon}_T] &\lesssim \sqrt{\frac{B^2 L^2 + (1 + B(X + u_*))^2 \ln 1/\epsilon}{T}} && \text{(H.74)}\\
&\lesssim B\, \frac{L + (B^{-1} + X + u_*)\sqrt{\ln 1/\epsilon}}{\sqrt{T}}. && \text{(H.75)}
\end{aligned}$$

Next, we can rewrite the convergence error in the theorem statement in terms of this error as

$$\varepsilon_T := \max_{c\in\mathcal{C}} R^c_{\text{s-hinge}}(\tilde{\beta}) - \min_{\beta}\max_{c\in\mathcal{C}} R^c_{\text{s-hinge}}(\beta) \tag{H.76}$$

$$= \widehat{\varepsilon}_T + \left(\max_{c\in\mathcal{C}} R^c_{\text{s-hinge}}(\tilde{\beta}) - \max_{k\in[1/\epsilon]} R^{c(k)}_{\text{s-hinge}}(\tilde{\beta})\right) \tag{H.77}$$

$$- \overbrace{\left(\min_{\beta}\max_{c\in\mathcal{C}} R^c_{\text{s-hinge}}(\beta) - \min_{\beta}\sum_{i=1}^{1/\epsilon}\tilde{\delta}_i R^{c(i)}_{\text{s-hinge}}(\beta)\right)}^{\geq 0} \tag{H.78}$$

$$\leq \widehat{\varepsilon}_T + \left(\max_{c\in\mathcal{C}} R^c_{\text{s-hinge}}(\tilde{\beta}) - \max_{k\in[1/\epsilon]} R^{c(k)}_{\text{s-hinge}}(\tilde{\beta})\right). \tag{H.79}$$

This last term represents the error due to discretization. Revisiting the regularized risk definition, for any $c$ and $k$ we have

$$R^c_{\text{s-hinge}}(\tilde{\beta}) - R^{c(k)}_{\text{s-hinge}}(\tilde{\beta}) = \mathbb{E}\Big[\max\{0, 1 - y(\tilde{\beta}^\top x + u_*\|\Sigma_c^{-1/2}\tilde{\beta}\|_*)\} \tag{H.80}$$

$$- \max\{0, 1 - y(\tilde{\beta}^\top x + u_*\|\Sigma_{c(k)}^{-1/2}\tilde{\beta}\|_*)\}\Big] \tag{H.81}$$

$$+ \lambda u_*(\|\Sigma_c^{-1/2}\tilde{\beta}\|_* - \|\Sigma_{c(k)}^{-1/2}\tilde{\beta}\|_*) \tag{H.82}$$

$$\leq u_*|1-\lambda|\left|\|\Sigma_{c(k)}^{-1/2}\tilde{\beta}\|_* - \|\Sigma_c^{-1/2}\tilde{\beta}\|_*\right| \tag{H.83}$$

$$\leq u_*|1-\lambda|B\cdot\sigma_{\max}\left(\Sigma_{c(k)}^{-1/2} - \Sigma_c^{-1/2}\right) \tag{H.84}$$

by the reverse triangle inequality. Since these two matrices have the same eigenvectors, the maximum eigenvalue of their difference is simply the maximum absolute difference between their respective eigenvalues. By construction, for any choice $\Sigma_c^{-1}$, there is a choice $k \in [1/\epsilon]$ which differs in spectrum by no more than $\epsilon\cdot\max_i(\sigma_{i\ell}^{-2} - \sigma_{iu}^{-2})$ in any given direction, and therefore we have

$$\max_{c\in\mathcal{C}} R^c_{\text{s-hinge}}(\tilde{\beta}) - \max_{k\in[1/\epsilon]} R^{c(k)}_{\text{s-hinge}}(\tilde{\beta}) \tag{H.85}$$

$$\leq u_* B\cdot|1-\lambda|\cdot\max_i\left|\sqrt{\sigma_i\left(\Sigma_{c(k)}^{-1}\right)} - \sqrt{\sigma_i\left(\Sigma_c^{-1}\right)}\right| \tag{H.86}$$

$$\leq u_* B\cdot|1-\lambda|\cdot\max_i\sqrt{\left|\sigma_i\left(\Sigma_{c(k)}^{-1}\right) - \sigma_i\left(\Sigma_c^{-1}\right)\right|} \tag{H.87}$$

$$\leq u_* B\cdot|1-\lambda|\cdot\max_i\sqrt{\epsilon\left(\sigma_{i\ell}^{-2} - \sigma_{iu}^{-2}\right)}. \tag{H.88}$$

Combining this bound with the one above on $\widehat{\varepsilon}_T$ and taking expectations gives the result. $\square$

**Corollary H.6.2.** *Recall* $D := \max_i(\sigma_{i\ell}^{-2} - \sigma_{iu}^{-2})$. *Choosing* $\epsilon = \Theta\left(\frac{\ln T}{T \max(1,D)}\right)$, *we have*

$$\mathbb{E}[\varepsilon_T] \lesssim \frac{LB}{\sqrt{T}} + B(X + u_*)\sqrt{\frac{\ln T + \max(0, \ln D)}{T}}. \tag{H.89}$$

## H.7 Goodhart's Law Under Known Costs

We here show that when each user's strategic response to a classifier is known, then in principle (*information theoretically,* discarding optimization concerns), strategic response has no effect on predictive performance.

We start with a correspondence between classifiers operating on (non-strategic) inputs and appropriately modified classifiers operating on strategically modified inputs. In line with prior works—and unlike the other results in this paper—we make the additional assumption that $u$ is fixed and $\phi$ is strictly monotone (thus $u_* = \phi^{-1}(u)$) We also explicitly parameterize the bias term in the classifiers because it plays an important role in the result.

**Proposition H.7.1.** *Let* $f(x) = \mathbf{1}\{\beta^\top x + b \geq 0\}$ *be the prediction of the classifier parameterized by* $(\beta, b)$, *and let* $f'(x) = \mathbf{1}\{\beta^\top x + b' \geq 0\}$ *denote this classifier with a shifted bias* $b' = b - u_*\|\beta\|_*$. *Then for all* $x$, *it holds that* $f(x) = f'(x')$, *where* $x' = x(\beta, b')$ *is the new location of* $x$ *after strategic response to the classifier* $(\beta, b')$.

This results states that $f$ outputs on every data point $x$ exactly what $f'$ (which only differs in its shifted bias term) outputs on $x'$—which represents the exact same "user" *after it has strategically responded.*

*Proof.* The proof is simple. We consider three separate cases:

- If $f(x) = 0$ and $x$ is too far to cross the decision boundary of $f$ (so $x'(\beta, b) = x$), then since $u_*\|\beta\|_* \geq 0$, $x$ is also too far to cross the decision boundary of $f'$, (that is, $x'(\beta, b') = x$). Therefore, $f'(x') = f'(x) = f(x)$.
- If $f(x) = 0$ and $x$ is close enough to cross the decision boundary (so $x'(\beta, b) \neq x$), then Lemma 9.2.3 implies that the maximum amount by which $x$ will change its linear prediction under $\beta$ is $u_*\|\beta\|_*$. Since $f(x) = 0 \implies 0 > \beta^\top x + b \implies -u_*\|\beta\|_* > \beta^\top x + b'$, this means $x$ cannot force $f'(x') = 1$ without increasing the prediction by more than $u_*\|\beta\|_*$, and thus it will not move. So, $f'(x') = f'(x) = f(x)$.

- If $f(x) = 1$, then $0 \leq \beta^\top x + b \implies -u_* \|\beta\|_* \leq \beta^\top x + b'$. Thus, either $x$ will already get a positive classification from $f'$, or it will be able to move enough to cross the decision boundary. Either way, $f'(x') = 1 = f(x)$. $\square$

Note that this proof applies even if we *don't* know the user's cost function—it is sufficient to know for each user the maximum potential increase in their predicted logit under $\beta$ after strategic response, i.e. $\beta^\top(x'(\beta) - x)$. Applying this insight to the optimal classifier gives the following result:

**Corollary H.7.2.** *Fix some distribution $\mathcal{D}$, and let $f^*$ with parameters $(\beta^*, b^*)$ be the classifier minimizing the* non-strategic *0-1 risk on $\mathcal{D}$. Denote this risk as $\alpha$. Then for $f'$ with $(\beta^*, b^* - u_*\|\beta^*\|_*)$, its expected* strategic *0-1 risk on $\mathcal{D}$ is exactly $\alpha$.*

The take-away is that if we are able to minimize the standard 0-1 loss, and we know how users will respond (e.g. by knowing the exact cost function), then a trivial modification would provide us with an optimal classifier for the strategic 0-1 loss. Thus, strategic response does not pose any additional statistical difficulty over standard classification.

Importantly, however, this result does *not* imply that minimizing a *proxy* loss (such as the hinge or logistic loss) on non-strategic data and applying the transformation would give a good strategic classifier; this precisely why the strategic hinge loss is needed in the first place. Further, this result does not account for the social cost of the classifier $f'$ versus some other classifier [Milli et al., 2019]—there could be a different classifier with similar accuracy under strategic response that induces a smaller cost to the users.

# Appendix I

# Appendix for Chapter 10

## I.1 Related Work

**Characterizing the NN loss landscape.** Earlier studies of the loss landscape commonly identified a heavy-tailedness with a small group of very large outlier Hessian eigenvalues or Jacobian singular values [Sagun et al., 2016, 2017, Papyan, 2018, Oymak et al., 2019, Papyan, 2019, Fort and Ganguli, 2019, Ghorbani et al., 2019, Li et al., 2020, Papyan, 2020, Kopitkov and Indelman, 2020]. Later efforts focused on concretely linking these observations to corresponding behavior, often with an emphasis on SGD's bias towards particular solutions [Wu et al., 2018, Jastrzębski et al., 2017, 2020] and what this may imply about its resulting generalization [Jastrzębski et al., 2019, Zhu et al., 2019b, Wu et al., 2022]. Our method for identifying these paired groups, along with Figure 10.2, indicates that these outlier directions in the Hessian/Jacobian spectrum are precisely the directions with opposing signals in the gradient, and that this pattern may be key to better understanding the generalization ability of NNs trained with SGD.

**Progressive sharpening and the edge of stability.** Shifting away from the overall structure, more recent focus has been specifically on top eigenvalue(s), where it was empirically observed that their magnitude (the loss "sharpness") grows when training with SGD [Jastrzębski et al., 2019, 2020] and GD [Kopitkov and Indelman, 2020, Cohen et al., 2021] (so-called "progressive sharpening"). This leads to rapid oscillation in weight space [Xing et al., 2018, Jastrzębski et al., 2019, Cohen et al., 2021, 2022]. Cohen et al. [2021] also found that for GD this coincides with a consistent yet non-monotonic decrease in training loss over long timescales, which they named the "edge of stability"; moreover, they noted that this behavior runs contrary to our traditional understanding of NN convergence. Many works have since investigated the possible origins of this phenomenon [Zhu et al., 2023, Kreisler et al., 2023]. Several of these are deeply related to our findings: Ma et al. [2022] connect this behavior to the existence of multiple “scales” of losses; the outliers we identify corroborate this point. Damian et al. [2022] prove that GD implicitly regularizes the sharpness—we identify a conceptually distinct source of such regularization, as described in Section 10.3. Arora et al. [2022] show under some conditions that the GD trajectory follows a minimum-loss manifold towards lower curvature regions. This is consistent with our findings, and we believe this manifold to be precisely the path which evenly balances the opposing gradients. Wang et al. [2022b] provide another thorough analysis of NN training dynamics at the edge of stability; their demonstrated phases closely align with our own. They further observe that this sharpening coincides with a growth in the norm of the last layer, which was also noted by MacDonald et al. [2023]. Our proposed explanation for the effect of opposing signals offers some insight into this relationship, but further investigation is needed.

## I.2 Examples of Opposing Signals in Text

Punctuation Ordering

```
the EU is \the best war-avoidance mechanism ever
invented["].
the 2008 economic crash and in doing so \triggered a crisis
of rejection["].
I really thought she was going to use another C-word besides
\coward["].
because it was one of the few that still \dry-farmed["].
He describes the taste as \almost minty["].
I did receive several offers to \help out a bit["].
Or \it won't make a difference anyway["].
and that's what they mean by \when complete["].
next big investment bubble to burst is the \carbon
bubble["].
personal bank account was a \genuine donation["].
filibuster what some openly called a \stolen seat["].
-----------------------------------------------------------------
 I used to catch me a few and make pets out of them.["]
\I'm guessing he didn't mean the drinking.["]
If you can't get to it, that doesn't make sense.'  ["]
and boring.  He loved excitement and attention.["]
is playing favorites,' and turned it against him.["]
or medical purposes, absolutely, it's fine.["]
You have to make big decisions in a hurry.["]
I'm really excited about the potential.["]
still believe that it was the right thing to do.["]
Compliance was low on the list, but I think it's a pretty
comfortable bike.["]
the opportunity to do that.'  I just needed to take it and
run with it.["]
```

Figure I.1: **Examples of opposing signals in text.** Found by training GPT-2 on a subset of OpenWebText. Sequences are on separate lines, the token in brackets is the target and all prior tokens are (the end of the) context. As both standards are used, it is not always clear whether punctuation will come before or after the end of a quotation (we include the period after the quote for clarity—the model does not condition on it). Note that the double quotation is encoded as the *pair* of tokens `[447, 251]`, and the loss oscillation is occurring for sequences that end with this pair, either before (top) or after (bottom) the occurrence of the period token (`13`).

**New Line or 'the' After Colon**

```
prepare your data, there are three things to do:[\n]
namely Scalaz or Cats.  It looks like this:[\n]
it will only accept one implementation:[\n]
Salcedo said of the work:[\n]
Enter your email address:[\n]
According to the CBO update:[\n]
Here's how the Giants can still make the playoffs:[\n]
5 reasons as to why self diagnosis is valid:[\n]
successive Lambda invocations.  It looks more or less like
this:[\n]
data, there are three things to do:[\n]
4.2 percent in early 2018.\n\nAccording to the CBO
update:[\n]
other than me being myself."\n\nWATCH:[\n]
is to make the entire construction plural.\n\nTwo recent
examples:[\n]
We offer the following talking points to anyone who is
attending the meeting:[\n]
is on the chopping block - and at the worst possible
moment:[\n]
-------------------------------------------------------------------
our MPs in Westminster.  But to me it is obvious:  [the]
The wheelset is the same as that on the model above:  [the]
not get so engrained or in a rut with what I had been doing.
Not to worry:  [the]
polemics against religion return in various ways to one core
issue:  [the]
which undergirds all other acts of love, both divine and
human:  [the]
integrate fighters from the Kurds' two main political
parties:  [the]
robs this incredible title of precisely what makes it so
wonderful:  [the]
you no doubt noticed something was missing:  [the]
Neil Gorsuch's 'sexist' comments on maternity leave:  [the]
```

Figure I.2: **Examples of opposing signals in text.** Found by training GPT-2 on a subset of OpenWebText. Sequences are on separate lines, the token in brackets is the target and all prior tokens are (the end of the) context. Sometimes a colon occurs mid-sentence—and is often followed by "the"—other times it announces the start of a new line. The model must *unlearn* ": $\mapsto$ `[\n]`" versus ": $\mapsto$ `[the]`" and instead use other contextual information.

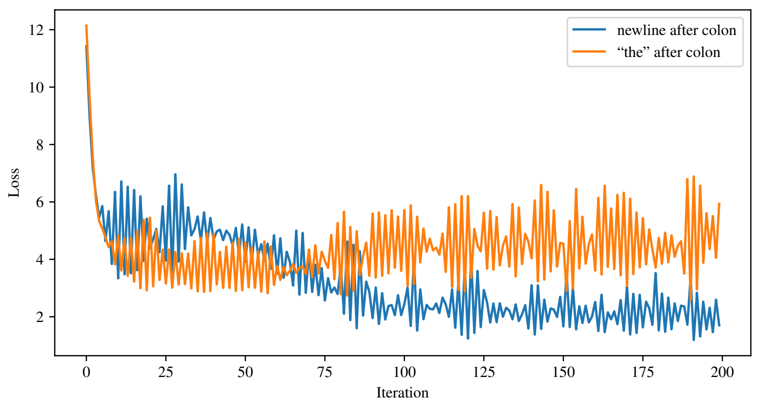


Figure I.3: Loss of GPT-2 on the above opposing signals.

## I.3 Reproducing Figure 10.5 in Other Settings

Though colors are straightforward, for some opposing signals such as grass texture it is not clear how to produce a synthetic image which properly captures what precisely the model is latching on to. Instead, we identify a real image which has as much grass and as little else as possible, with the understanding that the additional signal in the image could affect the results. We depict the grass image alongside the plots it produced.

### I.3.1 ResNet-18 Trained with GD on Other Inputs

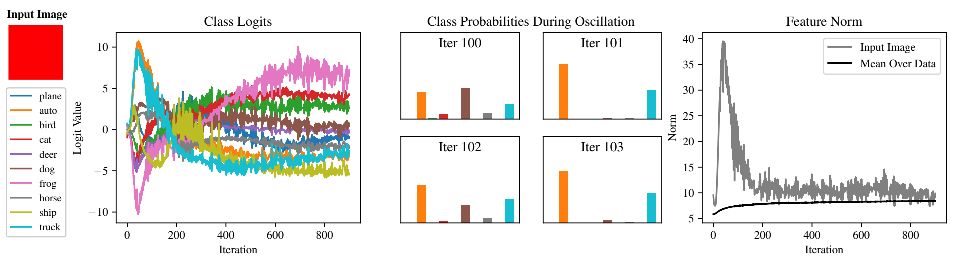


Figure I.4: ResNet-18 on a red color block.

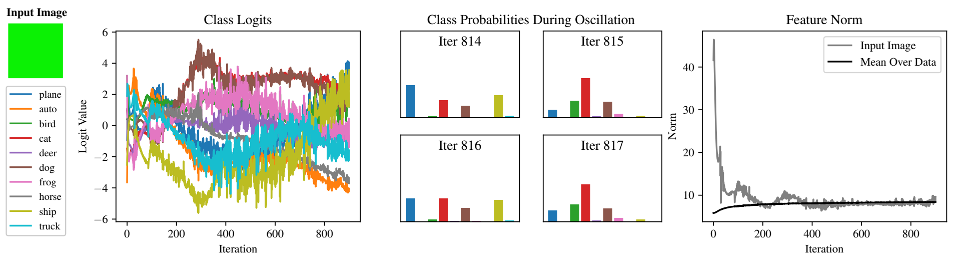


Figure I.5: ResNet-18 on a green color block. As this color seems unnatural, we've included two examples of relevant images in the dataset.

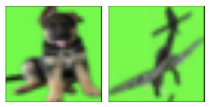


Figure I.6: Examples of images with the above green color.

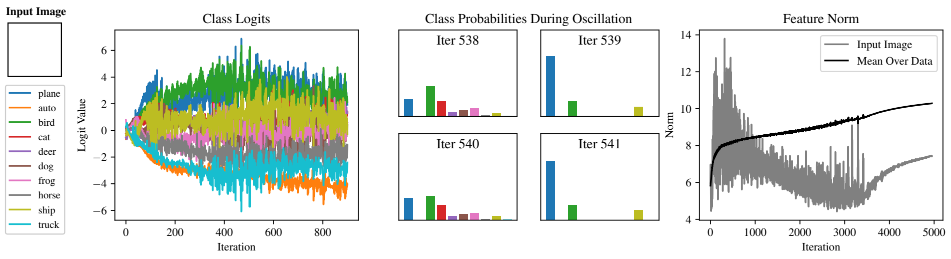


Figure I.7: ResNet-18 on a white color block.

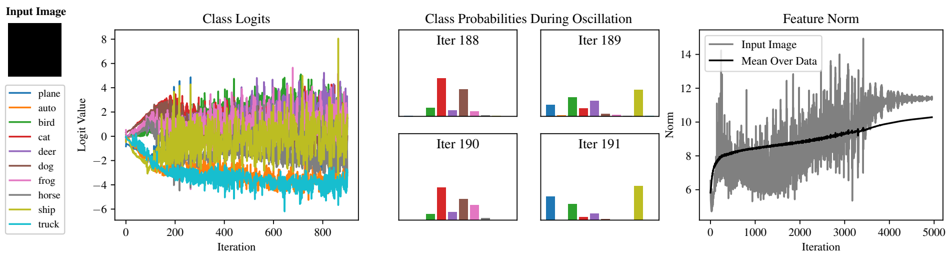


Figure I.8: ResNet-18 on a black color block.

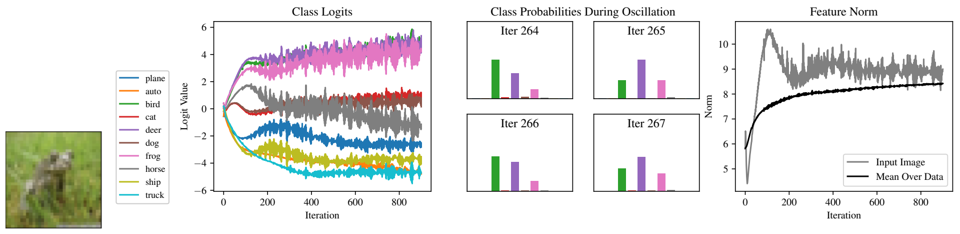


Figure I.9: ResNet-18 on an image with mostly grass texture.

### I.3.2 VGG-11-BN Trained with GD

For VGG-11, we found that the feature norm of the embedded images did not decay nearly as much over the course of training. We expect this has to do with the lack of a residual component. However, for the most part these features do still follow the pattern of a rapid increase, followed by a marked decline.

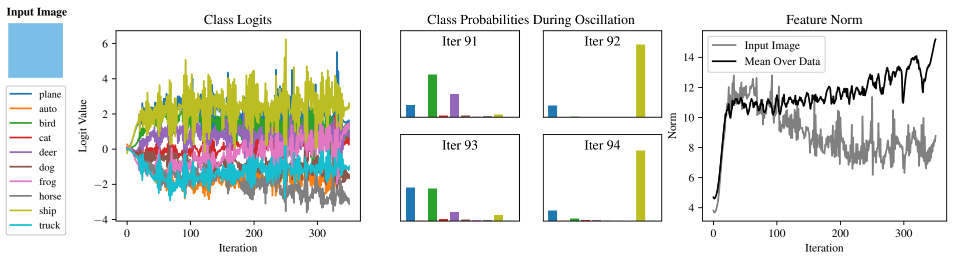


Figure I.10: VGG-11-BN on a sky color block.

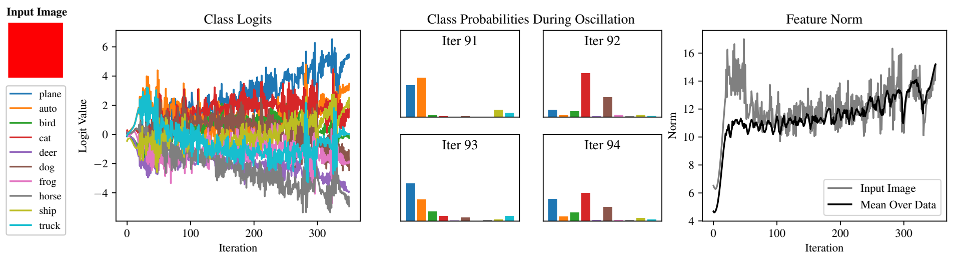


Figure I.11: VGG-11-BN on a red color block.

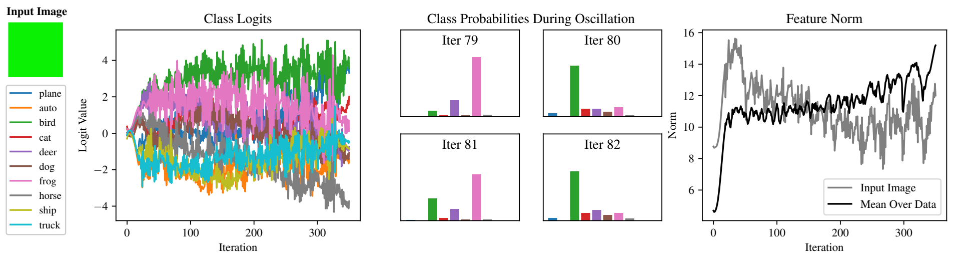


Figure I.12: VGG-11-BN on a green color block. See above for two examples of relevant images in the dataset.

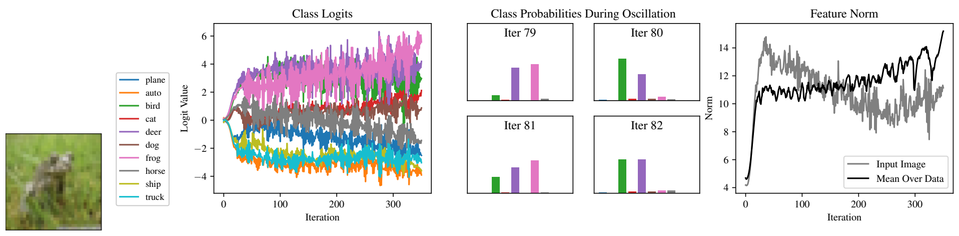


Figure I.13: VGG-11-BN on an image with mostly grass texture.

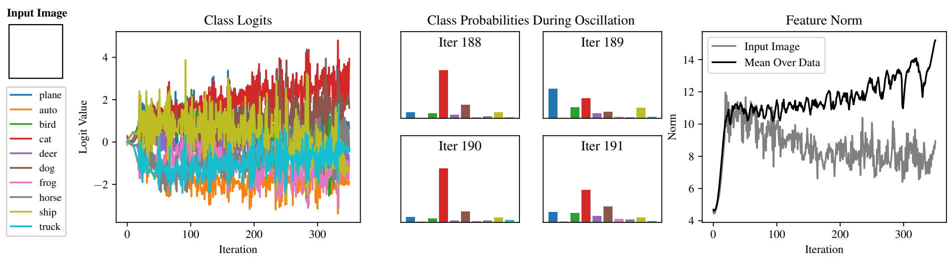


Figure I.14: VGG-11-BN on a white color block.

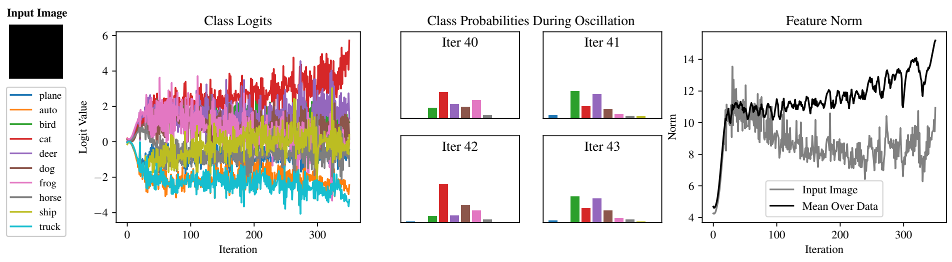


Figure I.15: VGG-11-BN on a black color block.

### I.3.3 VGG-11-BN with Small Learning Rate to Approximate Gradient Flow

Here we see that oscillation is a valuable regularizer, preventing the network from continuously upweighting opposing signals. As described in the main body, stepping too far in one direction causes an imbalanced gradient between the two opposing signals. Since the group which now has a larger loss is also the one which suffers from the use of the feature, the network is encouraged to downweight its influence. If we use a very small learning rate to approximate gradient flow, this regularization does not occur and the feature norms grow continuously. This leads to over-reliance on these features, suggesting that failing to downweight opposing signals is a likely cause of the poor generalization of networks trained with gradient flow.

The following plots depict a VGG-11-BN trained with learning rate $.0005$ to closely approximate gradient flow. We compare this to the feature norms of the same network trained with gradient descent with learning rate $0.1$, which closely matches gradient flow until it becomes unstable.

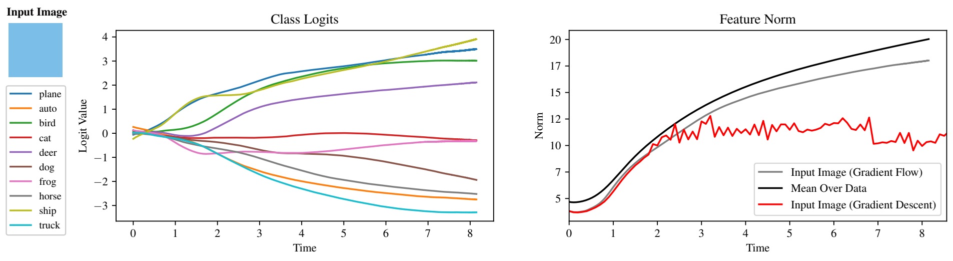


Figure I.16: VGG-11-BN on a sky color block with learning rate 0.005 (approximating gradient flow) compared to 0.1.

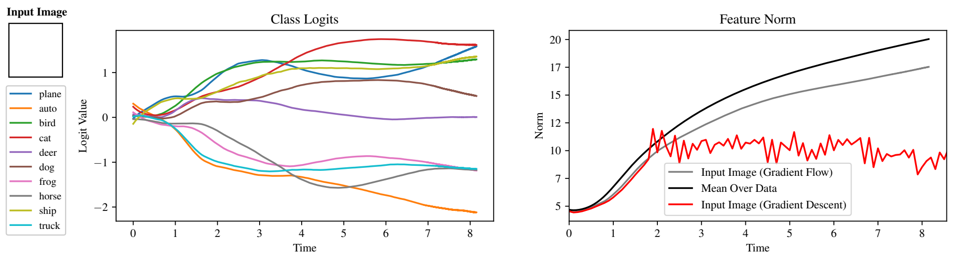


Figure I.17: VGG-11-BN on a white color block with learning rate 0.005 (approximating gradient flow) compared to 0.1.

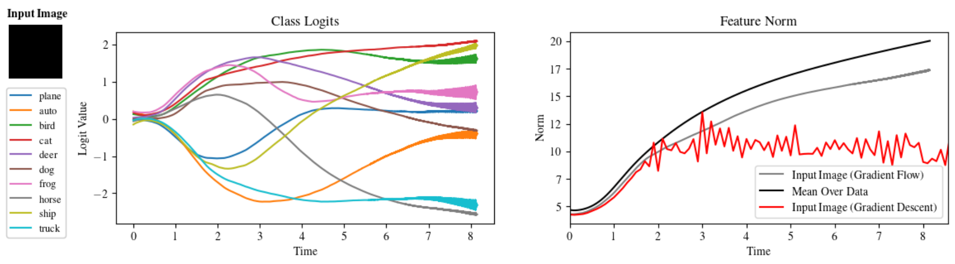


Figure I.18: VGG-11-BN on a black color block with learning rate 0.005 (approximating gradient flow) compared to 0.1.

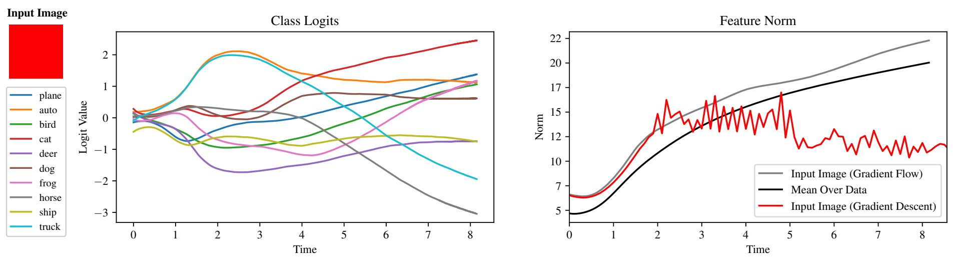


Figure I.19: VGG-11-BN on a red color block with learning rate 0.005 (approximating gradient flow) compared to 0.1.

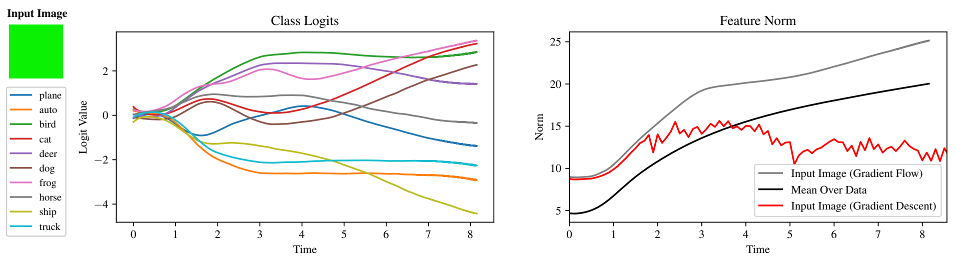


Figure I.20: VGG-11-BN on a green color block with learning rate 0.0005 (approximating gradient flow) compared to 0.1.

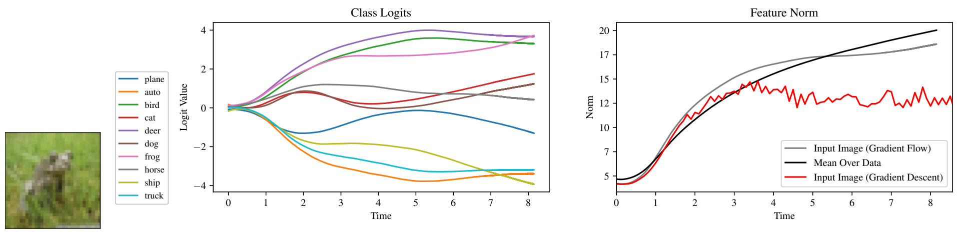


Figure I.21: VGG-11-BN on an image with mostly grass texture with learning rate 0.0005 (approximating gradient flow) compared to 0.1.

### I.3.4 ResNet-18 Trained with Full-Batch Adam

Finally, we plot the same figures for a ResNet-18 trained with full-batch Adam. We see that Adam consistently and quickly reduces the norm of these features, especially for more complex features such as texture, and that it also quickly reaches a point where oscillation ends. Note when comparing to plots above that the maximum iteration on the x-axis differs.

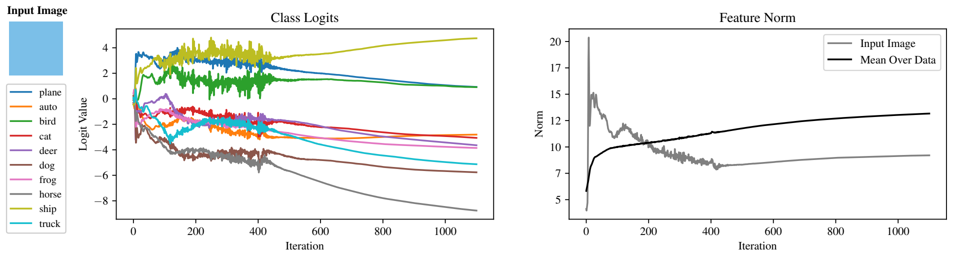


Figure I.22: ResNet-18 on a sky color block trained with Adam.

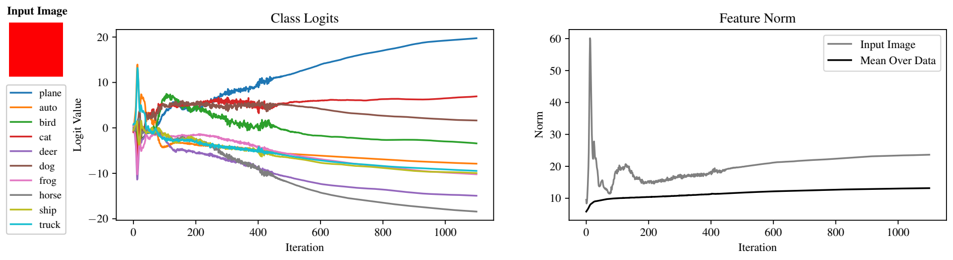


Figure I.23: ResNet-18 on a red color block trained with Adam.

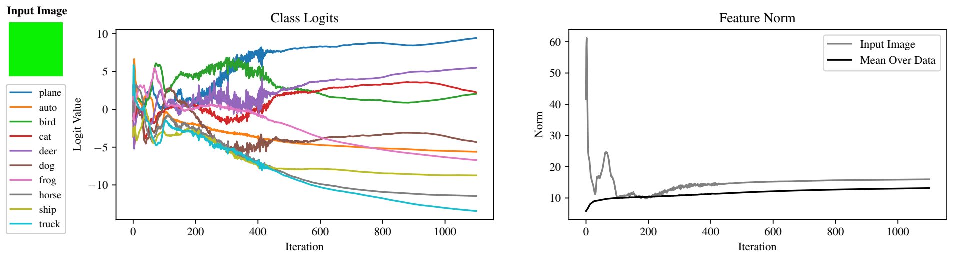


Figure I.24: ResNet-18 on a green color block trained with Adam.

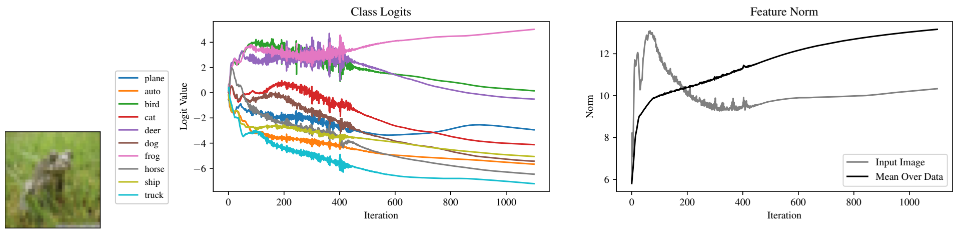


Figure I.25: ResNet-18 on an image with mostly grass texture trained with Adam.

## I.4 Tracking the Amount of Curvature in each Parameter Layer

Here we plot the “fraction of curvature” of different architectures at various training steps. Recall the fraction of curvature is defined with respect to the top eigenvector of the loss Hessian. We partition this vector by network layer and evaluate each sub-vector’s squared norm. This represents that layer’s contribution to the overall curvature. To keep the plots readable, we omit layers whose fraction is never greater than $0.01$ at any training step (including the intermediate ones not plotted), though we always include the last linear layer. The total number of layers is 45, 38, 106, and 39 for the ResNet, VGG-11, ViT, and NanoGPT respectively.

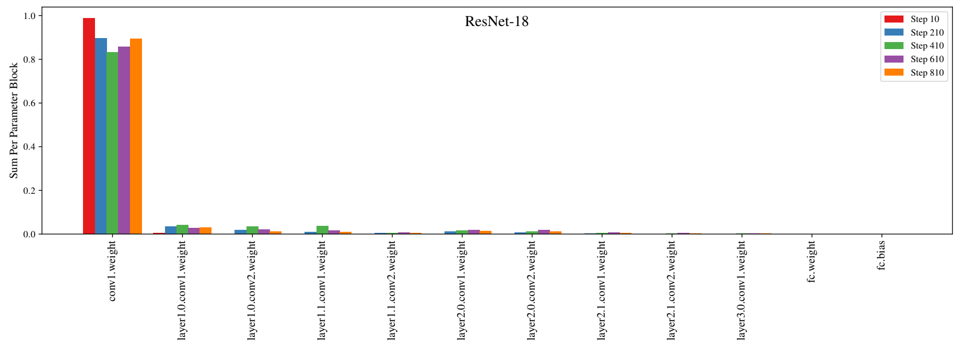


Figure I.26: Sum of squared entries of the top eigenvector of the loss Hessian which lie in each parameter layer of a ResNet-18 throughout training.

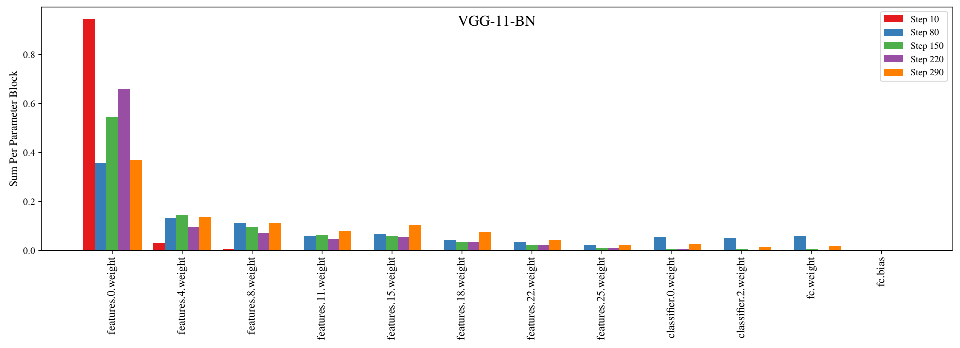


Figure I.27: Sum of squared entries of the top eigenvector of the loss Hessian which lie in each parameter layer of a VGG-11-BN throughout training.

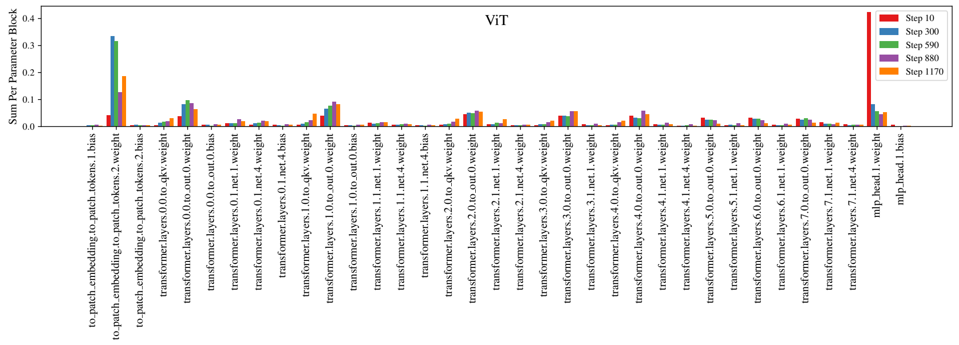


Figure I.28: Sum of squared entries of the top eigenvector of the loss Hessian which lie in each parameter layer of a ViT throughout training.

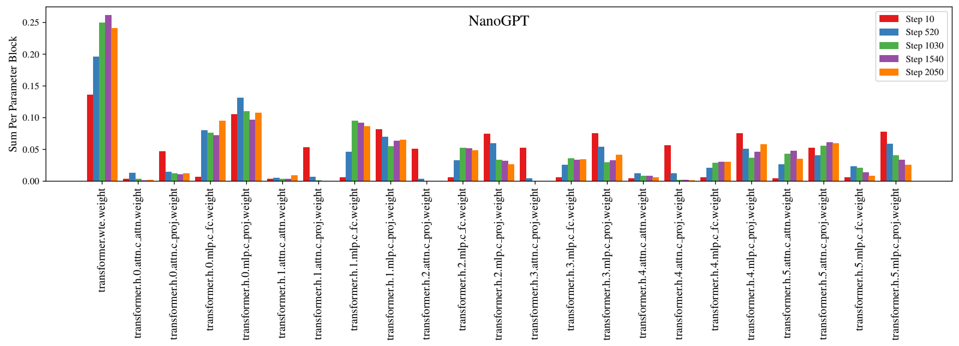


Figure I.29: NanoGPT (6 layers, 6 head per layer, embedding dimension 384) trained on the default shakespeare character dataset in the NanoGPT repository. Due to difficulty calculating the true top eigenvector, we approximate it with the exponential moving average of the squared gradient.

## I.5 Additional Figures

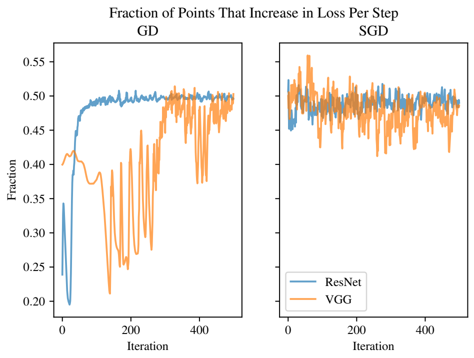


Figure I.30: The fraction of overall training points which increase in loss on any given step. For both SGD and GD, it hovers around 0.5 (VGG without batchnorm takes a long time to reach the edge of stability). Though the outliers have much higher amplitude in their loss change, many more images contain *some* small component of the features they exemplify (or are otherwise slightly affected by the weight oscillations), and so these points also oscillate in loss at a smaller scale.

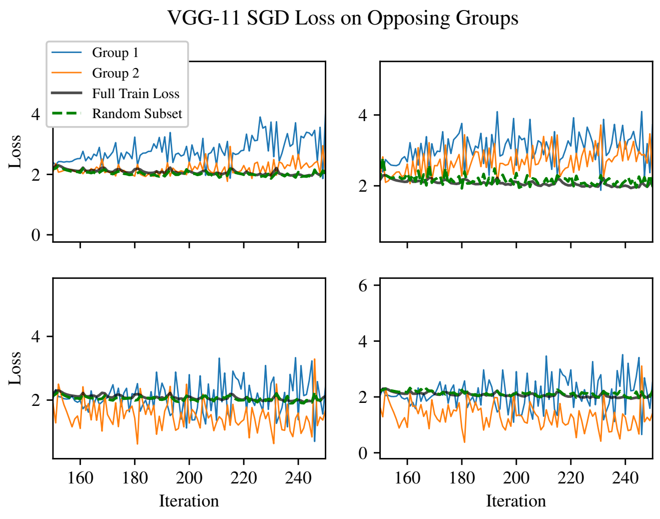
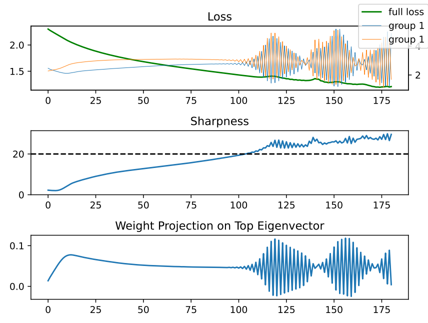
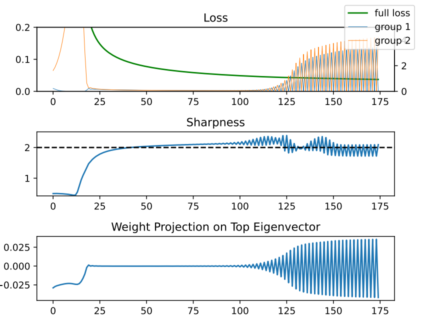


Figure I.31: We reproduce Figure 10.6a without batch normalization.



*(a)* A 3-layer ReLU MLP trained on a 5k-subset of CIFAR-10.



*(b)* Our model: a 2-layer linear network trained on mostly Gaussian data with opposing signals.

Figure I.32: We compare a small ReLU MLP on a subset of CIFAR-10 to our simple model of linear regression with a two-layer network.

## I.6 Comparing our Variant of SGD to Adam

**Algorithm 11** SplitSGD

```
**input:** Initial parameters $\theta_0$, SGD step size $\eta_1$, SignSGD step size $\eta_2$, momentum $\beta$, dampening $\tau$, threshold $r$.
**initialize:** $m_0 = \mathbf{0}$.
**for** $t \leftarrow 1, \dots, T$ **do**
    $g_t \leftarrow \nabla_\theta L_t(\theta_{t-1})$ {Get stochastic gradient}
    $m_t \leftarrow \beta m_{t-1} + (1-\tau) g_t$ {Update momentum with dampening}
    $\widehat{m}_t \leftarrow m_t / (1-\tau^t)$ {Debias}
    $v_{\text{mask}} \leftarrow \mathbf{1}\{|\widehat{m}_t| \leq r\}$ {Split parameters by threshold}
    $\theta_t \leftarrow \theta_{t-1} - \eta_1(\widehat{m}_t \odot v_{\text{mask}}) - \eta_2(\text{sign}(\widehat{m}_t) \odot (1 - v_{\text{mask}}))$ {unmasked SGD, masked SignSGD}
**end for**
```

As described in the main text, we find that simply including dampening and taking a fixed step size on gradients above a certain threshold results in performance matching that of Adam for the experiments we tried. We found that setting this threshold equal to the $q = .1$ quantile of the first gradient worked quite well—this was about `1e-4` for the ResNet-56/110 and `1e-6` for GPT-2.

Simply to have something to label it with, we name the method SplitSGD, because it performs SGD and SignSGD on different partitions of the parameters. The precise method is given above in Algorithm 11. We reiterate that we are not trying to suggest a new method—our goal is only to demonstrate the insight gained from knowledge of opposing signals' influence on NN optimization. For all plots, $\beta$ represents the momentum parameter and $\tau$ is dampening. Adam has a single parameter $\beta_1$ which represents both simultaneously, which we fix at 0.9, and we do the same for SplitSGD by setting $\beta = \tau = 0.9$. As in Algorithm 11, we let $\eta_1$ refer to the learning rate for standard SGD on the parameters with gradient below the magnitude threshold, and $\eta_2$ refers to the learning rate for the remainder which are optimized with SignSGD.

### I.6.1 SplitSGD on ResNet

We begin with a comparison on ResNets trained on CIFAR-10. Figure I.33 compares SplitSGD to standard versions of SGD with varying momentum and dampening on a ResNet-56. As expected, SGD is extremely sensitive to hyperparameters, particularly the learning rate, and even the best choice in a grid search underperforms SplitSGD. Furthermore, the rightmost plot depicts the fraction of

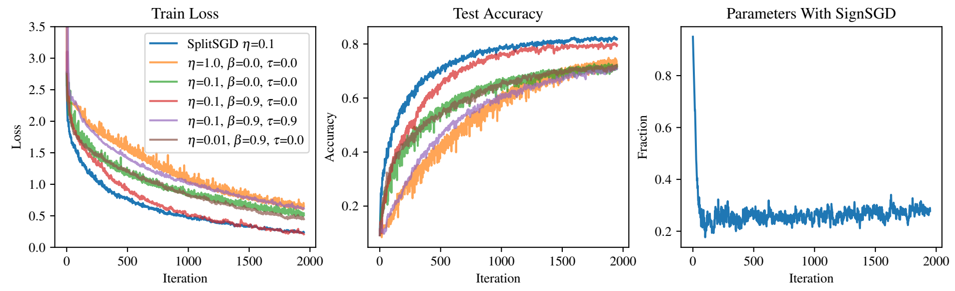


Figure I.33: Standard SGD with varying learning rates and momentum/dampening parameters on a ResNet-56 on CIFAR-10, with one run of SplitSGD for comparison. Omitted SGD hyperparameter combinations performed much worse. Notice that SGD is extremely sensitive to hyperparameters. Rightmost plot is the fraction of parameters with fixed step size by SplitSGD.

parameters for which SplitSGD takes a fixed-size signed step. This means that after the first few training steps, 70-80% of the parameters are being optimized simply with standard SGD (with $\beta = \tau = 0.9$).

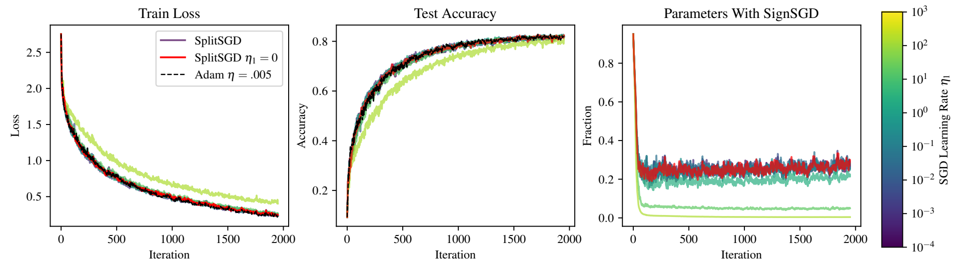


Figure I.34: SplitSGD with varying SGD learning rates $\eta_1$ versus Adam on a ResNet-56 on CIFAR-10. The SignSGD learning rate is fixed at $\eta_2 = .001$; Adam uses $\eta = .005$, which was found to be the best performing choice via oracle selection grid search. The rightmost plot is the fraction of parameters with fixed step size by SplitSGD—that is, 1 minus this value is the fraction of parameters taking a regular gradient step with step size as given in the legend. This learning rate ranges over several orders of magnitude, is used for ˜70-80% of parameters, and can even be set to 0, with no discernible difference in performance.

Next, Figure I.34 plots SplitSGD with varying $\eta_1$ and $\eta_2$ fixed at .001. This is compared to Adam with learning rate .005, which was chosen via oracle grid search. Even though the SGD learning rate $\eta_1$ ranges over *seven orders of magnitude* and is used for ˜70-80% of parameters, we see no real difference in the train loss or test accuracy of SplitSGD. In fact, we find that we can even eliminate it completely! This suggests that for most of parameters and most of training, it is only a small fraction of parameters in the entire network which are influencing the overall performance. We posit a deeper connection here to the “hidden” progress described in grokking [Barak et al., 2022, Nanda et al., 2023]—if the correct subnetwork and its influence on the output grows slowly during training, that behavior will not be noticeable until the dominating signals are first downweighted.

Figures I.35 and I.36 depict the train loss and test accuracy of Adam and SplitSGD for varying learning rates (the standard SGD learning rate $\eta_1$ is fixed at $0.1$). We see that SplitSGD is at least as robust as Adam to learning rate choice, if not more. The results also suggest that SplitSGD benefits from a slightly smaller learning rate than Adam, which we attribute to the fact that it will *always* take step sizes of that fixed size, whereas the learning rate for Adam represents an upper bound on the step size for each parameter.

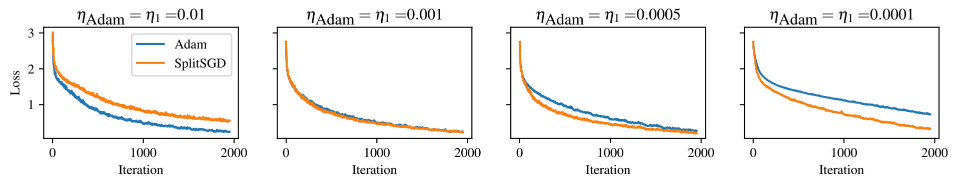


Figure I.35: Train loss of Adam and SplitSGD for varying learning rates. The regular SGD step size for SplitSGD is fixed at 0.1. SplitSGD seems at least as robust to choice of learning rate as Adam, and it appears to benefit from a slightly smaller learning rate because it cannot adjust per-parameter.

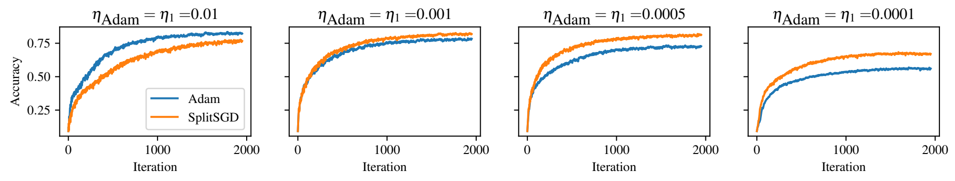


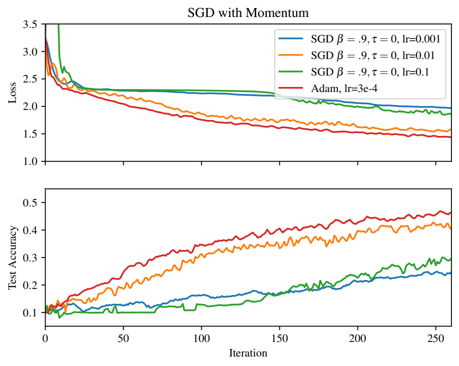
Figure I.36: Test accuracy of Adam and SplitSGD for varying learning rates. The regular SGD step size for SplitSGD is fixed at 0.1. SplitSGD seems at least as robust to choice of learning rate as Adam, and it appears to benefit from a slightly smaller learning rate because it cannot adjust per-parameter.

We repeat these experiments with a ResNet-110, with similar findings. Figure I.37a compares the train loss and test accuracy of SGD with $\beta = 0.9, \tau = 0$ to Adam, and again the sensitivity of this optimizer to learning rate is clear. Figure I.37b compares Adam to SplitSGD (both with fixed-step learning rate .0003) but ablates the use of dampening: we find that the fixed-size signed steps appear to be more important for early in training, while dampening is helpful for maintaining performance later. It is not immediately clear what causes this bifurcation, nor if it will necessarily transfer to attention models.

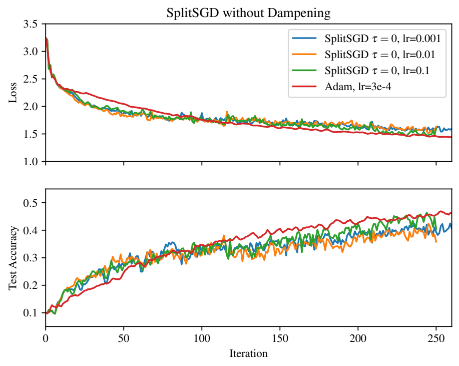
Finally, Figure I.38a compares Adam to the full version of SplitSGD; we see essentially the same performance, and furthermore SplitSGD maintains its robustness to the choice of standard SGD learning rate.

### I.6.2 SplitSGD on GPT-2

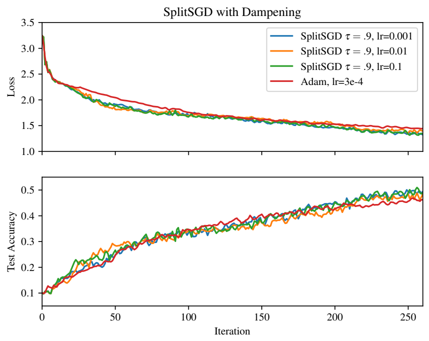
For the transformer, we use the public nanoGPT repository which trains GPT-2 on the OpenWebText dataset. As a full training run would be too expensive, we compare only for the early stage of optimization. All hyperparameters are the defaults from that repository, with the SGD learning rate $\eta_1$ set equal to the other learning rate $\eta_2$. We observe that not only do the two methods track each other closely in training loss, it appears that they experience *exactly* the same oscillations. Though we do not track the parameters themselves, this suggests that these two methods follow very similar optimization trajectories as well, which we believe is an intriguing possibility worth further study.



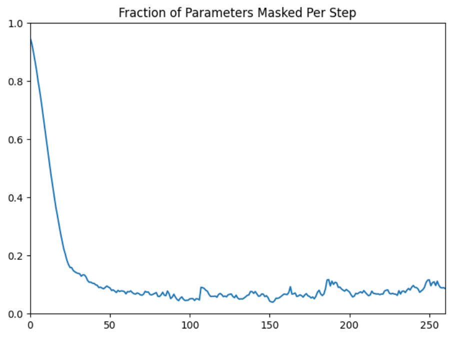
*(a)* Adam versus standard SGD with Momentum. SGD remains extremely sensitive to choice of learning rate.



*(b)* Adam vs. SplitSGD with $\tau = 0$. Fixed-size learning rate for both is .0003.



*(a)* Adam vs. SplitSGD with $\tau = 0.9$. Fixed-size learning rate for both is .0003.



*(b)* The fraction of parameters for which a fixed-size signed step was taken for each gradient step.

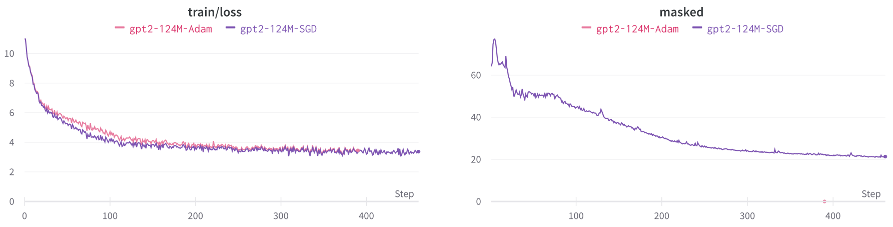


Figure I.39: Adam versus SplitSGD on the initial stage of training GPT-2 on the OpenWebText dataset, and the fraction of parameters with a fixed-size signed step. All hyperparameters are the defaults from the nanoGPT repository. Observe that not only is their performance similar, they appear to have *exactly* the same loss oscillations.

## I.7 Proofs of Theoretical Results

Before we begin the analysis, we must identify the quantities of interest during gradient flow and the system of equations that determines how they evolve.

We start by writing out the loss:

$$\begin{aligned}
2L(\theta) &= \mathbb{E}[(c(b^\top x + b_o^\top x_o) - (\beta^\top x + d_2^{-1/2}\mathbf{1}^\top |x_o|))^2] && \text{(I.1)}\\
&= \mathbb{E}[((cb-\beta)^\top x)^2] + \mathbb{E}[((cb_o - d_2^{-1/2}\operatorname{sign}(x_o)\mathbf{1})^\top x_o)^2] && \text{(I.2)}\\
&= \|cb-\beta\|^2 + \frac{p}{2}\left(\left(\sqrt{\frac{\alpha}{p}}(cb_o - 1)\right)^2 + \left(\sqrt{\frac{\alpha}{p}}(cb_o+1)\right)^2\right) && \text{(I.3)}\\
&= \|cb-\beta\|^2 + \alpha(c^2\|b_o\|^2 + 1). && \text{(I.4)}
\end{aligned}$$

This provides the gradients

$$\begin{aligned}
\nabla_b L &= c(cb-\beta), && \text{(I.5)}\\
\nabla_{b_o} L &= \alpha c^2 b_o, && \text{(I.6)}\\
\nabla_c L &= b^\top(cb-\beta) + \alpha\|b_o\|^2 c. && \text{(I.7)}
\end{aligned}$$

We will also make use of the Hessian to identify its top eigenvalue; it is given by

$$\nabla_\theta^2 L(\theta) = \begin{bmatrix} c^2 I_{d_1} & \mathbf{0}_{d_1\times d_2} & 2cb \\ \mathbf{0}_{d_2\times d_1} & \alpha c^2 I_{d_2} & 2c\alpha b_o \\ 2cb^\top & 2c\alpha b_o^\top & \|b\|^2 + \alpha\|b_o\|^2 \end{bmatrix}. \qquad \text{(I.8)}$$

The maximum eigenvalue $\lambda_{\max}$ at initialization is upper bounded by the maximum row sum of this matrix, and thus $\lambda_{\max} \le 3\frac{d_1+\alpha d_2}{d_1+d_2} < 3\alpha$. Clearly, we also have $\lambda_{\max} \ge \alpha$.

We observe that tracking the precise vectors $b, b_o$ are not necessary to uncover the dynamics when optimizing this loss. First, let us write $b := \epsilon\frac{\beta}{\|\beta\|} + \delta v$, where $v$ is the direction of the rejection of $b$ from $\beta$ (i.e., $\beta^\top v = 0$) and $\delta$ is its norm. Then we have the gradients

$$\nabla_\epsilon L = (\nabla_\epsilon b)^\top (\nabla_b L) \tag{I.9}$$
$$= \frac{\beta}{\|\beta\|}^\top \left( c^2 \left( \epsilon \frac{\beta}{\|\beta\|} + \delta v \right) - c\beta \right) \tag{I.10}$$
$$= c^2 \epsilon - c\|\beta\|, \tag{I.11}$$
$$\nabla_\delta L = (\nabla_\delta b)^\top (\nabla_b L) \tag{I.12}$$
$$= v^\top \left( c^2 \left( \epsilon \frac{\beta}{\|\beta\|} + \delta v \right) - c\beta \right) \tag{I.13}$$
$$= c^2 \delta, \tag{I.14}$$
$$\nabla_c L = \left( \epsilon \frac{\beta}{\|\beta\|} + \delta v \right)^\top \left( c \left( \epsilon \frac{\beta}{\|\beta\|} + \delta v \right) - \beta \right) + \alpha \|b_o\|^2 c \tag{I.15}$$
$$= c(\epsilon^2 + \delta^2 + \alpha \|b_o\|^2) - \epsilon \|\beta\|. \tag{I.16}$$

Finally, define the scalar quantity $o := \|b_o\|^2$, noting that $\nabla_o L = 2b_o^\top \nabla_{b_o} L = 2\alpha c^2 o$. Minimizing this loss via gradient flow is therefore characterized by the following ODE on four scalars:

$$\frac{d\epsilon}{dt} = -c^2 \epsilon + c\|\beta\|, \tag{I.17}$$
$$\frac{d\delta}{dt} = -c^2 \delta, \tag{I.18}$$
$$\frac{do}{dt} = -2\alpha c^2 o, \tag{I.19}$$
$$\frac{dc}{dt} = -c(\epsilon^2 + \delta^2 + \alpha o) + \epsilon \|\beta\|. \tag{I.20}$$
$$\tag{I.21}$$

Furthermore, we have the boundary conditions

$$\epsilon(0) = \sqrt{\frac{1}{d_1 + d_2}}, \tag{I.22}$$
$$\delta(0) = \sqrt{\frac{d_1 - 1}{d_1 + d_2}}, \tag{I.23}$$
$$o(0) = \frac{d_2}{d_1 + d_2}, \tag{I.24}$$
$$c(0) = 1. \tag{I.25}$$

Given these initializations and dynamics, we make a few observations: (i) all four scalars are initialized at a value greater than 0, and remain greater than 0 at all time steps; (ii) $\delta$ and $o$ will decrease towards 0 monotonically, and $\epsilon$ will increase monotonically until $c\epsilon = \|\beta\|$; (iii) $c$ will be decreasing at initialization. Lastly, for conciseness later on we define the quantities

$$r := (\epsilon(0)^2 + \delta(0)^2 + \alpha o(0)) = \frac{d_1 + \alpha d_2}{d_1 + d_2}, \tag{I.26}$$

$$k := \frac{d_2}{d_1}, \tag{I.27}$$

$$m := \frac{d_1}{d_1 + d_2} = \frac{1}{1+k}. \tag{I.28}$$

Before we can prove the main results, we present a lemma which serves as a key tool for deriving continuously valid bounds on the scalars we analyze:

**Lemma I.7.1.** *Consider a vector valued ODE with scalar indices $v_1, v_2, \ldots$, where each index is described over the time interval $[t_{\min}, t_{\max}]$ by the continuous dynamics $\frac{dv_i(t)}{dt} = a_i(v_{-i}(t)) \cdot v_i(t) + b_i(v_{-i}(t))$ with $a_i \leq 0, b_i \geq 0$ for all $i, t$ ($v_{-i}$ denotes the vector $v$ without index $i$). That is, each scalar's gradient is an affine function of that scalar with a negative coefficient. Suppose we define continuous functions $\widehat{a}_i, \widehat{b}_i : \mathbb{R} \to \mathbb{R}$ such that $\forall i, t$, $\widehat{a}_i(t) \leq a_i(v_{-i}(t))$ and $\widehat{b}_i(t) \leq b_i(v_{-i}(t))$. Let $\widehat{v}$ be the vector described by these alternate dynamics, with the boundary condition $\widehat{v}_i(t_{\min}) = v_i(t_{\min})$ and $v_i(t_{\min}) \geq 0$ for all $i$ (if a solution exists). Then for $t \in [t_{\min}, t_{\max}]$ it holds that*

$$\widehat{v}(t) \leq v(t), \tag{I.29}$$

*elementwise. If $\widehat{a}_i, \widehat{b}_i$ upper bound $a_i, b_i$, the inequality is reversed.*

*Proof.* Define the vector $w(t) := \widehat{v}(t) - v(t)$. This vector has the dynamics

$$\begin{aligned} \frac{dw_i}{dt} &= \frac{d\widehat{v}_i}{dt} - \frac{dv_i}{dt} && \text{(I.30)} \\ &= \widehat{a}_i(t) \cdot \widehat{v}_i(t) + \widehat{b}_i(t) - a_i(v_{-i}(t)) \cdot v_i(t) - b_i(v_{-i}(t)) && \text{(I.31)} \\ &\leq \widehat{a}_i(t) \cdot \widehat{v}_i(t) - a_i(v_{-i}(t)) \cdot v_i(t). && \text{(I.32)} \end{aligned}$$

The result will follow by showing that $w(t) \leq \mathbf{0}$ for all $t \in [t_{\min}, t_{\max}]$ (this clearly holds at $t_{\min}$). Assume for the sake of contradiction there exists a time $t' \in (t_{\min}, t_{\max}]$ and index $i$ such that $w_i(t') > 0$ (let $i$ be the first such index for which this occurs, breaking ties arbitrarily). By continuity, we can define $t_0 := \max\,\{t \in [t_{\min}, t'] \;:\; w_i(t) \leq 0\}$. By definition of $t_0$ it holds that $w_i(t_0) = 0$ and $\forall \epsilon > 0$, $w_i(t_0 + \epsilon) - w_i(t_0) = w_i(t_0 + \epsilon) > 0$, and thus $\frac{dw_i(t_0)}{dt} > 0$. But by the definition of $w$ we also have

$$\begin{aligned}\widehat{v}_i(t_0) &= v_i(t_0) + w_i(t_0) && \text{(I.33)}\\ &= v_i(t_0), && \text{(I.34)}\end{aligned}$$

and therefore

$$\begin{aligned}\frac{dw_i(t_0)}{dt} &\leq \widehat{a}_i(t_0) \cdot \widehat{v}_i(t_0) - a_i(v_{-i}(t_0)) \cdot v_i(t_0) && \text{(I.35)}\\ &= \big(\widehat{a}_i(t_0) - a_i(v_{-i}(t_0))\big) \cdot v_i(t_0) && \text{(I.36)}\\ &\leq 0, && \text{(I.37)}\end{aligned}$$

with the last inequality following because $\widehat{a}_i(t) \leq a_i(v_{-i}(t))$ and $v_i(t) > 0$ for all $i, t \in [t_{\min}, t_{\max}]$. Having proven both $\frac{dw_i(t_0)}{dt} > 0$ and $\frac{dw_i(t_0)}{dt} \leq 0$, we conclude that no such $t'$ can exist. The other direction follows by analogous argument. $\square$

We make use of this lemma repeatedly and its application is clear so we invoke it without direct reference. We are now ready to prove the main results:

### I.7.1 Proof of Theorem 10.3.1

*Proof.* At initialization, we have $\|\beta\| \geq \frac{d_1}{\sqrt{d_1+d_2}} \implies \|\beta\|\epsilon(0) \geq \frac{d_1}{d_1+d_2} = c(0)(\epsilon(0)^2 + \delta(0)^2)$. Therefore, we can remove these terms from $\frac{dc}{dt}$ at time $t = 0$, noting simple that $\frac{dc}{dt} \geq -\alpha o c$. Further, so long as $c$ is still decreasing (and therefore less than $c(0) = 1$),

$$\begin{aligned}\frac{d(\|\beta\|\epsilon - c(\epsilon^2+\delta^2))}{dt} &\geq \frac{d(\|\beta\|\epsilon - (\epsilon^2+\delta^2))}{dt} && \text{(I.38)}\\ &= (\|\beta\| - 2\epsilon)\frac{d\epsilon}{dt} - 2\delta\frac{d\delta}{dt} && \text{(I.39)}\\ &= (\|\beta\| - 2\epsilon)(-c^2\epsilon + \|\beta\|c) - 2\delta(-c^2\delta) && \text{(I.40)}\\ &= -c^2(\epsilon\|\beta\| - 2(\epsilon^2+\delta^2)) + c(\|\beta\|^2 - 2\epsilon) && \text{(I.41)}\\ &\geq -c(\epsilon\|\beta\| - 2(\epsilon^2+\delta^2)) + c(\|\beta\|^2 - 2\epsilon) && \text{(I.42)}\\ &= c(\|\beta\|^2 - 2\epsilon - \epsilon\|\beta\| + 2(\epsilon^2+\delta^2)) && \text{(I.43)}\\ &\geq c(\|\beta\|^2 - \epsilon(2 + \|\beta\|)). && \text{(I.44)}\end{aligned}$$

Since $c > 0$ at all times, this is non-negative so long as the term in parentheses is non-negative, which holds so long as $\epsilon \leq \frac{\|\beta\|^2}{\|\beta\|+2}$. Further, since $\epsilon c \leq \|\beta\|$ we have

$$\frac{d\epsilon^2}{dt} = 2\epsilon \frac{d\epsilon}{dt} \tag{I.45}$$

$$= -2c^2\epsilon^2 + 2\epsilon c\|\beta\| \tag{I.46}$$

$$\leq 2\|\beta\|^2. \tag{I.47}$$

This implies $\epsilon(t)^2 \leq \epsilon(0)^2 + 2t\|\beta\|^2$. Therefore, for $t \leq \frac{\ln \|\beta\|/2}{2\|\beta\|}$ we have $\epsilon(t)^2 \leq \frac{1}{d_1+d_2} + \|\beta\| \ln \|\beta\|/2 \leq \frac{\|\beta\|^4}{(\|\beta\|+2)^2}$ (this inequality holds for $\|\beta\| \geq 2$). This satisfies the desired upper bound.

Thus the term in Equation (I.44) is non-negative for all $t \leq \frac{\ln \|\beta\|/2}{2\|\beta\|}$, and so we have $\frac{dc}{dt} \geq -\alpha oc$ under the above conditions. Since the derivative of $o$ is negative in $c$, a lower bound on $\frac{dc}{dt}$ gives us an upper bound on $\frac{do}{dt}$, which in turn maintains a valid lower bound on $\frac{dc}{dt}$ This allows us to solve for just the ODE given by

$$\frac{dc^2}{dt} = -2\alpha c^2 o, \tag{I.48}$$

$$\frac{do}{dt} = -2\alpha c^2 o. \tag{I.49}$$

Recalling the initial values of $c^2, o$, The solution to this system is given by

$$c(t)^2 = \frac{m}{1 - \frac{(1-m)}{\exp(2\alpha mt)}}, \tag{I.50}$$

$$o(t) = \frac{m}{\frac{\exp(2\alpha mt)}{1-m} - 1} \tag{I.51}$$

$$= \frac{m}{\exp(2\alpha mt)(1+k^{-1}) - 1} \tag{I.52}$$

Since these are bounds on the original problem, we have $c(t)^2 \geq m$ and $o(t)$ shrinks exponentially fast in $t$. In particular, note that under the stated condition $\sqrt{\alpha} \geq \frac{\|\beta\| \ln k}{m(\ln \|\beta\|/2)}$ (recalling $k := \frac{d_2}{d_1} > 1$), we have $\frac{\ln k}{2\sqrt{\alpha}m} \leq \frac{\ln \|\beta\|/2}{2\|\beta\|}$. Therefore we can plug in this value for $t$, implying $o(t) \leq m \left(\frac{d_1}{d_2}\right)^{\sqrt{\alpha}} = mk^{-\sqrt{\alpha}}$ at some time before $t = \frac{\ln \|\beta\|/2}{2\|\beta\|}$.

Now we solve for the time at which $\frac{dc}{dt} \geq 0$. Returning to Equation (I.44), we can instead suppose that $\epsilon \leq \frac{\|\beta\|^2 - \gamma}{\|\beta\|+2} \implies \|\beta\|^2 - \epsilon(2 + \|\beta\|) \geq \gamma$ for some $\gamma > 0$. If this quantity was non-negative and has had a derivative of at least $\gamma$ until time

$t = \frac{\ln k}{2\sqrt{\alpha m}}$, then its value at that time must be at least $\frac{\gamma \ln k}{2\sqrt{\alpha m}}$. For $\frac{dc}{dt}$ to be non-negative, we need this to be greater than $c(t)^2 \alpha o(t)$, so it suffices to have $\frac{\gamma \ln k}{2\sqrt{\alpha m}} \geq \frac{\alpha m}{\exp(2\alpha m t)(1+k^{-1})-1} \impliedby \gamma \ln k \geq \frac{2\alpha^{3/2} m^2}{k^{\sqrt{\alpha}}(1+k^{-1})-1} \impliedby \gamma \geq \frac{2\alpha^{3/2} m^2 k^{-\sqrt{\alpha}}}{\ln k}$. Observe that the stated lower bound on $\alpha$ directly implies this inequality.

Finally, note that $\|b\|^2 = \epsilon^2 + \delta^2$, and therefore

$$\frac{d\|b\|^2}{dt} = 2\epsilon \frac{d\epsilon}{dt} + 2\delta \frac{d\delta}{dt} \tag{I.53}$$

$$= -2c^2(\epsilon^2 + \delta^2) + 2c\epsilon\|\beta\|. \tag{I.54}$$

Since $c(0) = 1$ and $c\epsilon < \|\beta\|$, this means $\|b\|^2$ will also be decreasing at initialization. Thus we have shown that all relevant quantities will decrease towards 0 at initialization, but that by time $t = \frac{\ln k}{2\sqrt{\alpha m}}$, we will have $\frac{dc}{dt} \geq 0$. □

### I.7.2 Proof of Proof of Theorem 10.3.2

*Proof.* Recall from the previous section that we have shown that at some time $t_1 \leq \frac{\ln k}{2\sqrt{\alpha m}}$, $c(t)^2$ will be greater than $m$ and increasing, and $o(t)$ will be upper bounded by $mk^{-\sqrt{\alpha}}$. Furthermore, $\epsilon(t)^2 \leq \frac{1}{d_1+d_2} + 2t\|\beta\|^2$. To show that the sharpness reaches a particular value, we must demonstrate that $c$ grows large enough before the point $c\epsilon \approx \|\beta\|$ where this growth will rapidly slow. To do this, we study the relative growth of $c$ vs. $\epsilon$.

Recall the derivatives of these two terms:

$$\frac{dc}{dt} = -(\epsilon^2 + \delta^2 + \alpha o^2)c + \|\beta\|\epsilon, \tag{I.55}$$

$$\frac{d\epsilon}{dt} = -c^2\epsilon + \|\beta\| c. \tag{I.56}$$

Considering instead their squares,

$$\frac{dc^2}{dt} = 2c\frac{dc}{dt} \tag{I.57}$$

$$= -2(\epsilon^2 + \delta^2 + \alpha o^2)c^2 + 2\|\beta\|\epsilon c, \tag{I.58}$$

$$\frac{d\epsilon^2}{dt} = 2\epsilon \frac{d\epsilon}{dt} \tag{I.59}$$

$$= -2\epsilon^2 c^2 + 2\|\beta\|\epsilon c. \tag{I.60}$$

Since $\delta, o$ decrease monotonically, we have $\frac{dc^2}{dt} \geq -2(\epsilon^2 + \frac{d_1}{d_1+d_2} + \alpha m \left(\frac{d_1}{d_2}\right)^{\sqrt{\alpha}})c^2 +$

$2\|\beta\|\epsilon$. Thus if we can show that

$$\|\beta\|\epsilon c \geq (\epsilon^2 + 2(\frac{d_1}{d_1+d_2} + \alpha m \left(\frac{d_1}{d_2}\right)^{\sqrt{\alpha}}))c^2, \tag{I.61}$$

we can conclude that $\frac{dc^2}{dt} \geq (\epsilon^2 c^2 + \|\beta\|\epsilon c) = \frac{1}{2}\frac{d\epsilon^2}{dt}$—that is, that $c(t)^2$ grows at least half as fast as $\epsilon(t)^2$. And since $\delta, o$ continue to decrease, this inequality will continue to hold thereafter.

Simplifying the above desired inequality, we get

$$\|\beta\|\frac{\epsilon}{c} \geq \epsilon^2 + 2m(1 + \alpha k^{-\sqrt{\alpha}}). \tag{I.62}$$

Noting that $\frac{\epsilon}{c} \geq 1$ and $m = \frac{d_1}{d_1+d_2} \leq \frac{1}{2}$, and recalling the upper bound on $\epsilon(t)^2$, this reduces to proving

$$\|\beta\| \geq \frac{1}{d_1+d_2} + 2t\|\beta\|^2 + 1 + \alpha k^{-\sqrt{\alpha}}. \tag{I.63}$$

Since this occurs at some time $t_1 \leq \frac{\ln k}{2\sqrt{\alpha}m}$, and since $m^{-1} = 1 + k$, we get

$$\|\beta\| \geq \frac{1}{d_1+d_2} + \frac{\|\beta\|^2(1+k)\ln k}{\sqrt{\alpha}} + 1 + \alpha k^{-\sqrt{\alpha}}. \tag{I.64}$$

The assumed lower bound on $\sqrt{\alpha}$ means the sum of the first three terms can be upper bounded by a small $1 + o(1)$ term (say, 9/5) and recalling $\|\beta\| \geq$ 24/5 it suffices to prove

$$\|\beta\| \geq \frac{9}{5} + \alpha k^{-\sqrt{\alpha}} \tag{I.65}$$

$$\Longleftarrow \alpha k^{-\sqrt{\alpha}} \leq 3. \tag{I.66}$$

Taking logs,

$$\frac{2\ln\sqrt{\alpha}}{\ln k} - \sqrt{\alpha} \leq \ln 3, \tag{I.67}$$

which is clearly satisfied for $\sqrt{\alpha} \geq 1 + k\ln k$. As argued above, this implies $\frac{dc^2}{dt} \geq \frac{1}{2}\frac{d\epsilon^2}{dt}$ by some time $t_2 \leq \frac{\ln k}{2\sqrt{\alpha}m}$.

Consider the time $t_2$ at which this first occurs, whereby $c(t_2)^2$ is growing by at least one-half the rate of $\epsilon(t_2)^2$. Here we note that we can derive an upper bound on $c$ and $\epsilon$ at this time using our lemma and the fact that

$$\frac{dc}{dt} \leq \|\beta\|\epsilon, \tag{I.68}$$

$$\frac{d\epsilon}{dt} \leq \|\beta\|c. \tag{I.69}$$

The solution to this system implies

$$c(t_2) \leq \frac{1}{2}\left(\frac{\exp(\|\beta\|t_2) - \exp(-\|\beta\|t_2)}{\sqrt{d_1+d_2}} + \exp(\|\beta\|t_2) + 1\right) \tag{I.70}$$

$$\leq \frac{1}{2}\left(\exp(\|\beta\|t_2)\left(1 + \frac{1}{\sqrt{d_1+d_2}}\right) + 1\right) \tag{I.71}$$

$$\leq \frac{1}{2}\left(\exp\left(\frac{\|\beta\|\ln k}{2\sqrt{\alpha}m}\right)\left(1 + \frac{1}{\sqrt{d_1+d_2}}\right) + 1\right), \tag{I.72}$$

$$\epsilon(t_2) \leq \frac{1}{2}\left(\exp(\|\beta\|t_2)\left(1 + \frac{1}{\sqrt{d_1+d_2}}\right) + \frac{1}{\sqrt{d_1+d_2}} - 1\right) \tag{I.73}$$

$$\leq \frac{1}{2}\left(\exp\left(\frac{\|\beta\|\ln k}{2\sqrt{\alpha}m}\right)\left(1 + \frac{1}{\sqrt{d_1+d_2}}\right) + \frac{1}{\sqrt{d_1+d_2}} - 1\right) \tag{I.74}$$

Then for $\alpha > \left(\frac{\|\beta\|\ln k}{m(\ln\|\beta\| - \ln 2)}\right)^2$, the exponential term is upper bounded by $\frac{\sqrt{\|\beta\|}}{2}$, giving

$$c(t_2) \leq \frac{1}{2}\left(\frac{\sqrt{\|\beta\|}}{2}\left(1 + \frac{1}{\sqrt{d_1+d_2}}\right) + 1\right) \tag{I.75}$$

$$\leq \frac{\sqrt{\|\beta\|}}{2}, \tag{I.76}$$

$$\epsilon(t_2) \leq \frac{1}{2}\left(\frac{\sqrt{\|\beta\|}}{2}\left(1 + \frac{1}{\sqrt{d_1+d_2}}\right) + \frac{1}{\sqrt{d_1+d_2}} - 1\right) \tag{I.77}$$

$$\leq \frac{\sqrt{\|\beta\|}}{2}. \tag{I.78}$$

We know that optimization will continue until $\epsilon^2 c^2 = \|\beta\|^2$, and also that $\frac{dc^2}{dt} \geq \frac{1}{2}\frac{d\epsilon^2}{dt}$. Since $c \leq \epsilon$, this implies that $\epsilon^2 \geq \|\beta\|$ before convergence. Suppose that starting from time $t_2$, $\epsilon^2$ grows until time $t'$ by an additional amount $s$. Then we have

$$s = \epsilon(t')^2 - \epsilon(t_2)^2 \tag{I.79}$$

$$= \int_{t_2}^{t'} \frac{d\epsilon(t)^2}{dt} \tag{I.80}$$

$$\leq \int_{t_2}^{t'} 2\frac{dc(t)^2}{dt} \tag{I.81}$$

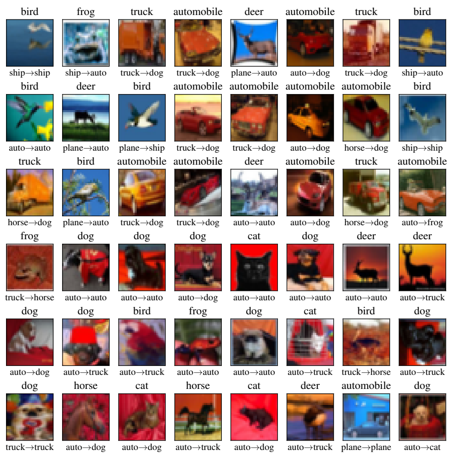
$$= 2(c(t')^2 - c(t_2)^2). \tag{I.82}$$

In other words, $c^2$ must have grown by at least half that amount. Since $\epsilon(t_2)^2 \leq \frac{\|\beta\|}{4}$ and therefore $\epsilon(t')^2 \leq \frac{\|\beta\|}{4} + s$, even if $c(t')^2$ is the minimum possible value of $\frac{s}{2}$ we must have at convergence $\frac{s}{2} = c^2 = \frac{\|\beta\|^2}{\epsilon^2} \geq \frac{\|\beta\|^2}{\frac{\|\beta\|}{4}+s}$. This is a quadratic in $s$ and solving tells us that we must have $s \geq \frac{5}{4}\|\beta\|$. Therefore, $c(t')^2 \geq \frac{5}{8}\|\beta\|$ is guaranteed to occur. Noting our derivation of the loss Hessian, this implies the sharpness must reach at least $\frac{5}{8}\alpha\|\beta\|$ for each dimension of $b_o$. $\square$

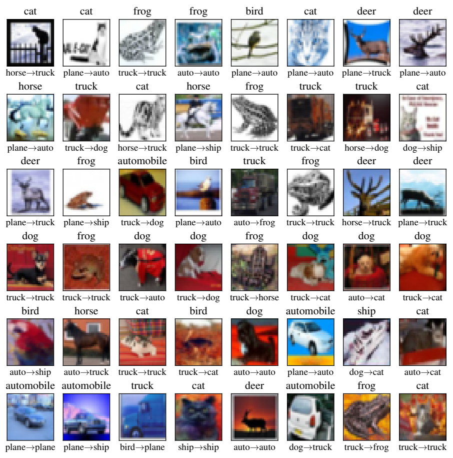
### I.8 Additional Samples Under Various Architectures/Seeds

To demonstrate the robustness of our finding we train a ResNet-18, VGG-11, and a Vision Transformer for 1000 steps with full-batch GD, each with multiple random initializations. For each run, we identify the 24 training examples with the most positive and most negative change in loss from step $i$ to step $i+1$, for $i \in \{100, 250, 500, 750\}$. We then display these images along with their label (above) and the network's predicted label before and after the gradient step (below). The change in the network's predicted labels display a clear pattern, where certain training samples cause the network to associate an opposing signal with a new class, which the network then overwhelmingly predicts whenever that feature is present.

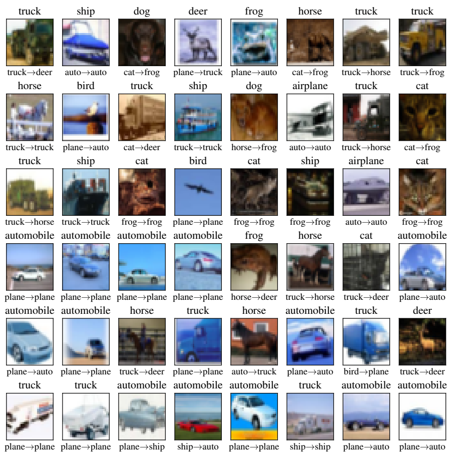
Consistent with our other experiments, we find that early opposing signals tend to be "simpler", e.g. raw colors, whereas later signals are more nuanced, such as the presence of a particular texture. We also see that the Vision Transformer seems to learn complex features earlier, and that they are less obviously aligned with human perception—this is not surprising since they process inputs in a fundamentally different manner than traditional ConvNets.


bird frog truck automobile deer automobile truck bird
ship→ship ship→auto truck→dog truck→dog plane→auto auto→dog truck→dog ship→auto
bird deer bird automobile automobile automobile automobile bird
auto→auto plane→auto plane→ship truck→dog truck→dog auto→dog horse→dog ship→ship
truck bird automobile automobile deer automobile truck automobile
horse→dog plane→auto truck→dog truck→dog auto→auto auto→dog horse→dog auto→frog
frog dog dog dog cat dog deer deer
truck→horse auto→auto auto→auto auto→dog auto→auto auto→auto auto→auto auto→truck
dog dog bird frog dog cat bird dog
auto→dog auto→truck auto→truck auto→auto auto→auto auto→truck truck→horse auto→truck
dog horse cat horse cat deer automobile dog
truck→truck auto→dog auto→dog auto→truck auto→dog auto→truck plane→plane auto→cat

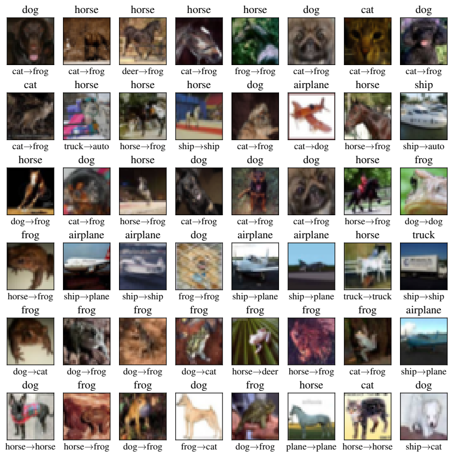
*(a)* Step 100 to 101


cat cat frog frog bird cat deer deer
horse→truck plane→auto truck→truck auto→auto plane→auto plane→auto plane→truck plane→auto
horse truck cat horse frog truck truck cat
plane→auto truck→dog horse→truck plane→ship truck→truck truck→cat horse→dog dog→ship
deer frog automobile bird truck frog deer deer
plane→truck plane→ship truck→dog plane→auto auto→frog truck→truck horse→truck plane→truck
dog frog dog dog frog dog dog dog
truck→truck truck→truck truck→auto truck→dog truck→horse truck→cat auto→cat truck→cat
bird horse cat bird dog automobile ship cat
auto→ship auto→truck truck→truck truck→cat auto→auto plane→auto dog→cat auto→cat
automobile automobile truck cat deer automobile frog cat
plane→plane plane→ship bird→plane ship→ship auto→auto dog→truck truck→frog truck→truck

*(b)* Step 250 to 251

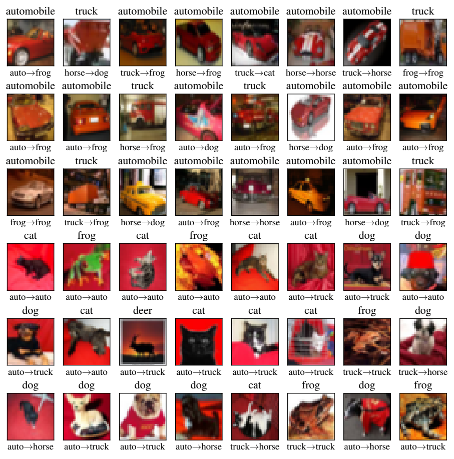

truck ship dog deer frog horse truck truck
truck→deer auto→auto cat→frog plane→truck plane→auto cat→frog truck→horse truck→frog
horse bird truck ship dog airplane truck cat
truck→truck plane→auto cat→deer truck→truck horse→frog auto→auto truck→horse cat→frog
truck ship cat bird cat ship airplane cat
truck→horse truck→truck frog→frog plane→plane frog→frog frog→frog auto→auto frog→frog
automobile automobile automobile automobile frog horse cat automobile
plane→plane plane→plane plane→plane plane→plane horse→deer truck→horse truck→deer plane→auto
automobile automobile horse truck horse automobile truck deer
plane→auto plane→plane truck→deer plane→plane auto→truck plane→auto bird→plane truck→deer
truck truck automobile automobile automobile truck automobile automobile
plane→plane plane→plane plane→ship ship→auto plane→plane ship→ship plane→auto plane→auto

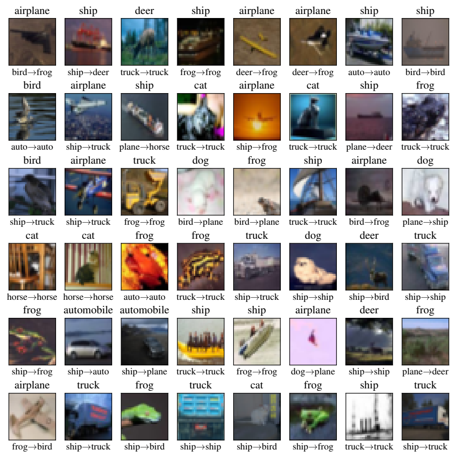
*(c)* Step 500 to 501

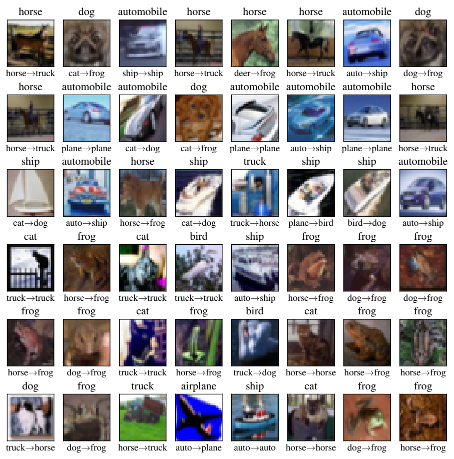

dog horse horse horse horse dog cat dog
cat→frog cat→frog deer→frog cat→frog frog→frog cat→frog cat→frog cat→frog
cat horse horse horse dog airplane horse ship
cat→frog truck→auto horse→frog ship→ship cat→frog cat→dog horse→frog ship→auto
horse dog horse dog dog dog horse frog
dog→frog cat→frog horse→frog cat→frog cat→frog cat→frog horse→frog dog→dog
frog airplane airplane dog airplane airplane horse truck
horse→frog ship→plane ship→ship frog→frog ship→plane ship→plane truck→truck ship→ship
frog frog frog frog frog frog frog airplane
dog→cat dog→frog dog→frog dog→cat horse→deer horse→frog cat→frog ship→plane
dog frog frog dog frog horse cat dog
horse→horse horse→frog dog→frog frog→cat dog→frog plane→plane horse→horse ship→cat

*(d)* Step 750 to 751

Figure I.40: **(ResNet-18, seed 1)** Images with the most positive (top 3 rows) and most negative (bottom 3 rows) change to training loss after steps 100, 250, 500, and 750. Each image has the true label (above) and the predicted label before and after the gradient update (below).


*(a)* Step 100 to 101


*(b)* Step 250 to 251


*(c)* Step 500 to 501

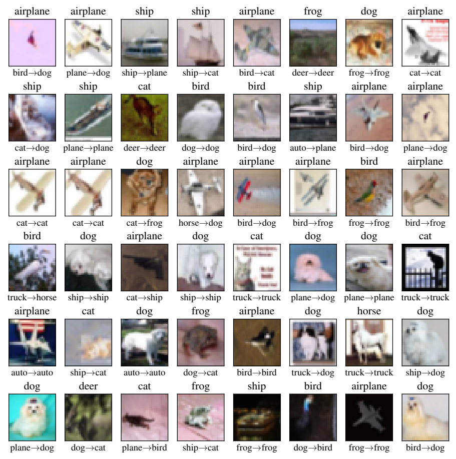
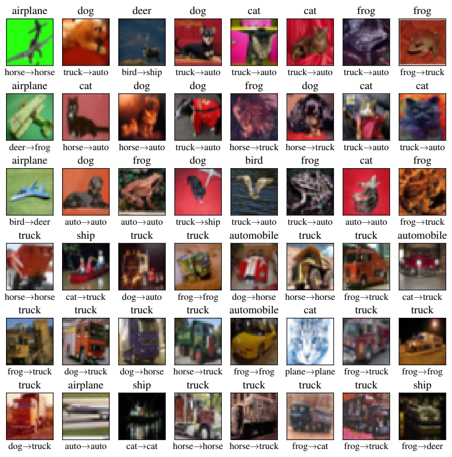
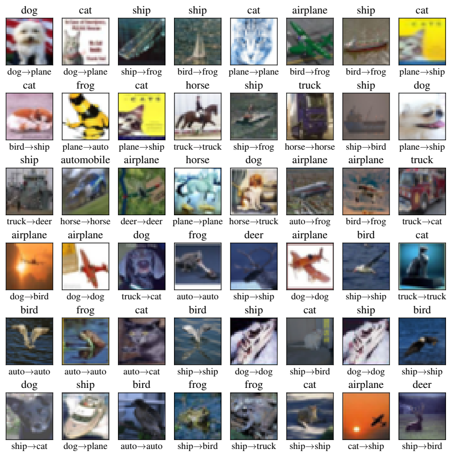

*(d)* Step 750 to 751

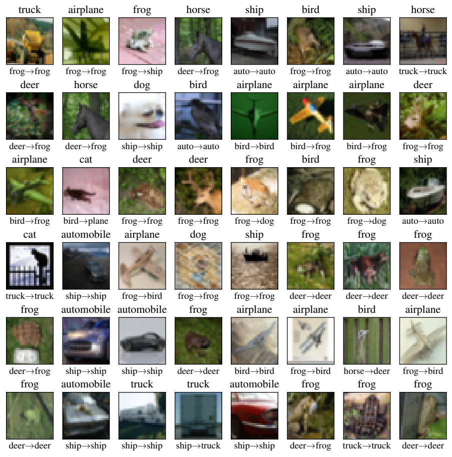
Figure I.41: **(ResNet-18, seed 2)** Images with the most positive (top 3 rows) and most negative (bottom 3 rows) change to training loss after steps 100, 250, 500, and 750. Each image has the true label (above) and the predicted label before and after the gradient update (below).



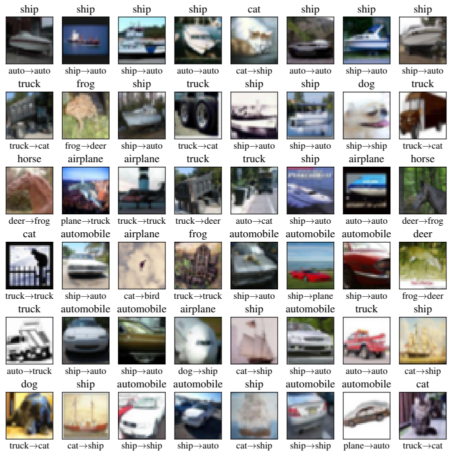
*(a)* Step 100 to 101



*(b)* Step 250 to 251

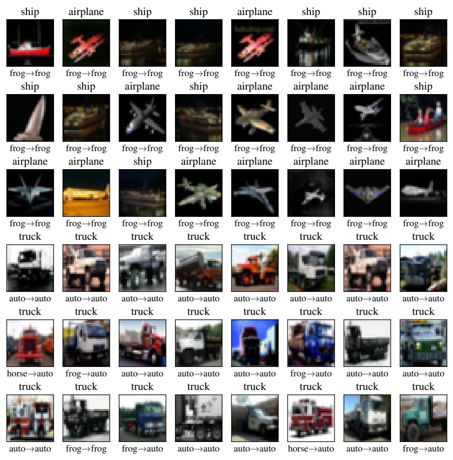


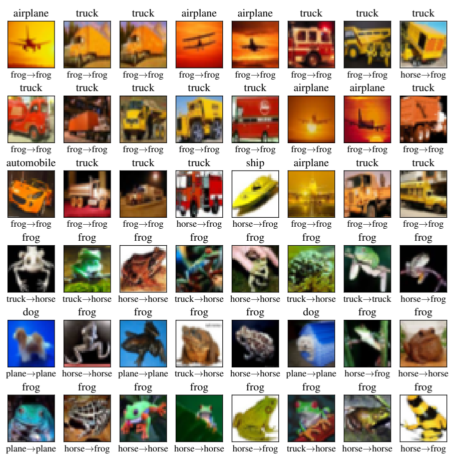
*(c)* Step 500 to 501

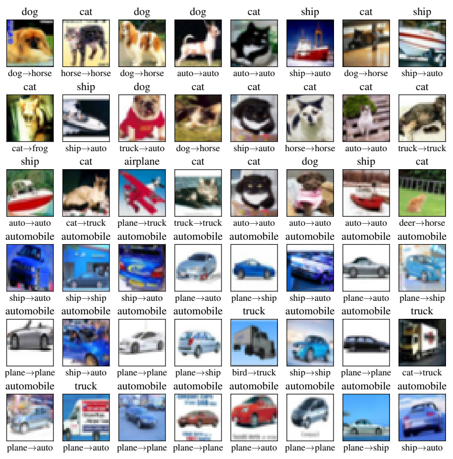


*(d)* Step 750 to 751

Figure I.42: **(ResNet-18, seed 3)** Images with the most positive (top 3 rows) and most negative (bottom 3 rows) change to training loss after steps 100, 250, 500, and 750. Each image has the true label (above) and the predicted label before and after the gradient update (below).


*(a)* Step 100 to 101


*(b)* Step 250 to 251


*(c)* Step 500 to 501

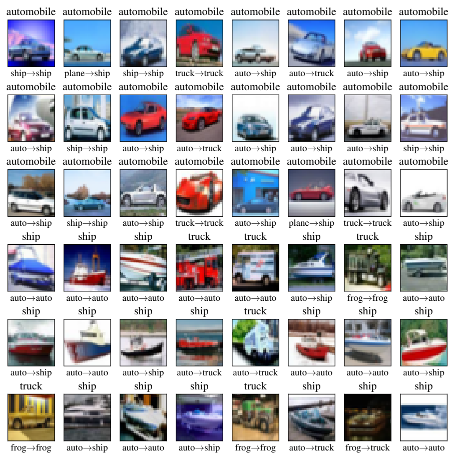
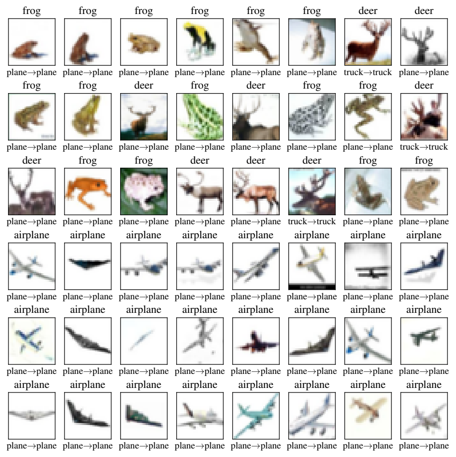
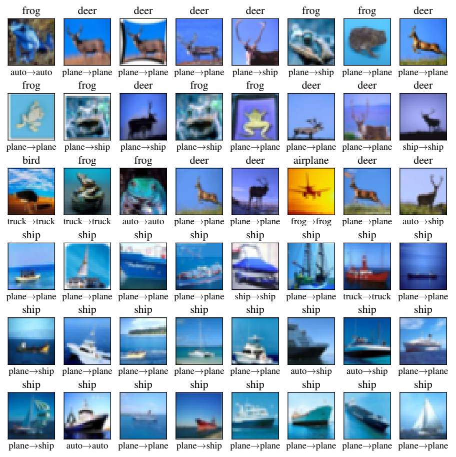

*(d)* Step 750 to 751

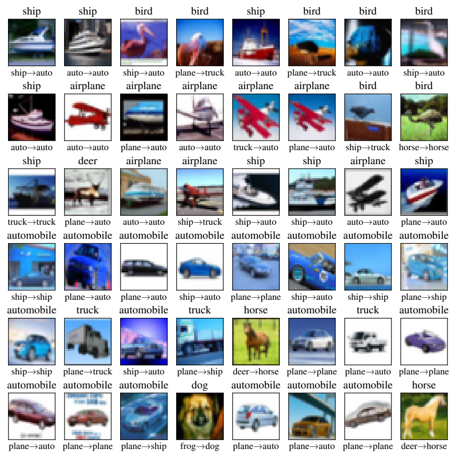
Figure I.43: **(VGG-11, seed 1)** Images with the most positive (top 3 rows) and most negative (bottom 3 rows) change to training loss after steps 100, 250, 500, and 750. Each image has the true label (above) and the predicted label before and after the gradient update (below).



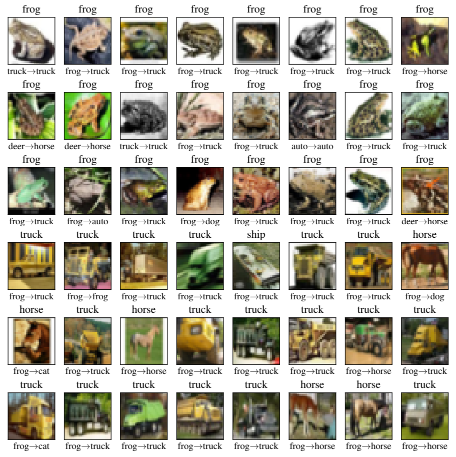
*(a)* Step 100 to 101



*(b)* Step 250 to 251

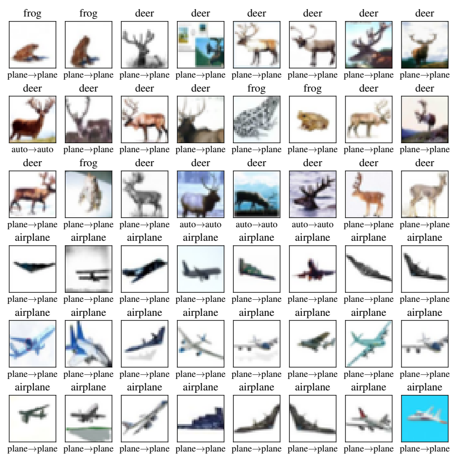


*(c)* Step 500 to 501

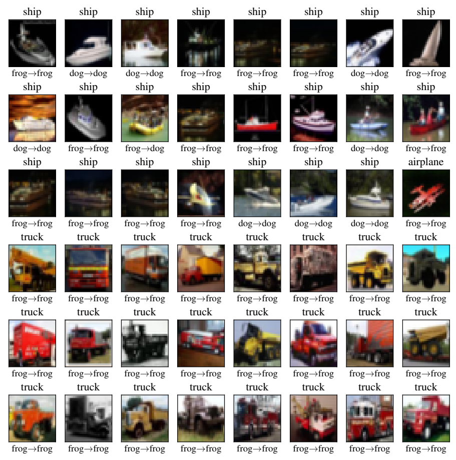


*(d)* Step 750 to 751

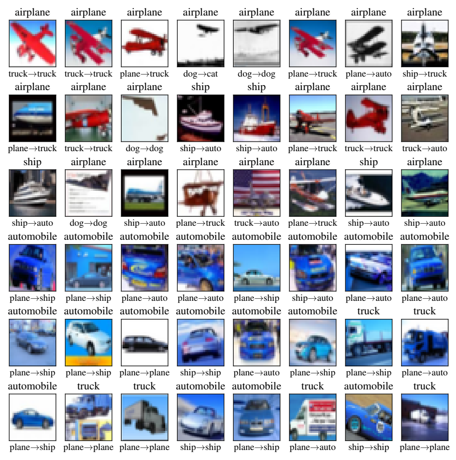
Figure I.44: **(VGG-11, seed 2)** Images with the most positive (top 3 rows) and most negative (bottom 3 rows) change to training loss after steps 100, 250, 500, and 750. Each image has the true label (above) and the predicted label before and after the gradient update (below).



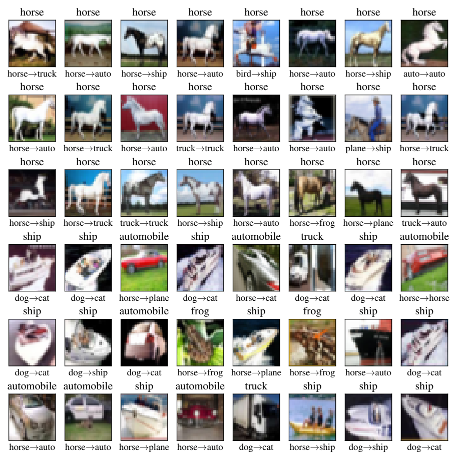
*(a)* Step 100 to 101



*(b)* Step 250 to 251

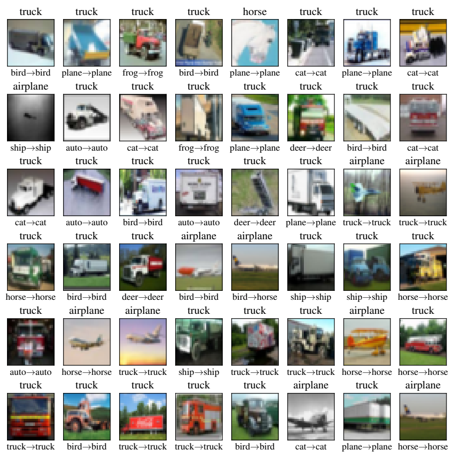


*(c)* Step 500 to 501

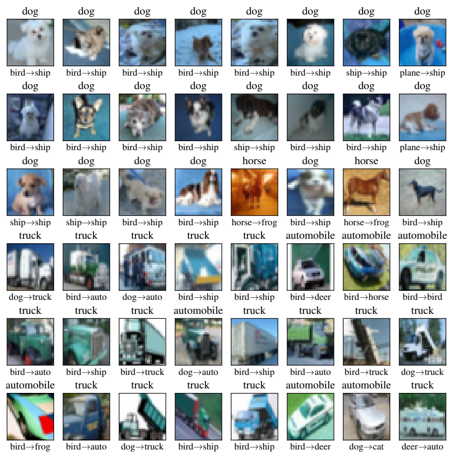


*(d)* Step 750 to 751

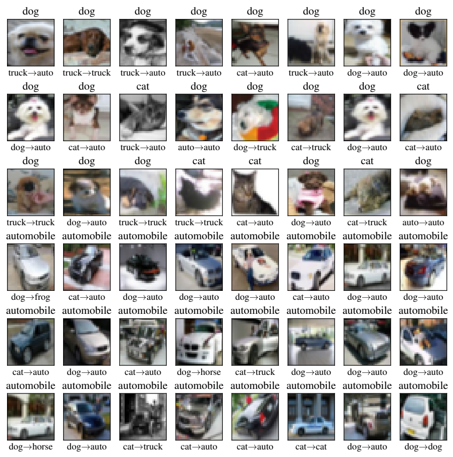
Figure I.45: **(VGG-11, seed 3)** Images with the most positive (top 3 rows) and most negative (bottom 3 rows) change to training loss after steps 100, 250, 500, and 750. Each image has the true label (above) and the predicted label before and after the gradient update (below).



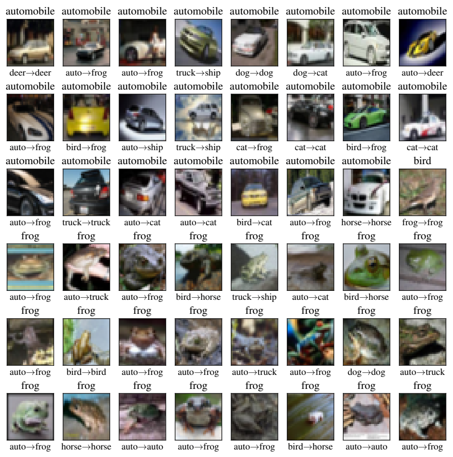
*(a)* Step 100 to 101



*(b)* Step 250 to 251

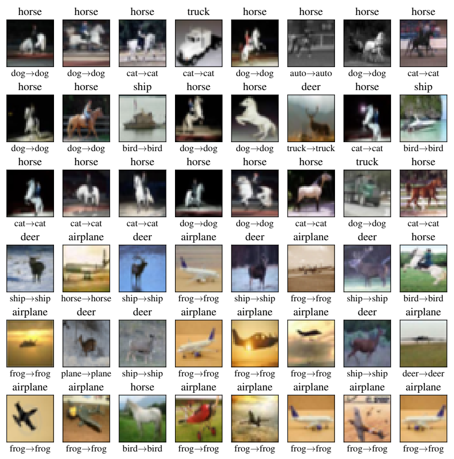


*(c)* Step 500 to 501

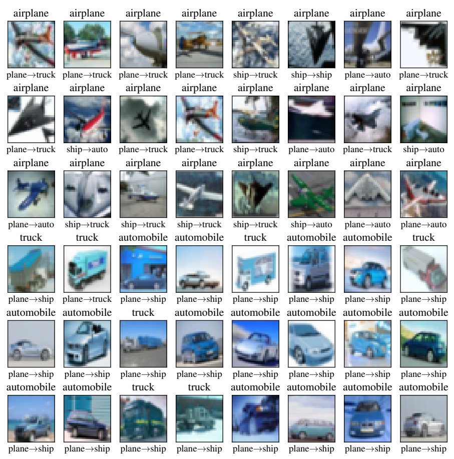


*(d)* Step 750 to 751

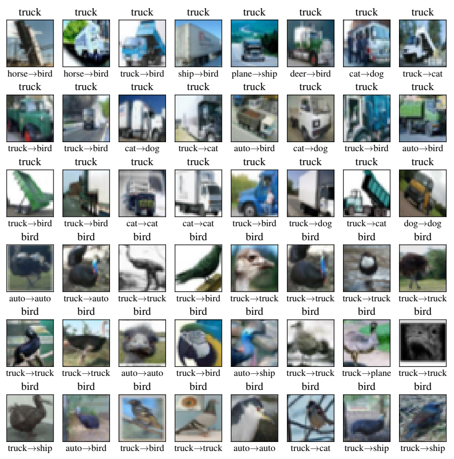
Figure I.46: **(ViT, seed 1)** Images with the most positive (top 3 rows) and most negative (bottom 3 rows) change to training loss after steps 100, 250, 500, and 750. Each image has the true label (above) and the predicted label before and after the gradient update (below).



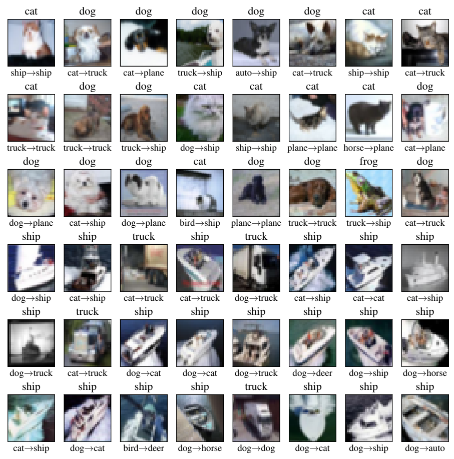
*(a)* Step 100 to 101



*(b)* Step 250 to 251

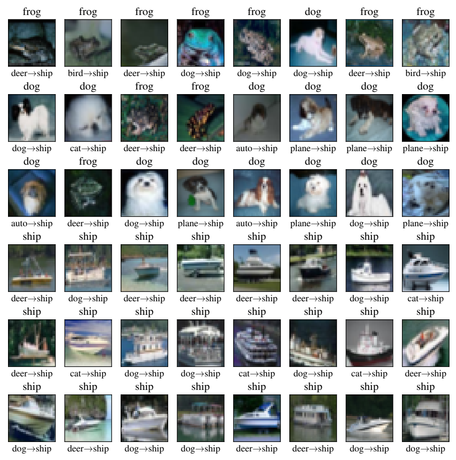


*(c)* Step 500 to 501

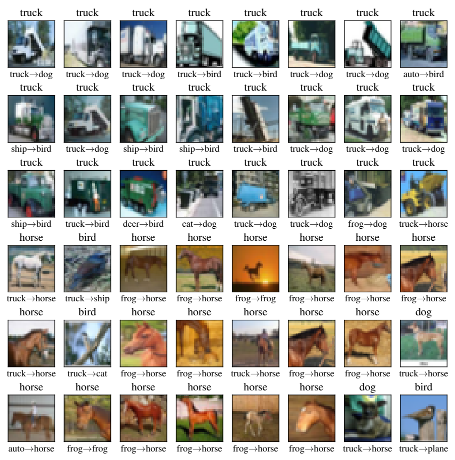


*(d)* Step 750 to 751

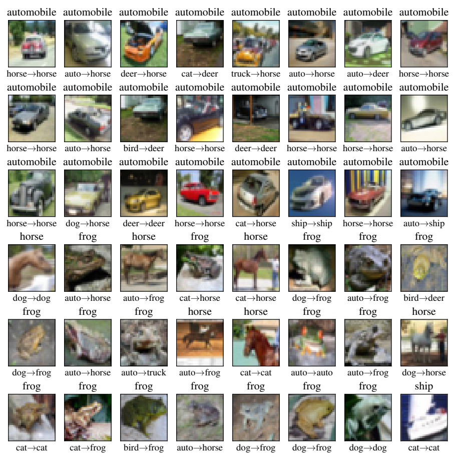
Figure I.47: **(ViT, seed 2)** Images with the most positive (top 3 rows) and most negative (bottom 3 rows) change to training loss after steps 100, 250, 500, and 750. Each image has the true label (above) and the predicted label before and after the gradient update (below).


frog frog frog frog frog dog frog frog
deer→ship bird→ship deer→ship dog→ship dog→ship dog→ship deer→ship bird→ship
dog dog frog frog dog dog dog dog
dog→ship cat→ship deer→ship deer→ship auto→ship plane→ship plane→ship plane→ship
dog frog dog dog dog dog dog dog
auto→ship deer→ship dog→ship plane→ship auto→ship plane→ship dog→ship plane→ship
ship ship ship ship ship ship ship ship
deer→ship dog→ship deer→ship deer→ship deer→ship deer→ship dog→ship cat→ship
ship ship ship ship ship ship ship ship
deer→ship cat→ship dog→ship dog→ship cat→ship dog→ship cat→ship deer→ship
ship ship ship ship ship ship ship ship
dog→ship deer→ship dog→ship dog→ship deer→ship deer→ship dog→ship dog→ship

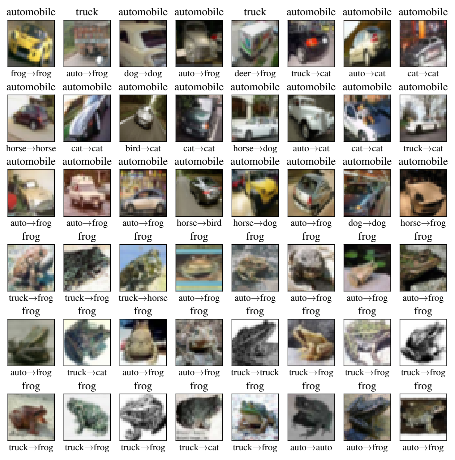
*(a)* Step 100 to 101


truck truck truck truck truck truck truck truck
truck→dog truck→dog truck→dog truck→bird truck→bird truck→dog truck→dog auto→bird
truck truck truck truck truck truck truck truck
ship→bird truck→dog ship→bird ship→bird truck→bird truck→dog truck→dog truck→dog
truck truck truck truck truck truck truck truck
ship→bird truck→bird deer→bird cat→dog truck→dog truck→dog frog→dog truck→horse
horse bird horse horse horse horse horse horse
truck→horse truck→ship frog→horse frog→horse frog→frog frog→horse frog→horse frog→horse
horse bird horse horse horse horse horse dog
truck→horse truck→cat frog→horse frog→horse truck→horse frog→horse frog→horse truck→horse
horse horse horse horse horse horse dog bird
auto→horse frog→frog frog→horse frog→horse frog→horse frog→horse truck→horse truck→plane

*(b)* Step 250 to 251


automobile automobile automobile automobile automobile automobile automobile automobile
horse→horse auto→horse deer→horse cat→deer truck→horse auto→horse auto→deer horse→horse
automobile automobile automobile automobile automobile automobile automobile automobile
horse→horse auto→horse bird→deer horse→horse deer→deer horse→horse horse→horse auto→horse
automobile automobile automobile automobile automobile automobile automobile automobile
horse→horse dog→horse deer→deer horse→horse cat→horse ship→ship horse→horse auto→ship
horse frog horse frog horse frog frog frog
dog→dog auto→horse auto→frog cat→horse cat→horse dog→frog auto→frog bird→deer
frog frog frog horse horse frog frog horse
dog→frog auto→horse auto→truck auto→frog cat→cat auto→auto auto→frog dog→horse
frog frog frog frog frog frog frog ship
cat→cat cat→frog bird→frog auto→horse dog→frog dog→frog dog→dog cat→cat

*(c)* Step 500 to 501


automobile truck automobile automobile truck automobile automobile automobile
frog→frog auto→frog dog→dog auto→frog deer→frog truck→cat auto→cat cat→cat
automobile automobile automobile automobile automobile automobile automobile automobile
horse→horse cat→cat bird→cat cat→cat horse→dog auto→cat cat→cat truck→cat
automobile automobile automobile automobile automobile automobile automobile automobile
auto→frog auto→frog auto→frog horse→bird horse→dog auto→frog dog→dog horse→frog
frog frog frog frog frog frog frog frog
truck→frog truck→frog truck→horse auto→frog auto→frog auto→frog auto→frog auto→frog
frog frog frog frog frog frog frog frog
auto→frog truck→cat auto→frog auto→frog truck→truck truck→frog truck→frog truck→frog
frog frog frog frog frog frog frog frog
truck→frog truck→frog truck→frog truck→cat truck→frog auto→auto auto→frog auto→frog

*(d)* Step 750 to 751

Figure I.48: **(ViT, seed 3)** Images with the most positive (top 3 rows) and most negative (bottom 3 rows) change to training loss after steps 100, 250, 500, and 750. Each image has the true label (above) and the predicted label before and after the gradient update (below).

# Bibliography

Jacob Abernethy, Peter L Bartlett, Alexander Rakhlin, and Ambuj Tewari. Optimal strategies and minimax lower bounds for online convex games. In *Proceedings of the 21st annual conference on learning theory*, pages 414–424, 2008.

Saba Ahmadi, Hedyeh Beyhaghi, Avrim Blum, and Keziah Naggita. The strategic perceptron. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 6–25, 2021.

Kartik Ahuja, Jun Wang, Amit Dhurandhar, Karthikeyan Shanmugam, and Kush R. Varshney. Empirical or invariant risk minimization? a sample complexity perspective. In *International Conference on Learning Representations*, 2021.

Isabela Albuquerque, João Monteiro, Mohammad Darvishi, Tiago H. Falk, and Ioannis Mitliagkas. Generalizing to unseen domains via distribution matching. *arXiv preprint arXiv:1911.00804*, 2020.

Pierre Alquier, The Tien Mai, and Massimiliano Pontil. Regret Bounds for Lifelong Learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 261–269, 2017.

Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301. PMLR, 2019.

Ifigeneia Apostolopoulou, Ian Char, Elan Rosenfeld, and Artur Dubrawski. Deep attentive variational inference. In *International Conference on Learning Representations*, 2021.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, pages 244–253. PMLR, 2018.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32, 2019.

Sanjeev Arora, Zhiyuan Li, and Abhishek Panigrahi. Understanding gradient descent on the edge of stability in deep learning. In *International Conference on Machine Learning*, pages 948–1024. PMLR, 2022.

Anish Athalye and Nicholas Carlini. On the robustness of the cvpr 2018 white-box adversarial example defenses. *arXiv preprint arXiv:1804.03286*, 2018.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, July 2018.

Benjamin Aubin, Agnieszka Słowik, Martin Arjovsky, Leon Bottou, and David Lopez-Paz. Linear unit-tests for invariance discovery. *arXiv preprint arXiv:2102.10867*, 2021.

Christina Baek, Yiding Jiang, Aditi Raghunathan, and J Zico Kolter. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. In *Advances in Neural Information Processing Systems*, 2022.

J Andrew Bagnell. Robust supervised learning. In *Proceedings of the 20th national conference on Artificial intelligence-Volume 2*, pages 714–719, 2005.

Mahsa Baktashmotlagh, Mehrtash T. Harandi, Brian C. Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *2013 IEEE International Conference on Computer Vision*, pages 769–776, 2013.

Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. Efficient representations for lifelong learning and autoencoding. In *Proceedings of The 28th Conference on Learning Theory*, pages 191–210, 2015.

Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE Transactions on Medical Imaging*, 2018.

Yamini Bansal, Gal Kaplun, and Boaz Barak. For self-supervised learning, rationality implies generalization, provably. In *International Conference on Learning Representations*, 2021.

Boaz Barak, Benjamin L. Edelman, Surbhi Goel, Sham M. Kakade, eran malach, and Cyril Zhang. Hidden progress in deep learning: SGD learns parities near the computational limit. In *Advances in Neural Information Processing Systems*, 2022.

Marco Barreno, Blaine Nelson, Russell Sears, Anthony D. Joseph, and J. D. Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, pages 16–25, New York, NY, USA, 2006.

Peter L. Bartlett, Elad Hazan, and Alexander Rakhlin. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems*, page 65–72, 2007.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in neural information processing systems*, pages 6240–6249, 2017.

Yahav Bechavod, Chara Podimata, Steven Wu, and Juba Ziani. Information discrepancy in strategic learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 1691–1715. PMLR, 17–23 Jul 2022.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 456–473, 2018.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Alexis Bellot and Mihaela van der Schaar. Generalization and invariances in the presence of unobserved confounding. *arXiv preprint arXiv:2007.10653*, 2020.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010a.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010b.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton University Press, 2009.

Frederik Benzing. Gradient descent on neurons and its link to approximate second-order optimization. In *International Conference on Machine Learning*, pages

1817–1853. PMLR, 2022.

Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.

Kush Bhatia, Prateek Jain, and Purushottam Kar. Robust regression via hard thresholding. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, pages 721–729, Cambridge, MA, USA, 2015.

Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10(9), 2009.

B. Biggio, G. Fumera, and F. Roli. Security evaluation of pattern classifiers under attack. *IEEE Transactions on Knowledge and Data Engineering*, 26(4):984–996, April 2014.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Support vector machines under adversarial label noise. In *Proceedings of the Asian Conference on Machine Learning*, volume 20, pages 97–112, South Garden Hotels and Resorts, Taoyuan, Taiwain, 14–15 Nov 2011.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*, pages 387–402. Springer, 2013.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Kernel descriptors for visual recognition. In *Advances in Neural Information Processing Systems 23*, pages 244–252. 2010.

Kenneth A Bollen. Structural equation models. *Encyclopedia of biostatistics*, 7, 2005.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 343–351, Red Hook, NY, USA, 2016. Curran Associates Inc.

Michael Brückner and Tobias Scheffer. Nash equilibria of static prediction games. In *Advances in neural information processing systems*, pages 171–179, 2009.

Michael Brückner, Christian Kanzow, and Tobias Scheffer. Static prediction games for adversarial learning problems. *The Journal of Machine Learning Research*, 13(1):2617–2654, 2012.

Simon Buchholz, Goutham Rajendran, Elan Rosenfeld, Bryon Aragam, Bernhard Schölkopf, and Pradeep Ravikumar. Learning linear causal representations from interventions under general nonlinear mixing. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Rudy R Bunel, Ilker Turkaslan, Philip Torr, Pushmeet Kohli, and Pawan K Mudigonda. A unified view of piecewise linear neural network verification. *Advances in Neural Information Processing Systems*, 31, 2018.

Xiaoyu Cao and Neil Zhenqiang Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, pages 278–287, 2017.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14, New York, NY, USA, 2017.

Nicholas Carlini, Guy Katz, Clark Barrett, and David L Dill. Provably minimally-distorted adversarial examples. *arXiv preprint arXiv:1709.10207*, 2017.

Marcus Carlsson. Perturbation theory for the matrix square root and matrix modulus. 10 2018.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. *arXiv preprint arXiv:1906.03950*, 2019.

Jiefeng Chen, Frederick Liu, Besim Avci, Xi Wu, Yingyu Liang, and Somesh Jha. Detecting errors and estimating accuracy on unlabeled data with self-training ensembles. *Advances in Neural Information Processing Systems*, 34:14980–14992, 2021a.

Mayee Chen, Karan Goel, Nimit S Sohoni, Fait Poms, Kayvon Fatahalian, and Christopher Ré. Mandoline: Model evaluation under distribution shift. In *International Conference on Machine Learning*, pages 1617–1629. PMLR, 2021b.

Ruidi Chen and Ioannis Ch. Paschalidis. A robust learning approach for regression models based on distributionally robust optimization. *J. Mach. Learn. Res.*, 19 (1):517–564, January 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, et al. Symbolic discovery of optimization algorithms. *arXiv preprint arXiv:2302.06675*, 2023a.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2(5):6, 2021c.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Yatong Chen, Jialu Wang, and Yang Liu. Linear classifiers that encourage constructive adaptation. *Transactions on Machine Learning Research*, 2023b.

Yining Chen, Elan Rosenfeld, Mark Sellke, Tengyu Ma, and Andrej Risteski. Iterative feature matching: Toward provable domain generalization with logarithmic environments. In *Advances in Neural Information Processing Systems*, volume 35, pages 1725–1736. Curran Associates, Inc., 2022.

Yuansi Chen and Peter Bühlmann. Domain adaptation under structural causal models. *Journal of Machine Learning Research*, 22(261):1–80, 2021.

Yudong Chen, Constantine Caramanis, and Shie Mannor. Robust sparse regression under adversarial corruption. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 774–782, Atlanta, Georgia, USA, 17–19 Jun 2013.

Chih-Hong Cheng, Georg Nührenberg, and Harald Ruess. Maximum resilience of artificial neural networks. In *Automated Technology for Verification and Analysis: 15th International Symposium, ATVA 2017, Pune, India, October 3–6, 2017, Proceedings 15*, pages 251–268. Springer, 2017.

Anirudh Choudhary, Li Tong, Yuanda Zhu, and May D Wang. Advancing medical imaging informatics by deep learning-based domain adaptation. *Yearbook of medical informatics*, 29(01):129–138, 2020.

Rune Christiansen, Niklas Pfister, Martin Emil Jakobsen, Nicola Gnecco, and Jonas Peters. A causal framework for distribution generalization. *arXiv preprint arXiv:2006.07433*, 2020.

Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

K Alec Chrystal, Paul D Mizen, and PD Mizen. Goodhart's law: its origins, meaning and implications for monetary policy. *Central banking, monetary theory and practice: Essays in honour of Charles Goodhart*, 1:221–243, 2003.

Ching-Yao Chuang, Antonio Torralba, and Stefanie Jegelka. Estimating generalization under distribution shifts via domain-invariant representations. *International conference on machine learning*, 2020.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International conference on machine learning*, pages 854–863. PMLR, 2017.

Charles J Clopper and Egon S Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019.

Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021.

Jeremy M Cohen, Behrooz Ghorbani, Shankar Krishnan, Naman Agarwal, Sourabh Medapati, Michal Badura, Daniel Suo, David Cardoze, Zachary Nado, George E Dahl, et al. Adaptive gradient methods at the edge of stability. *arXiv preprint arXiv:2207.14484*, 2022.

Francesco Croce, Maksym Andriushchenko, and Matthias Hein. Provable robustness of relu networks via maximization of linear regions. In *the 22nd International Conference on Artificial Intelligence and Statistics*, pages 2057–2066. PMLR, 2019.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

Ju L. Daleckii and S.G. Krein. Integration and differentiation of functions of hermitian operators and applications to the theory of perturbations. *AMS Translations (2), 47(1-30)*, 1965.

Alex Damian, Eshaan Nichani, and Jason D. Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. In *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*, 2022.

Etienne De Klerk. The complexity of optimizing over a simplex, hypercube or sphere: a short survey. *Central European Journal of Operations Research*, 16(2): 111–125, 2008.

Victor H de la Peña and Stephen J Montgomery-Smith. Decoupling inequalities for the tail probabilities of multivariate u-statistics. *The Annals of Probability*, pages 806–816, 1995.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Computer Vision and Pattern Recognition (CVPR)*, 2009.

Weijian Deng and Liang Zheng. Are labels always necessary for classifier accuracy evaluation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15069–15078, 2021.

Weijian Deng, Stephen Gould, and Liang Zheng. What does rotation prediction tell us about classifier accuracy under varying testing environments? *arXiv preprint arXiv:2106.05961*, 2021.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Ilias Diakonikolas, Weihao Kong, and Alistair Stewart. Efficient algorithms and lower bounds for robust linear regression. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2745–2754, Philadelphia, PA, USA, 2019.

Lee H. Dicker. Variance estimation in high-dimensional linear models. *Biometrika*, 101(2):269–284, 03 2014.

Vanessa Didelez, A Philip Dawid, and Sara Geneletti. Direct and indirect effects of sequential treatments. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pages 138–146, 2006.

Samuel Dodge and Lina Karam. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th international conference on computer communication and networks (ICCCN)*, pages 1–7. IEEE, 2017.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 647–655, Bejing, China, 22–24 Jun 2014.

Jinshuo Dong, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu. Strategic classification from revealed preferences. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pages 55–70, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Simon S Du, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019.

John Duchi, Tatsunori Hashimoto, and Hongseok Namkoong. Distributionally robust losses for latent covariate mixtures. *arXiv preprint arXiv:2007.13982*, 2019.

John C Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3): 1378–1406, 2021.

Souradeep Dutta, Susmit Jha, Sriram Sanakaranarayanan, and Ashish Tiwari. Output range analysis for deep neural networks. *arXiv preprint arXiv:1709.09130*, 2017.

Krishnamurthy Dvijotham, Sven Gowal, Robert Stanforth, Relja Arandjelovic, Brendan O'Donoghue, Jonathan Uesato, and Pushmeet Kohli. Training verified learners with learned verifiers. *arXiv preprint arXiv:1805.10265*, 2018a.

Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy A Mann, and Pushmeet Kohli. A dual approach to scalable verification of deep networks. In *UAI*, volume 1, page 3, 2018b.

Krishnamurthy Dvijotham, Jamie Hayes, Borja Balle, Zico Kolter, Chongli Qin, Andras Gyorgy, Kai Xiao, Sven Gowal, and Pushmeet Kohli. A framework for robustness certification of smoothed classifiers using f-divergences. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, Conference Track Proceedings*, 2020.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Ruediger Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In *Automated Technology for Verification and Analysis: 15th International Symposium, ATVA 2017, Pune, India, October 3–6, 2017, Proceedings 15*, pages 269–286. Springer, 2017.

Itay Eilat, Ben Finkelshtein, Chaim Baskin, and Nir Rosenfeld. Strategic classification with graph neural networks. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Lewis Elton. Goodhart's law and performance indicators in higher education. *Evaluation & Research in Education*, 18(1-2):120–128, 2004.

Chen Fang, Ye Xu, and Daniel N. Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *2013 IEEE International Conference on Computer Vision*, pages 1657–1664, 2013.

Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. *Advances in neural information processing systems*, 29, 2016.

Michael Fire and Carlos Guestrin. Over-optimization of academic publishing metrics: observing goodhart's law in action. *GigaScience*, 8(6):giz053, 2019.

Matteo Fischetti and Jason Jo. Deep neural networks as 0-1 mixed integer linear programs: A feasibility study. *arXiv preprint arXiv:1712.06174*, 2017.

Nic Ford, Justin Gilmer, Nicolas Carlini, and Dogus Cubuk. Adversarial examples are a natural consequence of test error in noise. *arXiv preprint arXiv:1901.10513*, 2019.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Stanislav Fort and Surya Ganguli. Emergent properties of the local geometry of neural loss landscapes. *arXiv preprint arXiv:1910.05929*, 2019.

Jean-Yves Franceschi, Alhussein Fawzi, and Omar Fawzi. Robustness of classifiers to uniform

$$\backslash \ell\_p$$

and gaussian noise. In *International Conference on Artificial Intelligence and Statistics*, pages 1280–1288. PMLR, 2018.

Matteo Gamba, Hossein Azizpour, and Mårten Björkman. On the lipschitz constant of deep networks and double descent. *arXiv preprint arXiv:2301.12309*, 2023.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, pages 7576–7586, 2018.

Saurabh Garg, Sivaraman Balakrishnan, J Zico Kolter, and Zachary C Lipton. Ratt: Leveraging unlabeled data to guarantee generalization. *arXiv preprint arXiv:2105.00303*, 2021.

Saurabh Garg, Sivaraman Balakrishnan, Zachary Chase Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. In *International Conference on Learning Representations*, 2022.

Saurabh Garg, Nick Erickson, James Sharpnack, Alex Smola, Siva Balakrishnan, and Zachary Lipton. Rlsbench: A large-scale empirical study of domain adaptation under relaxed label shift. In *International Conference on Machine Learning (ICML)*, 2023.

Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2018.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

Ganesh Ghalme, Vineet Nair, Itay Eilat, Inbal Talgam-Cohen, and Nir Rosenfeld. Strategic classification in the dark. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

AmirEmad Ghassami, Saber Salehkaleybar, Negar Kiyavash, and Kun Zhang. Learning causal structures using regression invariance. In *Advances in Neural Information Processing Systems*, pages 3011–3021, 2017.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*. PMLR, 09–15 Jun 2019.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus, 2019.

Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *International conference on machine learning*, pages 2839–2848, 2016.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016.

Charles AE Goodhart. Problems of monetary management: The UK experience. In *Papers in Monetary Economics, vol. I.* Reserve Bank of Australia, 1975.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael Cree. Regularisation of neural networks by enforcing lipschitz continuity. arxiv e-prints, page. *arXiv preprint arXiv:1804.04368*, 2018.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

Michael Großhans, Christoph Sawade, Michael Brückner, and Tobias Scheffer. Bayesian games for adversarial regression problems. In *International Conference on Machine Learning*, pages 55–63, 2013.

Victor Guillemin and Alan Pollack. *Differential topology*, volume 370. American Mathematical Soc., 2010.

Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. *arXiv preprint arXiv:2107.03315*, 2021.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

Moritz Hardt, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters. Strategic classification. In *Proceedings of the 2016 ACM conference on innovations in theoretical computer science*, pages 111–122, 2016.

Keegan Harris, Chara Podimata, and Zhiwei Steven Wu. Strategic apple tasting. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Johan Håstad. Clique is hard to approximate within 1- $\varepsilon$. *Acta Mathematica*, 182 (1):105–142, 1999.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192, 2007.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

Patrick J. Heagerty and Scott L. Zeger. Marginalized multilevel models and likelihood inference (with comments and a rejoinder by the authors). *Statistical Science*, 15(1):1 – 26, 2000.

Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. *Advances in neural information processing systems*, 30, 2017.

Christina Heinze-Deml and Nicolai Meinshausen. Conditional variance penalties and domain shift robustness. *Machine Learning*, 2020.

Christina Heinze-Deml, Jonas Peters, and Nicolai Meinshausen. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2), 2018.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Guy Horowitz and Nir Rosenfeld. Causal strategic classifiaction. In *International Conference on Machine Learning*. PMLR, 2023.

Daniel Hsu, Sham M. Kakade, and Tong Zhang. Random design analysis of ridge regression. *Found. Comput. Math.*, 14(3):569–600, June 2014.

Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. Does distributionally robust supervised learning give robust classifiers? In *Proceedings of the 35th International Conference on Machine Learning*, pages 2029–2037. PMLR, 2018.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger.

Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017a.

Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*, pages 3–29. Springer, 2017b.

Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *arXiv preprint arXiv:2103.10427*, 2021.

Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks, 2022.

David Hume. *An enquiry concerning human understanding: A critical edition*, volume 3. Oxford University Press on Demand, 2000.

Kenneth Hung and William Fithian. Rank verification for exponential families. *The Annals of Statistics*, 47(2):758 – 782, 2019. doi: 10.1214/17-AOS1634.

A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, May 1999.

Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. *arXiv preprint arXiv:2110.14503*, 2021.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 125–136. Curran Associates, Inc., 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Meena Jagadeesan, Celestine Mendler-Dünner, and Moritz Hardt. Alternative microfoundations for strategic classification. In *International Conference on Machine Learning*, pages 4687–4697. PMLR, 2021.

Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

Stanisław Jastrzębski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amost Storkey. On the relation between the sharpest directions of DNN loss and the SGD step length. In *International Conference on Learning Representations*, 2019.

Stanisław Jastrzębski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho*, and Krzysztof Geras*. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations*, 2020.

Stanisław Jastrzębski, Devansh Arpit, Oliver Astrand, Giancarlo B Kerg, Huan Wang, Caiming Xiong, Richard Socher, Kyunghyun Cho, and Krzysztof J Geras. Catastrophic fisher explosion: Early phase fisher matrix impacts generalization. In *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2021.

Junguang Jiang, Yang Shu, Jianmin Wang, and Mingsheng Long. Transferability in deep learning: A survey, 2022a.

Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J Zico Kolter. Assessing generalization of SGD via disagreement. In *International Conference on Learning Representations*, 2022b.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Domain extrapolation via regret minimization. *arXiv preprint arXiv:2006.03908*, 2020.

Fredrik Johansson et al. *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 0.18)*, December 2013.

Fredrik D Johansson, David Sontag, and Rajesh Ranganath. Support and invertibility in domain-invariant representations. *arXiv preprint arXiv:1903.03448*, 2019.

Pritish Kamath, Akilesh Tangella, Danica Sutherland, and Nathan Srebro. Does invariant risk minimization capture invariance? In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 4069–4077. PMLR, 13–15 Apr 2021.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

Sushrut Karmalkar and Eric Price. Compressed sensing with adversarial sparse noise via l1 regression. *arXiv preprint arXiv:1809.08055*, 2018.

Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*, pages 97–117. Springer, 2017.

Divyansh Kaushik and Zachary C. Lipton. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *EMNLP*, pages 5010–5015, 2018.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, 2020.

Zeljko Kereta and Timo Klock. Estimating covariance and precision matrices along subspaces. *Electronic Journal of Statistics*, 15:554–588, 01 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

Adam Klivans, Pravesh K. Kothari, and Raghu Meka. Efficient algorithms for outlier-robust regression. *arXiv preprint arXiv:1803.03241*, 2018.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pages 1885–1894, 2017.

Pang Wei Koh, Jacob Steinhardt, and Percy Liang. Stronger data poisoning attacks break data sanitization defenses. *arXiv preprint arXiv:1811.00741*, 2018.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021.

Vladimir Koltchinskii and Karim Lounici. Concentration inequalities and moment bounds for sample covariance operators. *Bernoulli*, 23(1):110 – 133, 2017.

Dmitry Kopitkov and Vadim Indelman. Neural spectrum alignment: Empirical study. In *Artificial Neural Networks and Machine Learning–ICANN 2020: 29th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 15–18, 2020, Proceedings, Part II 29*, pages 168–179. Springer, 2020.

Itai Kreisler, Mor Shpigel Nacson, Daniel Soudry, and Yair Carmon. Gradient descent monotonically decreases the sharpness of gradient flow solutions in scalar networks and beyond. In *Proceedings of the 40th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 23–29 Jul 2023.

Alex Krizhevsky and Geoffrey Hinton. Learning Multiple Layers of Features from Tiny Images. Technical report, Citeseer, 2009.

David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). *arXiv preprint arXiv:2003.00688*, 2020.

Frank R Kschischang. The complementary error function. Accessed 10-08-2020, 2017.

Ananya Kumar, Ruoqi Shen, Sébastien Bubeck, and Suriya Gunasekar. How to fine-tune vision models with sgd. *arXiv preprint arXiv:2211.09359*, 2022.

Frederik Kunstner, Jacques Chen, Jonathan Wilder Lavington, and Mark Schmidt. Noise is not the main factor behind the gap between sgd and adam on transformers, but sign descent might be. In *The Eleventh International Conference on Learning Representations*, 2023.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *Ann. Statist.*, 28(5):1302–1338, 10 2000. doi: 10.1214/aos/1015957395.

Q. V. Le. Building high-level features using large scale unsupervised learning. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8595–8598, May 2013.

Tosca Lechner and Ruth Urner. Learning losses for strategic classification. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI*, pages 7337–7344. AAAI Press, 2022.

Tosca Lechner, Ruth Urner, and Shai Ben-David. Strategic classification with unknown user manipulations. In *International Conference on Machine Learning*.

PMLR, 2023.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86, 1998.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521 (7553):436–444, May 2015.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE symposium on security and privacy (SP)*, pages 656–672. IEEE, 2019.

Guang-He Lee, Yang Yuan, Shiyu Chang, and Tommi S. Jaakkola. Tight certificates of adversarial robustness for randomly smoothed classifiers. In *Advances in Neural Information Processing Systems 31*, 2019.

Sagi Levanon and Nir Rosenfeld. Strategic classification made practical. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Sagi Levanon and Nir Rosenfeld. Generalized strategic classification and the case of aligned incentives. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Second-order adversarial attack and certifiable robustness. 2018a.

D. Li, Y. Yang, Y. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5543–5551, Los Alamitos, CA, USA, oct 2017. IEEE Computer Society.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, 2018b.

Ker-Chau Li and Naihua Duan. Regression Analysis Under Link Violation. *The Annals of Statistics*, 17(3):1009 – 1052, 1989.

Xinyan Li, Qilong Gu, Yingxue Zhou, Tiancong Chen, and Arindam Banerjee. Hessian based analysis of sgd for deep nets: Dynamics and generalization. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages

190–198. SIAM, 2020.

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 624–639, 2018c.

Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *Pattern Recognition*, 80, 03 2016.

Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Licong Lin and Tijana Zrnic. Plug-in performative optimization. *arXiv preprint arXiv:2305.18728*, 2023.

Zachary C Lipton, Yu-Xiang Wang, and Alex Smola. Detecting and Correcting for Label Shift with Black Box Predictors. In *International Conference on Machine Learning (ICML)*, 2018.

Bingbin Liu, Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. Analyzing and improving the optimization landscape of noise-contrastive estimation. In *International Conference on Learning Representations*, 2022.

Chang Liu, Bo Li, Yevgeniy Vorobeychik, and Alina Oprea. Robust linear regression against training data poisoning. In *AISec 2017 - Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, co-located with CCS 2017*, 11 2017.

Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, Aug 1989.

T. Liu and D. Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, March 2016.

Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the european conference on computer vision (ECCV)*, pages 369–385, 2018.

Alessio Lomuscio and Lalit Maganti. An approach to reachability analysis for feed-forward relu neural networks. *arXiv preprint arXiv:1706.07351*, 2017.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018.

Philip M Long and Hanie Sedghi. Generalization bounds for deep convolutional neural networks. *arXiv preprint arXiv:1905.12600*, 2019.

Yuzhe Lu, Zhenlin Wang, Runtian Zhai, Soheil Kolouri, Joseph Campbell, and Katia Sycara. Predicting out-of-distribution error with confidence optimal transport. In *ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML*, 2023.

Zelun Luo, Yuliang Zou, Judy Hoffman, and Li F Fei-Fei. Label efficient learning of transferable representations acrosss domains and tasks. *Advances in neural information processing systems*, 30, 2017.

Chao Ma and Lexing Ying. On linear stability of sgd and input-smoothness of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 16805–16817. Curran Associates, Inc., 2021.

Chao Ma, Daniel Kunin, Lei Wu, and Lexing Ying. Beyond the quadratic approximation: the multiscale structure of neural network loss landscapes. *arXiv preprint arXiv:2204.11326*, 2022.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 142–150, Stroudsburg, PA, USA, 2011.

Marloes H Maathuis, Markus Kalisch, Peter Bühlmann, et al. Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, 37(6A):3133–3164, 2009.

Lachlan Ewen MacDonald, Jack Valmadre, and Simon Lucey. On progressive sharpening, flat minima and generalisation. *arXiv preprint arXiv:2305.14683*, 2023.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. *arXiv preprint arXiv:2006.07500*, 2020.

Karttikeya Mangalam and Vinay Uday Prabhu. Do deep neural networks learn shallow learnable examples first? In *ICML 2019 Workshop on Identifying and*

*Understanding Deep Learning Phenomena*, 2019.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Celestine Mendler-Dünner, Frances Ding, and Yixin Wang. Anticipating performativity by predicting from predictions. *Advances in Neural Information Processing Systems*, 35:31171–31185, 2022.

William Merrill, Nikolaos Tsilivis, and Aman Shukla. A tale of two circuits: Grokking as competition of sparse and dense subnetworks. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.

Sebastian Mika. *Kernel Fisher Discriminants*. Doctoral thesis, Technische Universität Berlin, Fakultät IV - Elektrotechnik und Informatik, Berlin, 2003.

John Miller, Karl Krauth, Benjamin Recht, and Ludwig Schmidt. The effect of natural distribution shift on question answering models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6905–6916. PMLR, 13–18 Jul 2020a.

John Miller, Smitha Milli, and Moritz Hardt. Strategic classification is causal modeling in disguise. In *International Conference on Machine Learning*, pages 6917–6926. PMLR, 2020b.

Smitha Milli, John Miller, Anca D Dragan, and Moritz Hardt. The social cost of strategic classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 230–239, 2019.

Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pages 3578–3586. PMLR, 2018.

Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir Mohamed, Ndapa Nakashole, Emmanouil Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard Wang, Derry Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. Never-ending learning. In *AAAI Conference on Artificial Intelligence*, 2015.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

T. S. Motzkin and E. G. Straus. Maxima for graphs and a new proof of a theorem of turán. *Canadian Journal of Mathematics*, 17:533–540, 1965. doi: 10.4153/CJM-1965-053-6.

Luis Muñoz González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 27–38, New York, NY, USA, 2017.

Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 10–18, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

Rotem Mulayoff and Tomer Michaeli. Unique properties of flat minima in deep networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*. PMLR, 13–18 Jul 2020.

Vaishnavh Nagarajan and J Zico Kolter. Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience. *arXiv preprint arXiv:1905.13344*, 2019a.

Vaishnavh Nagarajan and J. Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019b.

Preetum Nakkiran and Yamini Bansal. Distributional generalization: A new kind of generalization. *ArXiv*, abs/2009.08092, 2019.

Preetum Nakkiran, Gal Kaplun, Dimitris Kalimeris, Tristan Yang, Benjamin L Edelman, Fred Zhang, and Boaz Barak. Sgd on neural networks learns functions of increasing complexity. *arXiv preprint arXiv:1905.11604*, 2019.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020.

Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. *Advances in neural information processing systems*, 29, 2016.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems 26*, pages 1196–1204. 2013.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science &amp; Business Media, 2012.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Jerzy Neyman and Egon Sharpe Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

Behnam Neyshabur. Implicit regularization in deep learning. *arXiv preprint arXiv:1709.01953*, 2017.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2018.

Pascal Notsawo Jr, Hattie Zhou, Mohammad Pezeshki, Irina Rish, Guillaume Dumas, et al. Predicting grokking long before it happens: A look into the loss landscape of models which grok. *arXiv preprint arXiv:2306.13253*, 2023.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.

Samet Oymak, Zalan Fabian, Mingchen Li, and Mahdi Soltanolkotabi. Generalization guarantees for neural networks via harnessing the low-rank structure of the jacobian. *arXiv preprint arXiv:1906.05392*, 2019.

Matteo Pagliardini, Martin Jaggi, François Fleuret, and Sai Praneeth Karimireddy. Agree to disagree: Diversity through disagreement for better transferability. In *The Eleventh International Conference on Learning Representations*, 2023.

Alain Pajor. Metric entropy of the grassmann manifold. *Convex Geometric Analysis*, 34:181–188, 1998.

Yan Pan and Yuanzhi Li. Toward understanding why adam converges faster than sgd for transformers. *arXiv preprint arXiv:2306.00204*, 2023.

N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman. Sok: Security and privacy in machine learning. In *2018 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 399–414, April 2018.

Vardan Papyan. The full spectrum of deepnet hessians at scale: Dynamics with sgd training and sample size. *arXiv preprint arXiv:1811.07062*, 2018.

Vardan Papyan. Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet hessians. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*. PMLR, 09–15 Jun 2019.

Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra. *The Journal of Machine Learning Research*, 21(1):10197–10260, 2020.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. pages 2233–2241, 07 2017.

Andrea Paudice, Luis Muñoz-González, and Emil C. Lupu. Label sanitization against label flipping poisoning attacks. In *ECML PKDD 2018 Workshops*, pages 5–15, Cham, 2019.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge, 2017.

Xingchao Peng, Ben Usman, Kuniaki Saito, Neela Kaushik, Judy Hoffman, and Kate Saenko. Syn2real: A new benchmark forsynthetic-to-real visual domain adaptation. *arXiv preprint arXiv:1806.09755*, 2018.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019a.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019b.

Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. Performative prediction. In *International Conference on Machine Learning*, pages

7599–7609. PMLR, 2020.

J Peters, D Janzing, and B Schölkopf. *Elements of Causal Inference-Foundations and Learning Algorithms*. The MIT Press, 2017.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society*, 2016.

Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, 34:1256–1272, 2021.

Emmanouil A Platanios, Hoifung Poon, Tom M Mitchell, and Eric Horvitz. Estimating accuracy from unlabeled data: A probabilistic logic approach. *arXiv preprint arXiv:1705.07086*, 2017.

Emmanouil Antonios Platanios, Avinava Dubey, and Tom Mitchell. Estimating accuracy from unlabeled data: A bayesian approach. In *International Conference on Machine Learning*, pages 1416–1425. PMLR, 2016.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Adarsh Prasad, Arun Sai Suggala, Sivaraman Balakrishnan, and Pradeep Ravikumar. Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018a.

Aditi Raghunathan, Jacob Steinhardt, and Percy S Liang. Semidefinite relaxations for certifying robustness to adversarial examples. *Advances in neural information processing systems*, 31, 2018b.

Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018a.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? 2018b.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5389–5400. PMLR, 09–15 Jun 2019.

Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *J. Mach. Learn. Res.*, 19(1):1309–1342, jan 2018.

Elan Rosenfeld and Andrej Risteski. Outliers with opposing signals have an outsized effect on neural network optimization. In *The Twelfth International Conference on Learning Representations*, 2024.

Elan Rosenfeld and Nir Rosenfeld. One-shot strategic classification under unknown costs. In *International Conference on Machine Learning (ICML)*, 2024.

Elan Rosenfeld, Santosh Vempala, and Manuel Blum. Human-usable password schemas: Beyond information-theoretic security. *arXiv preprint arXiv:1906.00029*, 2019.

Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, and Zico Kolter. Certified robustness to label-flipping attacks via randomized smoothing. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8230–8241. PMLR, 13–18 Jul 2020.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. The risks of invariant risk minimization. In *International Conference on Learning Representations*, 2021.

Elan Rosenfeld, Preetum Nakkiran, Hadi Pouransari, Oncel Tuzel, and Fartash Faghri. Ape: Aligning pretrained encoders to quickly learn aligned multimodal representations. *arXiv preprint arXiv:2210.03927*, 2022a.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. Domain-adjusted regression or: Erm may already learn features sufficient for out-of-distribution generalization. *arXiv preprint arXiv:2202.06856*, 2022b.

Elan Rosenfeld, Pradeep Ravikumar, and Andrej Risteski. An online learning approach to interpolation and extrapolation in domain generalization. In *International Conference on Artificial Intelligence and Statistics*, pages 2641–2657. PMLR, 2022c.

Dominik Rothenhäusler, Nicolai Meinshausen, Peter Bühlmann, and Jonas Peters. Anchor regression: Heterogeneous data meet causality. *Journal of the Royal Statistical Society Series B*, 83(2):215–246, 2021.

Benjamin I.P. Rubinstein, Blaine Nelson, Ling Huang, Anthony D. Joseph, Shinghon Lau, Satish Rao, Nina Taft, and J. D. Tygar. Antidote: Understanding and defending against poisoning of anomaly detectors. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, pages 1–14, New York, NY, USA, 2009.

Mark Rudelson, Roman Vershynin, et al. Hanson-wright inequality and sub-gaussian concentration. *Electronic Communications in Probability*, 18, 2013.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Sorawit Saengkyongam, Elan Rosenfeld, Pradeep Kumar Ravikumar, Niklas Pfister, and Jonas Peters. Identifying representations for intervention extrapolation. In *The Twelfth International Conference on Learning Representations*, 2024.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020a.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *Proceedings of the 37th International Conference on Machine Learning*, 2020b.

Shiori Sagawa, Pang Wei Koh, Tony Lee, Irena Gao, Sang Michael Xie, Kendrick Shen, Ananya Kumar, Weihua Hu, Michihiro Yasunaga, Henrik Marklund, Sara Beery, Etienne David, Ian Stavness, Wei Guo, Jure Leskovec, Kate Saenko, Tatsunori Hashimoto, Sergey Levine, Chelsea Finn, and Percy Liang. Extending the wilds benchmark for unsupervised adaptation. In *NeurIPS Workshop on Distribution Shifts*, 2021.

Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv:1611.07476*, 2016.

Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. *arXiv preprint arXiv:2008.04859*, 2020.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 5014–5026. Curran Associates, Inc., 2018.

Mattia Segù, Alessio Tonioni, and Federico Tombari. Batch normalization embeddings for deep domain generalization. *arXiv preprint arXiv:2011.12672*, 2020.

Seonguk Seo, Yumin Suh, Dongwan Kim, Geeho Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. *arXiv preprint arXiv:1907.04275*, 2019.

D. Serre. *Matrices: Theory and Applications*. Graduate Texts in Mathematics. Springer, 2010. ISBN 9781441930101.

Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems 31*, pages 6103–6113. 2018.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

Han Shao, Avrim Blum, and Omar Montasser. Strategic classification under unknown personalized manipulation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016.

Yanyao Shen and Sujay Sanghavi. Learning with bad training data via iterative trimmed loss minimization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 5739–5748, Long Beach, California, USA, 09–15 Jun 2019.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. *Advances in neural information processing systems*, 31, 2018.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33, 2020.

Daniel A Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)*, 51(3):385–463, 2004.

Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3520–3532, USA, 2017.

Petar Stojanov, Zijian Li, Mingming Gong, Ruichu Cai, Jaime G. Carbonell, and Kun Zhang. Domain adaptation with invariant representation learning: What transformations to learn? In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

Adarsh Subbaswamy, Peter Schulam, and Suchi Saria. Preventing failures due to dataset shift: Learning predictive models that transport. In *The 22nd International*

*Conference on Artificial Intelligence and Statistics*, pages 3118–3127. PMLR, 2019.

Arun Suggala, Adarsh Prasad, and Pradeep K Ravikumar. Connecting optimization and regularization paths. In *Advances in Neural Information Processing Systems*, pages 10608–10619, 2018.

Arun Sai Suggala and Praneeth Netrapalli. Online non-convex learning: Following the perturbed leader is optimal. In *Algorithmic Learning Theory*, pages 845–861. PMLR, 2020.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*. Springer, 2016.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016a.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016b.

Ravi Sundaram, Anil Vullikanti, Haifeng Xu, and Fan Yao. PAC-learning for strategic classification. In *International Conference on Machine Learning*, pages 9978–9988. PMLR, 2021.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013a.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013b.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Remi Tachet des Combes, Han Zhao, Yu-Xiang Wang, and Geoffrey J Gordon. Domain adaptation with conditional distribution matching and generalized label shift. *Advances in Neural Information Processing Systems*, 33, 2020.

Rahim Taheri, Reza Javidan, Mohammad Shojafar, Zahra Pooranian, Ali Miri, and Mauro Conti. On defending against label flipping attacks on malware detection systems. *arXiv preprint arXiv:1908.04473*, 2019.

Damien Teney, Ehsan Abbasnejad, Kushal Kafle, Robik Shrestha, Christopher Kanan, and Anton Van Den Hengel. On the value of out-of-distribution testing:

An example of goodhart's law. *Advances in neural information processing systems*, 33:407–417, 2020.

Vimal Thilak, Etai Littwin, Shuangfei Zhai, Omid Saremi, Roni Paiss, and Joshua Susskind. The slingshot mechanism: An empirical study of adaptive optimizers and the grokking phenomenon. *arXiv preprint arXiv:2206.04817*, 2022.

Sebastian Thrun. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer, 1998.

Jin Tian and Judea Pearl. Causal discovery from changes. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 512–521, 2001.

Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. *arXiv preprint arXiv:1711.07356*, 2017.

Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems 31*, pages 8000–8010. 2018.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. *Advances in neural information processing systems*, 31, 2018.

Jonathan Uesato, Brendan O'donoghue, Pushmeet Kohli, and Aaron Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pages 5025–5034. PMLR, 2018.

Guillermo Valle-Perez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *International Conference on Learning Representations*, 2019.

V.N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages

5385–5394, 2017a.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017b.

R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Y. Eldar and G. Kutyniok, editors, *Compressed Sensing, Theory and Applications*, chapter 5, pages 210–268. Cambridge University Press, 2012.

Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. In *Advances in neural information processing systems*, pages 5334–5344, 2018.

Yoav Wald, Amir Feder, Daniel Greenfeld, and Uri Shalit. On calibration and out-of-domain generalization. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

Haoxiang Wang, Haozhe Si, Bo Li, and Han Zhao. Provable domain generalization via invariant-feature subspace recovery. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23018–23033. PMLR, 17–23 Jul 2022a.

Shengjie Wang, Abdel-rahman Mohamed, Rich Caruana, Jeff Bilmes, Matthai Plilipose, Matthew Richardson, Krzysztof Geras, Gregor Urban, and Ozlem Aslan. Analysis of deep neural networks with extended data jacobian matrix. In *International Conference on Machine Learning*, pages 718–726. PMLR, 2016.

Shiqi Wang, Yizheng Chen, Ahmed Abdou, and Suman Jana. Mixtrain: Scalable training of verifiably robust neural networks. *arXiv preprint arXiv:1811.02625*, 2018a.

Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Efficient formal safety analysis of neural networks. *Advances in neural information processing systems*, 31, 2018b.

Zixuan Wang, Zhouzi Li, and Jian Li. Analyzing sharpness along GD trajectory: Progressive sharpening and edge of stability. In *Advances in Neural Information Processing Systems*, 2022b.

Stefan Webb, Tom Rainforth, Yee Whye Teh, and M Pawan Kumar. A statistical approach to assessing neural network robustness. *arXiv preprint arXiv:1811.07209*, 2018.

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. Sharpness minimization algorithms do not only minimize sharpness to achieve better generalization. *arXiv preprint arXiv:2307.11007*, 2023.

Lily Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. Towards fast computation of certified robustness for relu networks. In *International Conference on Machine Learning*, pages 5276–5285. PMLR, 2018a.

Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578*, 2018b.

Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, 1996.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*, pages 5286–5295. PMLR, 2018.

Eric Wong, Frank Schmidt, Jan Hendrik Metzen, and J Zico Kolter. Scaling provable adversarial defenses. *Advances in Neural Information Processing Systems*, 31, 2018.

Lei Wu, Chao Ma, and Weinan E. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Lei Wu, Mingze Wang, and Weijie J Su. The alignment property of SGD noise and how it helps select flat minima: A stability analysis. In *Advances in Neural Information Processing Systems*, 2022.

Han Xiao, Huang Xiao, and Claudia Eckert. Adversarial label flips attack on support vector machines. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 870–875, Amsterdam, The Netherlands, The Netherlands, 2012.

Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. Is feature selection secure against training data poisoning? In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1689–1698, Lille, France, 07–09 Jul 2015.

Kai Yuanqing Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. In *International Conference on Learning Representations*, 2021.

Chuanlong Xie, Fei Chen, Yue Liu, and Zhenguo Li. Risk variance penalization:

From distributional robustness to causality. *arXiv preprint arXiv:2006.07544*, 2020.

Chen Xing, Devansh Arpit, Christos Tsirigotis, and Yoshua Bengio. A walk with sgd. *arXiv preprint arXiv:1802.08770*, 2018.

Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen. Generative poisoning attack method against neural networks. *arXiv preprint arXiv:1703.01340*, 2017.

Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer point selection for explaining deep neural networks. In *Advances in Neural Information Processing Systems 31*, pages 9291–9301. 2018.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems 27*, pages 3320–3328. 2014.

Y. Yu, T. Wang, and R. J. Samworth. A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 04 2014.

Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2100–2110, 2019.

Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. Efficient defenses against adversarial attacks. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 39–49, 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Hanrui Zhang and Vincent Conitzer. Incentive-aware PAC learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. *Advances in neural information processing systems*, 31, 2018.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33: 15383–15393, 2020.

Kun Zhang, Mingming Gong, and Bernhard Scholkopf. Multi-source domain adaptation: a causal view. In *Proceedings of the Twenty-Ninth AAAI Conference* *on Artificial Intelligence*, pages 3150–3157, 2015.

Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*. PMLR, 2019.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7523–7532. PMLR, 09–15 Jun 2019.

Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P Adams, and Peter Orbanz. Non-vacuous generalization bounds at the imagenet scale: a pac-bayesian compression approach. *arXiv preprint arXiv:1804.05862*, 2018.

Chen Zhu, W. Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 7614–7623, Long Beach, California, USA, 09–15 Jun 2019a.

Xingyu Zhu, Zixuan Wang, Xiang Wang, Mo Zhou, and Rong Ge. Understanding edge-of-stability training dynamics with a minimalist example. In *The Eleventh International Conference on Learning Representations*, 2023.

Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. In *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 7654–7663. PMLR, 09–15 Jun 2019b.