Objective Criteria for Explainable Machine Learning

Chih-Kuan Yeh

AUGUST 2022
CMU-ML-22-106

Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
USA

Thesis Committee:

Pradeep Ravikumar, *Chair*	CARNEGIE MELLON UNIVERSITY
Ameet Talwalkar	CARNEGIE MELLON UNIVERSITY
Hima Lakkaraju	HARVARD UNIVERSITY
Mukund Sundararajan	GOOGLE LLC
Bin Yu	UNIVERSITY OF CALIFORNIA, BERKELEY

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

This research was supported by: National Science Foundation awards IIS1664720, IIS1955532 and OAC1934584; Office of Naval Research award N000141812861; and a grant from Rakuten, Inc.

Keywords: Deep Learning, Explainable AI (XAI), Feature Importance, Data Importance, Concept Importance, Model Interpretation, Algorithmic Game Theory, Post-hoc Explanations

To my family and friends.

Abstract

As deep learning methods have obtained tremendous success over the years, our understanding of these models has yet to keep up with the development the models. Explainable machine learning is one of the main research fields dedicated to understanding complex machine learning models. While there are ever-increasing proposed instances of explanations, the evaluation of explanations has been an open question. Evaluations involving humans are expensive in the development phase of explanations. To address the difficulty to involve human-in-the-loop during the design of explanations, this thesis aims to define objective criteria which allow one to measure some goodness property explanations without humans and design explanations that are desirable with respect to the objective criteria.

In this thesis, we discuss different criteria for making evaluating explainable AI methods more objective, where our methods can mainly be categorized in three prongs: (a) faithfulness-oriented (b) theoretically-motivated (c) application-driven. A faithfulness-oriented metric is usually connected to the core concept that an explanation of the model should faithfully "explain" the model. Theoretically-motivated objective criteria usually have the form "when the model and data satisfy a certain property, the explanation should satisfy a corresponding property". Application-driven objective criteria simulate quantitatively how explanations can help in certain applications without humans. We design objective criteria for different types of explanations and use these objective criteria to guide the design of new explanations. Finally, some human studies are done to verify the design of these new explanations.

Acknowledgments

I would like to thank my advisor Pradeep Ravikumar, for his crucial role in my Ph.D. journey. If there was one moment that defined my Ph.D. and research philosophy, it is the one time when Pradeep stopped me when I tried to resubmit a rejected paper to an upcoming conference. He asked me to add a large portion of new context to significantly improve the submission, and I initially was reluctant to do so as it would have taken me at least a couple of months to do all the work. Years later, I am grateful for Pradeep's persistence in asking me to always have the highest standard for my research, as it has gradually become part of my philosophy of research. I would like to thank Pradeep deeply for not only his great ideas and inspiration during our discussions but also for teaching me how to become the researcher I am through his demonstration.

I would also like to thank my many research mentors during my journey of research. I would like to thank my first research mentor during undergraduate study, Ho-Lin Chen, for leading me into the world of computer science research. He was a role model that made doing research sound cool to me, which made me decide to pursue a Ph.D. I would like to thank Hsuan-Tien Lin and Yu-Chiang Frank Wang for leading me into the world of machine learning research. I am indebted to them to be fortunate to pursue my Ph.D. degree in the machine learning department of CMU, which has been a great journey for me.

My two internships at Google have played a crucial role in my Ph.D., where I have worked on projects regarding concept explanations and instance-based explanations. I have had the chance to understand and think deeply about how explainable machine learning can be used in an industrial setting, which greatly affected my mindset and viewpoint for explainable machine learning research. I am grateful to all the great internship mentors I have had during my Ph.D.: Been Kim, Sercan Arik, Tomas Pfister, Ankur Taly, and Mukund Sundararajan.

I would also like to thank my thesis committee members, Ameet Talwalkar, Hima Lakkaraju, Mukund Sundararajan, and Bin Yu, who have provided invaluable suggestions and advice for my dissertation. Their impactful research in the field of explainable machine learning has greatly inspired my research. I would also like to thank the administrative staff at the machine learning department, Diane Stidle and Sharon Cavlovich, who have been extremely helpful and patient.

I am grateful to my several collaborators/ mentors during my Ph.D., whose fruitful discussions were invaluable to my study, Joon Sik Kim, Arun Sai Suggala, David Inouye, Cheng-Yu Hsieh, Ian En-Hsu Yen, Jianshu Chen, Chengzhu Yu, Dong Yu, Xuanqing Liu, Seungyeon Kim, Sanjiv Kumar, Cho-Jui Hsieh, Chung-Wei Lee, Wei Fang, Kuan-Yun Lee, Frederick Liu, Che-Ping Tsai, Biswajit Paria, Ning Xu, Barnabás Póczos, Chun-Liang Li, Ching-Yuan Bai, Chung-Wei Lee, Wei Fang, Hong-Min Chu. Many of these collaborators are also great personal friends of mine, which made the research discussions so much more enjoyable. I am also grateful to my friends in the machine learning department including Hubert (Yao-Hung), Shaojie, Will, Greg, Leqi, Sebastian, Kin, Siddarth, Amanda, Kartik, Nicholay, Devendra,

Tim, Bingbin, Adarsh, Xun, Dan, Otilia, Tom, Helen, Paul, Juyong, Charvi, Ivan, Tiffany, Po-Wei, Chieh, Fish, Youngseog, Yewen, and Rattana. I am also grateful for all my personal friends (too many to list) that made my Ph.D. study duration such joyful.

I am grateful to my parents Jack and Marie for their unconditional support on any path that I have chosen to pursue, and my sister Stephanie for the mental support during my study. I have always known that my family will stand by me whatever I do, which made my bumpy ride in Ph.D. study so much smoother.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The field of explainable artificial intelligence (XAI) is concerned with the task of explaining machine learning models, which has attracted increasing attention as the complexity of modern machine learning models grows. The need to explain the complicated internal workings of machine learning models has also grown significantly, especially when the machine learning models are applied to high-stakes decisions, including finance, law, medical, social applications, and autonomous driving. In these applications, explanations of high-stakes decisions are useful to understand and debug the models, enhance users' trust in the model, clarify accountability of the model, and communicate with models on human-AI collaborations. For instance, a doctor using AI to help with diagnosis would benefit by understanding how the AI predicted to determine whether to trust it. In social applications, it is also crucial to understand why the models make certain decisions to examine whether the algorithm is fair or not. Furthermore, the General Data Protection Regulation claims that data protection authorities have the right to be explained the output of the algorithm [123].

One crucial difficulty to explain a machine learning model is that the term "explain" or "interpretability" is not well-defined. Most current explanations explain some type of "property" of the complex model that can be digested by humans. Some common properties include but are not limited to the most salient data input features used by the model, the most salient training data used by the model, the most salient human understandable concept used by the model, and how to alter the feature of a data point to change the prediction of the model. However, there are many different explanations with contradicting philosophies. For instance, given an image classifier, the crucial pixels of the image classifier may be considered a good explanation to certain users as it shed light on how the model makes its prediction, but may also be considered not interpretable as the most salient features may not be sufficient to infer the reasoning rationale of the model. One can perform human studies and interviews and ask users to select the most interpretable algorithms among a given set of different explanations, which is related to metric elicitation in fairness [27, 75]. However, asking humans to select the most interpretable explanations may also have its flaw. Human is well-known to suffer from confirmation bias, and an explanation may seem to be interpretable but is unrelated to a model. Recent works have even shown that many key explanations disagree with each other, and the users may decide which explanations to use based on personal preference [92]. How does one choose the right "property" of a complex machine learning model to explain?

An alternative to gauge the effectiveness of explanations may be through evaluating the usefulness of explanations along with humans in applications, which is suggested by many recent works. Doshi-Velez and Kim [39], Murdoch et al. [116] have proposed to evaluate explanations in real-world applications involving human users and test how explanations may aid users in real-world applications. Similarly, Chen et al. [25] encourages interpretable machine learning problems to be more closely connected to target use cases, and simulations based on a simulated version of the real task are suggested to be considered. While such evaluation is based on real applications, it may be expensive to utilize such types of evaluations, especially in the development stages of explanations, as the evaluation often requires the involvement of real humans. Therefore, a sensible functional-grounded evaluation may be useful for designing/ choosing an explanation to be used, and application-driven evaluations could be used to validate that the designed/ chosen explanation can aid humans in real-world applications or simulated use cases. We name this class of functional-grounded evaluation as objective criteria, mainly since it does not require actual human involvement during the evaluation stage.

In this thesis, we mainly consider three classes of objective criteria (functional-grounded evaluations): (1) faithfulness-motivated objective criteria, which are motivated by how well can the explanation describe the model (2) application-motivated objective criteria, which are motivated by how the explanations can be used in real-world applications (3) theoretically-motivated axiomatic criteria, which aid the design for explanations by certain theoretical property of explanations. In the following, we discuss these three classes of objective criteria in deeper detail.

Faithfulness-motivated Objective Criteria One class of functional-grounded evaluations is based on how faithfully the explanation explains a given model and is also termed the faithfulness of explanations or descriptive accuracy [116]. The faithfulness of explanations are crucial as the "faithfulness" is usually difficult to be measured by human – human may prefer explanations that look visually appealing but is unrelated to the model to be explained. One form of objective criteria is based on the question "does the explanation explain this model?". The core idea for these evaluations is to identify properties that a faithful explanation should satisfy, and perform tests on the model-explanation pair to verify if the properties are satisfied. It is also called descriptive accuracy by Murdoch et al. [116], as it measures how accurately an explanation explains the model. For instance, many explanations are linear approximations within a local neighborhood, and faithfulness metrics measure how well the explanations approximate the model in the local neighborhood. One popular example by Adebayo et al. [2] is to design a sanity check for explanations that changing the model weights randomly should also change the resulting explanations. Surprisingly, not all explanations pass this sanity check convincingly, which may imply some explanations are not faithful to the model.

Application-motivated Objective Criteria A different form of objective criteria is based on applications related to the explanation, especially those that applications where human involvement is not required or could be simulated automatically. For instance, finding hurtful training examples is a key application for example-based explanations, and certain evaluations for example-based explanations involve removing the hurtful training examples based on the explanations and retraining the model and measuring the new model performance. As human involvement is not

necessary for such applications, the remove-and-retrain evaluation has been a crucial objective criterion for example-based explanations motivated by real-world applications.

Theoretical-motivated Objective Criteria An alternative form of functional-grounded evaluation for explanations is theoretical properties in the form of axioms. Axioms can be seen as theoretical constraints of how explanations should behave in certain specific inputs. If the machine learning model to explain has some desired property, one would hope that such desired property may be reflected in the explanations. Such constraints of explanations are called axiomatic properties. For example, if the machine learning models are completely symmetric in two features, and the two features have the same value for some given input, the explanation value for these two features to this input should be the same. This is the symmetric axiom that is widely used for explanation methods. Perhaps the most common line of work in incorporating axioms to design explanations is the family of Shapley values [139], which originated from the cooperative game theory community.

The goal of this thesis is to develop and define meaningful objective criteria and use these objective criteria to aid our design for different types of explanations. As different explanation types would naturally follow different objective criteria, we aim to design objective criteria for a wide variety of explanation types, including feature-importance explanations, feature-set explanations, feature interaction importance explanations, example importance explanations, and concept-based explanations.

1.1 Background

We first briefly introduce some classes of explanation types in the field of interpretable machine learning. One way to classify explanations is regarding the model to be explained by explanations. Post-hoc explanations aim to explain an arbitrarily given model, while self-interpretable models embed some form of explainability in the model design during training. Another classification of explanations is in the form of explanations, including the importance of individual features, the importance of individual training examples, counterfactual explanations, and concept explanations.

Post-hoc Explanations Consider the following general supervised learning setting: input space $\mathcal{X} \subseteq \mathbb{R}^d$, an output space $\mathcal{Y} \subseteq \mathbb{R}$, and a (machine-learnt) predictor $\mathbf{f} \in \mathcal{F} : \mathbb{R}^d \mapsto \mathbb{R}$, which at some test input $\mathbf{x} \in \mathbb{R}^d$, predicts the output $\mathbf{f}(\mathbf{x})$. Local post-hoc explanations for machine learning models explains the function $\mathbf{f}(\mathbf{x})$, while global post-hoc explanations for machine learning models explains the function $\mathbf{f}(\mathbf{x})$ for $\mathbf{x} \in \mathbb{X}$. Many explanation types for machine learning models explain the target $\mathbf{f}(\mathbf{x})$ by an importance vector for v units, and the explanation can be seen as a function that takes the machine learning model $\mathbf{f}$ and the input $\mathbf{x}$ as input, and outputs the importance for each of the v units $\Phi : \mathcal{F} \times \mathcal{X} \mapsto \mathbb{R}^v$. Post-hoc explanations have the benefit to be very widely applicable to different model types and can be applied to models when the user does not have control over the training process.

Self-interpretable Models Self-interpretable models embed some form of inherent interpretability in the model design. Consider the following general supervised learning setting: input space

$\mathcal{X} \subseteq \mathbb{R}^d$, an output space $\mathcal{Y} \subseteq \mathbb{R}$, an importance-based self-interpretable model outputs both the prediction and the explanation associated with the prediction $\mathbf{f} \in \mathcal{F} : \mathbb{R}^d \mapsto \mathbb{R} \times \mathbb{R}^v$, where the explanation here is a v-dimension importance vector. Some common self-interpretable models include sparse linear models [21, 161], decision trees or decision lists [32, 42, 170], case-based models [86, 117]. Rudin [131] has argued to use self-interpretable models for explainability instead of post-hoc explanations, as inherently interpretable models may have more faithful explanations compared to post-hoc explanations which may suffer from not being faithful to the model [2]. However, a self-interpretable model may be limited since it may not be straightforward how state-of-the-art deep learning architecture can be altered to be inherently interpretable, thus limiting its use cases. Moreover, while it is argued that there is no necessary trade-off between interpretability and accuracy, most self-interpretable models are not tested on state-of-the-art architecture possibly due to the expensive tuning required.

Feature Importance Explanations Given a machine learning model with input $\mathbb{R}^d$ trained on a set of training data with d features, we may explain a specific prediction by showing a heatmap of feature importance weights. Such explanations are called feature importance explanations, as they output explanations of how important each feature is with respect to the prediction $\mathbf{f}(\mathbf{x})$. However, there have been different prongs of feature importance explanations, as there may be several different definitions for the "importance" of features. One popular idea of feature importance is based on how perturbing the value of the feature will affect the model output $\mathbf{f}(\mathbf{x})$, which resembles the definition of gradient $\nabla \mathbf{f}(\mathbf{x})$, which is a popular feature importance [11, 144, 150]. As shown by Ancona et al. [5], many recent explanations such as ϵ-LRP [10], Deep LIFT [142], and Integrated Gradients [156] can also be seen as variants of gradient explanations.

Another class of feature importance explanations is based on perturbing the feature values and measuring the prediction difference. In this line of work, [45, 187] use perturbations with grey patch occlusions on CNNs, while [20, 124, 193] improve upon these perturbations via generative models and advanced smoothing designs. Another way to view perturbation-based features is to first select a local neighborhood of interest, learn a linear model in the neighborhood and use the parameter of each feature as importance [129]. This has been further applied to different domains by [126, 186] and different local neighborhood weights [104] by subsequent works.

Example Importance Explanations Example importance explanations attribute the model prediction to individual training examples that are used to train the model $\mathbf{f}(\mathbf{x})$ in the training process. Prototype selection methods [14, 85, 86] explain a model by constructing the model with a (usually Bayesian) case-based reasoning classifier which provides a set of "representative" samples from the training data set during the prediction. Kim et al. [87] additionally provides criticism alongside the prototypes to explain what is not captured by prototypes via greedy selection. Koh and Liang [89] provide approximations to estimating the influence of training samples defined as how down-weighting the training example would affect prediction loss via gradient of the training point and the sum of hessian of all training data. This was further accelerated by [65] by using a k-nearest-neighbors-based selection over training samples. [6] use a graph over the training samples to select influential training samples. [128] propose to estimate the training data importance by the gradient product between the training point and

testing point along the training trajectory, which approximates the change of testing point loss during the training process.

Counterfactual and Contrastive Explanations Counterfactual explanations [35, 58, 74, 80, 167] answer the question of what to alter in the current input to change the model outcome. Such a contrastive perspective is very amenable to interactive explanations that enable users to understand the model [80, 127]. Counterfactual explanations are related to adversarial examples [18, 56] as they both try to find small perturbations to the test input that changes the model prediction [168], while the distinction is that counterfactual explanations usually target a . Xu et al. [176] add group sparsity regularization to improve the semantic structure of such adversarial perturbations. Ribeiro et al. [130] deems a set of features important if the model prediction does not change a lot when only perturbing features are not in the set.

Concept Explanations Concepts are human-defined descriptions that match a set of the training example. For instance, the "white" concept can be matched to any training example where the main object is mostly white, and "stripe" can be matched to any training example that contains a stripe. These training examples that match the concepts are usually labeled by a human. Given these labeled concepts and the associated training examples, concept-based explanations aim to provide human-centered explanations which answer the question "does this human understandable concept relates to the model prediction?"[88, 192]. Some follows-up for concept-based explanations include when are concept sufficient to explain a model [181], computing interventions on concepts for post-hoc models [57] and self-interpretable models [90], combining concept with other feature attributions [138], unsupervised discovery of concepts [49, 53, 182], and debiasing concepts [12], and addressing the possibility that concept does not necessarily lie in the linear subspace of some activation layer [26] by a self-interpretable model.

1.2 Our Contributions

In this thesis, we mainly target post-hoc explanations as they are currently much more widely applicable to different model architecture, but many objective criteria that we developed in this thesis may also be applied to self-interpretable models. Our goal is to develop meaningful objective criteria that can be used to evaluate explanations of a certain type, and use the objective criteria to guide our design for new explanations. We apply application-based human evaluation in certain works to verify the designed explanations are useful in certain applications. Our designed objective criteria span different explanation types including feature-importance explanations, feature-set explanations, feature interaction importance explanations, example importance explanations, and concept-based explanations.

Unified Faithfulness for Feature Importance Explanations [180] In chapter 2, we propose a faithfulness-motivated objective criterion for feature importance explanations. The objective criterion is the infidelity measure, which captures how well a feature importance explanation (one feature importance value per feature) matches the given model. The idea for infidelity is that if

we perturb the input by some specific perturbations, the model output will change accordingly. We find that for gradient-based feature explanations and perturbation-based feature explanations, the explanations can be seen as a linear approximation of the model with a certain perturbation neighborhood of the input. Thus, the infidelity measures the expected difference between the model output perturbation, and its approximation using the explanation. Note that this general setup allows for any choice of significant input perturbations, and we can evaluate different forms of explanations when human users are interested in a certain perturbation. It moreover holds that the optimal explanation for minimizing infidelity can be calculated in closed form. This in turn allows one to propose new explanations by introducing new classes of perturbations. Finally, we verify the usefulness of infidelity in a certain application where the perturbation of interest is clear, and the optimal explanation of infidelity can be the most useful explanation via human studies.

Axiomatic Characterization for Value Function of Feature Importance Explanations [184] Despite the success of feature importance explanations, there is some hidden design choice for feature importance that may not be clear in a different setting. One of the hidden design choices is the value function, a function that transforms the machine learning model to a set function such that perturbation-based explanations can be applied. In chapter 3, we visit a specific feature explanation motivated through theoretic properties – the Shapley value [31, 104, 139]. However, there has been an ongoing debate on what is the right value function to use along with the Shapley value, as the theoretic properties of Shapley value do not cover the value functions, as they are developed in the setting of set functions. To tackle this problem, we choose to provide theoretical-motivated objective criteria for value functions of Shapley value by introducing axioms to value functions. We show that a new value function – Joint Baseline Shapley Value – can uniquely satisfy a set of axioms designed for the value function when used along with the Shapley value explanation.

Robustness-Based Feature Set Explanations [77] In chapter 4, we design objective criteria for feature set explanations. Here, feature set explanations can be seen as the top-k most important features based on feature importance score. As it seems less feasible to extract model perturbation sensitivity from a numerical standpoint as above, we propose faithfulness-motivated objective criteria that measure robustness to perturbations of a set of features instead. The key idea is that if we get the important feature set right, then the model will be robust to perturbations restricted to the complement of these important features (namely, the unimportant features): changing these unimportant features should not change the model output too much. Similarly, if we restrict the perturbations to only the important features, we expect the model to be very non-robust to such perturbations: changing the truly important features should change the model outputs a lot. Drawing from the test-time adversarial robustness literature, they measure robustness using the maximum function difference under such perturbations. We then propose scalable greedy algorithms that optimize this robustness criterion, to extract the "important features". The feature set feature importance method is also baseline-free, which may be useful in cases where the right baseline is not obvious.

Completeness Axiom for Example-Based Explanations [179] In section 5, we focus on a completely different type of explanation – example-based explanations. For example-based explanations, where the goal is to explain the model prediction at a test input in terms of contributions of different training points, we focus on a simple objective criterion – the sum of the contributions of different training points should sum up to the model prediction at the test point. We then design a "representer theorem," that provides just such a decomposition for general deep neural networks, with the slight addendum that the model is trained till convergence with a small ℓ_2 regularization weight. The representer theorem is a natural way to decompose the function value of the test point to training data points in a way that aligns with the training of the model, and we show that our resulting algorithm is very efficient to apply to real-world applications.

Sufficiency and Attribution for Concept Explanations [182] In section 6, we focus on objective criteria for concept-based explanations, which are arguably closest to how humans reason [88]. Existing concept-based explanations do not account for whether a set of concepts are sufficient for the model prediction, which may lead to a set of relevant concepts that cannot fully explain the model, which may not be fully trusted. To overcome this shortcoming, we propose an objective completeness evaluation metric, "completeness", which measures how sufficient are a set of concepts to explain the prediction of a model by assuming the concept scores of "sufficient concepts" will be sufficient statistics to the model prediction. We then use game theoretic notions to aggregate the importance of each concept by how much they contribute to the completeness score, which is also motivated by the axiomatic properties of the importance of each concept. We then propose a concept discovery algorithm to discover concepts that are both complete and interpretable by optimizing the completeness score along with regularizations to ensure the concepts are interpretable, which we validate our experiment results through qualitative and quantitative studies on both image and language datasets.

Axiomatic-motivated Feature Interaction Explanations [162] In section 7, we focus on objective criteria for feature interaction importance explanations, which assign importance to not only the features but also the interaction between features. The rationale for feature interaction explanation is that certain features are only crucial to the model output when some complementary feature exist, and thus simple feature importance explanation may not reflect the interaction effect between features for a model. One natural solution is to assign importance to both feature and feature interactions, such that one can gauge whether a feature is helpful to the model on its own or by co-occurring with another feature. We then provide theoretically-motivated objective criteria in the form of axioms for feature interactions. We also consider a generalized version of feature infidelity as one of the properties that feature interactions should satisfy. By combining the constraint that feature interaction should be faithful to the model along with a natural set of well-known axioms, we can derive novel and unique feature interactions which we term Faith-Shap.

Layer Selection Based on Fixing Evaluation for Example-Based Explanations [185] In section 8, we revisit objective criteria for example-based explanations and propose a new application-driven objective criterion for example-based explanations. One of the main use cases for example-

based explanations is to find helpful and hurtful training examples for specific test points, especially when the test points are mispredicted by the model. We, therefore, propose to measure whether removing hurtful training examples can actually fix the testing example's prediction. This evaluation was never conducted by prior example-based explanations perhaps due to the computational cost. While many existing influence functions use the last layer weight parameters as an approximation, we find that this approximation leads to a significant performance drop on the fixing evaluation, which can be attributed to the "cancellation effect". Instead, we show that by using the first layer weight parameters, we can reduce the cancellation effect, and verify that the resulting example-based explanations achieve much better results on the fixing evaluation.

1.2.1 Other Contributions

This thesis includes a coherent line of works (either submitted work or accepted conference/journal papers) in explainable AI that designs explanations based on objective criteria, we also briefly list some other contributions during the graduate study that were not discussed in the thesis:

Unsupervised Speech Classification [178]: We study how to do unsupervised speech classification (phoneme classification) based on the empirical distribution. The key idea is to first cluster and segment the phonemes with unsupervised techniques, and match the n-gram statistics of unsupervised phonemes with the n-gram statistics of phonemes based on texts. We then propose to iterate between the segmentation components and the unsupervised components to reach stronger results.

Zero-shot Multi-Label Classification [99]: We study how using a knowledge graph neural network can aid in predicting the unseen labels for multi-label classification, which is also called generalized zero-shot learning. The key idea is to build a label graph based on our knowledge graph and to learn how label predictions can propagate in the graph during training. The propagation can then be applied to unseen labels during test time to achieve zero-shot learning.

Learning Sparse Representations for Faster Retrieval [122]: We study learning sparse representation to optimize the retrieval time (FLOPS) for the representation. While most representation learning aims to find a dense low-dimension embedding that can be optimized for efficient retrieval, we propose to learn a sparse high-dimension embedding to achieve even more efficient retrieval. Our theoretical analysis shows that to achieve efficient retrieval, each dimension of the embedding space should be equally activated. We incorporate this insight in our algorithm to learn sparse representation, which we show to achieve faster retrieval speed.

Survey on Concept-Based Explanations [183]: We survey recent advances for concept-based explanations and discuss the connections of these advances to other fields of explanations. For instance, the saliency of concept-based methods is often inspired by advances in feature attribution, and the what-if questions involving concepts are often related to the counterfactual class of explanations. We then discuss some major problems in concept explanations and end with some examples of how concept explanations are used in various science fields.

Chapter 2

Feature Importance: Infidelity and Sensitivity

We consider the task of how to explain a complex machine learning model, abstracted as a function that predicts a response given an input feature vector, given only black-box access to the model. A popular approach to do so is to attribute any given prediction to the set of input features: ranging from providing a vector of importance weights, one per input feature, to simply providing a set of important features. For instance, given a deep neural network for image classification, we may explain a specific prediction by showing the set of salient pixels, or a heatmap image showing the importance weights for all the pixels. But how good is any such explanation mechanism? We can distinguish between two classes of explanation evaluation measures [94, 113]: objective measures and subjective measures. The predominant evaluations of explanations have been subjective measures, since the notion of explanation is very human-centric; these range from qualitative displays of explanation examples, to crowd-sourced evaluations of human satisfaction with the explanations, as well as whether humans are able to understand the model. Nonetheless, it is also important to consider objective measures of explanation effectiveness, not only because these place explanations on a sounder theoretical foundation, but also because they allow us to *improve* our explanations by improving their objective measures.

One way to objectively evaluate explanations is to verify whether the explanation mechanism satisfies (or does not satisfy) certain axioms, or properties [104, 156]. In this paper, we focus on quantitative objective measures, and provide and analyze two such measures. First, we formalize the notion of fidelity of an explanation to the predictor function. One natural approach to measure fidelity, when we have apriori information that only a particular subset of features is relevant, is to test if the features with high explanation weights belong to this relevant subset [30]. In the absence of such apriori information, Ancona et al. [5] provide a more quantitative perspective on the earlier notion by measuring the correlation between the sum of a subset of feature importances and the difference in function value when setting the features in the subset to some reference value; by varying the subsets, we get different values of such subset correlations. In this work, we consider a simple generalization of this notion, that produces a single fidelity measure, which we call the infidelity measure.

Our infidelity measure is defined as the expected difference between the two terms: (a) the dot product of the input perturbation to the explanation and (b) the output perturbation (i.e., the

difference in function values after *significant perturbations* on the input). This general setup allows for a varied class of significant perturbations: a non-random perturbation towards a single reference or baseline value, perturbations towards multiple reference points e.g. by varying subsets of features to perturb, and a random perturbation towards a reference point with added small Gaussian noise, which allows the infidelity measure to be robust to small mis-specifications or noise in either the test input or the reference point.

We then show that the optimal explanation that minimizes this infidelity measure could be loosely cast as a novel combination of two well-known explanation mechanisms: Smooth-Grad [148] and Integrated Gradients [156] using a kernel function specified by the random perturbations. As another validation of our formalization, we show that many recently proposed explanations can be seen as optimal explanations for the infidelity measure with specific perturbations. We also introduce new perturbations which lead to novel explanations by optimizing the infidelity measure, and we validate the explanations are qualitatively better through human experiments. It is worth emphasizing that the infidelity measure, while objective, may not capture all the desiderata of a successful explanation; thus, it is still of interest to take a given explanation that does not have the form of the optimal explanation with respect to a specified infidelity measure and *modify it* to have lesser infidelity.

Analyzing this question leads us to another objective measure: the sensitivity of an explanation, which measures the degree to which the explanation is affected by insignificant perturbations from the test point. It is natural to wish for our explanation to have low sensitivity, since that would entail differing explanations with minor variations in the input (or prediction values), which might lead us to distrust the explanations. Explanations with high sensitivity could also be more amenable to adversarial attacks, as Ghorbani et al. [52] show in the context of gradient based explanations. Regardless, we largely expect explanations to be simple, and lower sensitivity could be viewed as one such notion of simplicity. Due in part to this, there have been some recent efforts to quantify the sensitivity of explanations [4, 52, 115]. We propose and analyze a simple robust variant of these recent proposals that is amenable to Monte Carlo sampling-based approximation. Our key contribution, however, is in relating the notion of sensitivity to our proposed notion of infidelity, which also allows us to address the earlier raised question of how to modify an explanation to have better fidelity. Asking this question for sensitivity might seem vacuous, since the optimal explanation that minimizes sensitivity (for all its related variants) is simply a trivial constant explanation, which is naturally not a desired explanation. So a more interesting question would be: how do we modify a given explanation so that it has lower sensitivity, but not too much. To quantify the latter, we could in turn use fidelity.

As one key contribution of the paper, we show that a restrained lowering of the sensitivity of an explanation also *increases* its fidelity. In particular, we consider a simple kernel smoothing based algorithm that appropriately lowers the sensitivity of any given explanation, but importantly also lowers its infidelity. Our meta-algorithm encompasses Smooth-Grad [148] which too modifies any existing explanation mechanism by averaging explanations in a small local neighborhood of the test point. In the appendix, we also consider an alternative approach to improve gradient explanation sensitivity and fidelity by adversarial training, which however requires that we be able to modify the given predictor function itself, which might not always be feasible. Our modifications improve both sensitivity and fidelity in most cases, and also provides explanations

that are qualitatively better, which we validate in a series of experiments.[1]

2.1 Infidelity: A measure of the Faithfulness of Feature Explanation to the Model

Consider the following general supervised learning setting: input space $\mathcal{X} \subseteq \mathbb{R}^d$, an output space $\mathcal{Y} \subseteq \mathbb{R}$, and a (machine-learnt) black-box predictor $\mathbf{f} : \mathbb{R}^d \mapsto \mathbb{R}$, which at some test input $\mathbf{x} \in \mathbb{R}^d$, predicts the output $\mathbf{f}(\mathbf{x})$. Then a feature attribution explanation is some function $\Phi : \mathcal{F} \times \mathbb{R}^d \mapsto \mathbb{R}^d$, that given a black-box predictor $\mathbf{f}$, and a test point $\mathbf{x}$, provides importance scores $\Phi(\mathbf{f}, \mathbf{x})$ for the set of input features. We let $\| \cdot \|$ denote a given norm over the input and explanation space. In experiments, if not specified, this will be set to the ℓ_2 norm.

2.1.1 Defining the infidelity measure

A natural notion of the goodness of an explanation is to quantify the degree to which it captures how the predictor function itself changes in response to significant perturbations. Along this spirit, [10, 142, 156] propose the completeness axiom for explanations consisting of feature importances, which states that the sum of the feature importances should sum up to the difference in the predictor function value at the given input and some specific baseline. [5] extend this to require that the sum of a subset of feature importance weights should sum up to to the difference in the predictor function value at the given input and to a perturbed input setting the subset of features to some specific baseline value. When the the subset of features is large, this would entail that explanations capture the combined importance of the subset of features even if not the individual feature importances, and when the the subset of features is small, this would entail that explanations capture the individual importance of features. We note that this can be contrasted with requiring the explanations to capture the function values itself as in the causal local explanation metric of [126], rather than the difference in function values, but we focus on the latter. Letting S_k denote a subset of k features, [5] then measured the discrepancy of the above desiderata as the correlation between $\sum_{i \in S_k} \Phi(\mathbf{f}, \mathbf{x})_i$ and $\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}[\mathbf{x}_{S_k} = 0])$, where $x[x_S = a]_j = a\,\mathbb{I}(j \in S) + x_j\,\mathbb{I}(j \notin S)$.

One minor caveat with the above, is that we may be interested in perturbations more general than setting feature values to 0, or even to a single baseline; for instance, we might simultaneously require smaller discrepancy over a set of subsets, or some distribution of subsets (as is common in game theoretic approaches to deriving feature importances [31, 104, 151]), or even simply a prior distribution over the baseline input. The correlation measure also focuses on second order moments, and is not as easy to optimize. We thus build on the above developments, by first allowing random perturbations on feature values instead of setting certain features to some single value, and secondly, by replacing correlation with expected mean square error (our development could be further generalized to allow for more general loss functions). We term our evaluation measure *explanation infidelity*.

Definition 1. Given a black-box function $\mathbf{f}$, explanation functional Φ, a random variable $\mathbf{I}$ taking values in $\mathbf{I}$ and with probability measure $\mu_{\mathbf{I}}$, which represents meaningful perturbation of interest,

[1] Implementation available at https://github.com/chihkuanyeh/saliency_evaluation.

we define the explanation infidelity of Φ as:

$$\text{INFD}(\Phi, \mathbf{f}, \mathbf{x}) = \mathbb{E}_{\mathbf{I} \sim \mu_{\mathbf{I}}}[\|\mathbf{I}^T \Phi(\mathbf{f}, \mathbf{x}) - (\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x} - \mathbf{I}))\|^2]. \tag{2.1}$$

$\mathbf{I}$ represents significant perturbations around $\mathbf{x}$, and can be specified in various ways. We begin by listing various plausible perturbations of interest.

- Difference to baseline: $\mathbf{I} = \mathbf{x} - \mathbf{x}_0$, the difference between input and baseline.
- Subset of difference to baseline: for any fixed subset $S_k \subseteq [d]$, $\mathbf{I}_{S_k} = \mathbf{x} - \mathbf{x}[\mathbf{x}_{S_k} = (\mathbf{x}_0)_{S_k}]$ corresponds to the perturbation in the correlation measure of [5] when $\mathbf{x}_0 = 0$.
- Difference to noisy baseline: $\mathbf{I} = \mathbf{x} - \mathbf{z}_0$, where $\mathbf{z}_0 = \mathbf{x}_0 + \epsilon$, for some zero mean random vector ϵ, for instance $\epsilon \sim \mathcal{N}(0, \sigma^2)$.
- Difference to multiple baselines: $\mathbf{I} = \mathbf{x} - \mathbf{x}_0$, where $\mathbf{x}_0$ is a random variable that can take multiple values.

As we will next show in Section 2.1.3, many recently proposed explanations could be viewed as optimizing the aforementioned infidelity for varying perturbations $\mathbf{I}$. Our proposed infidelity measurement can thus be seen as a unifying framework for these explanations, but moreover, as a way to design new explanations, and evaluate any existing explanations.

2.1.2 Explanations with least Infidelity

Given our notion of infidelity, a natural question is: what is the explanation that are optimal with respect to infidelity, that is, have the least infidelity possible. This naturally depends on the distribution of the perturbations $\mathbf{I}$, and its surprisingly simple form is detailed in the following proposition.

Proposition 1. *Suppose the perturbations* $\mathbf{I}$ *satisfy the condition that* $\left(\int \mathbf{I}\mathbf{I}^T d\mu_{\mathbf{I}}\right)^{-1}$ *be invertible. The optimal explanation* $\Phi^*(\mathbf{f}, \mathbf{x})$ *that minimizes infidelity for perturbations* $\mathbf{I}$ *can then be written as*

$$\Phi^\star(\mathbf{f}, \mathbf{x}) = \left(\int \mathbf{I}\mathbf{I}^T d\mu_{\mathbf{I}}\right)^{-1} \left(\int \mathbf{I}\mathbf{I}^T IG(\mathbf{f}, x, \mathbf{I}) d\mu_{\mathbf{I}}\right), \tag{2.2}$$

where $\text{IG}(\mathbf{f}, \mathbf{x}, \mathbf{I}) = \int_{t=0}^{1} \nabla \mathbf{f}(\mathbf{x} + (t-1)\mathbf{I})\, dt$ is the integrated gradient [156] of $\mathbf{f}(\cdot)$ between $(\mathbf{x} - \mathbf{I})$ and $\mathbf{x}$, but can be replaced by any functional that satisfies $\mathbf{I}^T \text{IG}(\mathbf{f}, \mathbf{x}, \mathbf{I}) = \mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x} - \mathbf{I})$. We note that the optimal solution in (2.2) can be seen as applying a smoothing operation reminiscent of SmoothGrad on Integrated Gradients (or any explanation that satisfies the completeness axiom) with a special kernel $\mathbf{I}\mathbf{I}^T$.

2.1.3 Many Recent Explanations Optimize Infidelity

As we show in the sequel, many recently proposed explanation methods can be shown to be optimal with respect to our infidelity measure in Definition 1, for varying perturbations $\mathbf{I}$.

Proposition 2. *Suppose the perturbation* $\mathbf{I} = \mathbf{x} - \mathbf{x}_0$ *is a fixed scalar equal to the difference between* $\mathbf{x}$ *and some baseline* $\mathbf{x}_0$*. Then for any explanation* $\Phi^*(\mathbf{f}, \mathbf{x})$ *that is optimal with respect to infidelity for perturbations* $\mathbf{I}$*, satisfying* $INFD(\Phi^*, \mathbf{f}, \mathbf{x}) = 0$*,* $\Phi^*(\mathbf{f}, \mathbf{x}) \odot \mathbf{I}$ *satisfies the completeness axiom; also satisfied by IG [156], DeepLift [142], LRP [10].*

Proposition 3. *Suppose the perturbation* $\mathbf{I}_\epsilon = \epsilon \cdot \mathbf{e}_i$ *where* $\mathbf{e}_i$ *is a coordinate basis vector, then the optimal explanation* $\Phi^*_\epsilon(\mathbf{f}, \mathbf{x})$ *with respect to infidelity for perturbations* $\mathbf{I}_\epsilon$*, satisfies:*

$\lim_{\epsilon \to 0} \Phi^*_\epsilon(\mathbf{f}, \mathbf{x}) = \nabla \mathbf{f}(\mathbf{x})$, *so that the limit point of the optimal explanations is the gradient explanation [141].*

Proposition 4. *Suppose the perturbation* $\mathbf{I} = \mathbf{e}_i \odot \mathbf{x}$, *where* $\mathbf{e}_i$ *is a coordinate basis vector. Then for the optimal explanation* $\Phi^*(\mathbf{f}, \mathbf{x})$ *with respect to infidelity for perturbations* $\mathbf{I}$, $\Phi^*(\mathbf{f}, \mathbf{x}) \odot \mathbf{x}$ *is the occlusion-1 explanation[187].*

Proposition 5. *Following the notation in [104], given a test input* $\mathbf{x}$, *suppose there is a mapping* $h_\mathbf{x} : \{0,1\}^d \mapsto \mathbb{R}^d$ *that maps simplified binary inputs* $\mathbf{z} \in \{0,1\}^d$ *to the real-valued inputs* $\mathbf{x} \in \mathbb{R}^d$, *and suppose that the given test input* $\mathbf{x} = h_\mathbf{x}(\mathbf{z}_0)$ *where* $\mathbf{z}_0$ *is a vector with all ones and* $h_\mathbf{x}(\mathbf{0}) = \mathbf{0}$ *where* $\mathbf{0}$ *is the zero vector. Now, suppose the perturbation* $\mathbf{I} = h_\mathbf{x}(\mathbf{Z})$ *where* $\mathbf{Z} \in \{0,1\}^d$ *is a binary random vector with distribution:* $\mathbb{P}(\mathbf{Z} = \mathbf{z}) \propto \frac{d-1}{\binom{d}{\|\mathbf{z}\|_1} \|\mathbf{z}\|_1 (d - \|\mathbf{z}\|_1)}$. *Then for the optimal explanation* $\Phi^*(\mathbf{f}, \mathbf{x})$ *with respect to infidelity for perturbations* $\mathbf{I}$, $\Phi^*(\mathbf{f}, \mathbf{x}) \odot \mathbf{x}$ *is the Shapley value[104].*

2.1.4 Some Novel Examples of Optimal Explanations

By varying the perturbations $\mathbf{I}$ in our infidelity definition 1, we can not only recover existing explanations (as those that optimize the corresponding infidelity), as discussed in the previous section, but also design some novel explanations. We provide two such instances below.

Noisy Baseline. The completeness axiom is one of the most commonly adopted axioms in the context of explanations, but a caveat is that the baselines is set to some fixed vector, which does not account for noise in the input (or the baseline itself). We thus set the baseline to be a Gaussian random vector centered around a certain clean baseline (such as the mean input or zero) depending on the context. The explanation that optimizes infidelity with corresponding perturbations $\mathbf{I}$ is a novel explanation that can be seen as satisfying a robust variant of the completeness axiom.

Square Removal. Our second example is specific for image data. We argue that perturbations that remove random subsets of pixels in images may be somewhat meaningless, since there is very little loss of information given surrounding pixels that are not removed. Also ranging over all possible subsets to remove (as in SHAP [104]) is infeasible for high dimension images. We thus propose a modified subset distribution from that described in Proposition 5 where the perturbation $\mathbf{Z}$ has a uniform distribution over square patches with predefined length, which is in spirit similar to the work of [193]. This not only improves the computational complexity, but also better captures spatial relationships in the images. One can also replace the square with more complex random masks designed specifically for the image domain [124].

2.1.5 Local and Global Explanations

As discussed in [5], we can contrast between local and global feature attribution explanations: global feature attribution methods directly provide the change in the function value given changes in the features, whereas local feature attribution methods focus on the sensitivity of the function to the changes to the features, so that the local feature attributions need to be multiplied with the input to obtain an estimate of the change in the function value. Thus, for gradient-based explanations considered in [5], the raw explanation itself such as the gradient is a local explanation, while the raw explanation multiplied with the raw input is called a global explanation. In our

context, explanations optimizing Definition 2.1.1 are naturally local explanations as $\mathbf{I}$ is real-valued. However, this can be easily modified to a global explanation by multiplying with $\mathbf{x} - \mathbf{x}_0$ when $\mathbf{I}$ is a subset of $\mathbf{x} - \mathbf{x}_0$. The reason we emphasize this distinction is that since global and local explanations capture subtly different aspects, they should be compared separately. We note that our definition of local and global explanations follows the description of [5], distinct from that in [126].

2.2 Objective Measure: Explanation Sensitivity

A classical approach to measure the sensitivity of a function, would simply be the gradient of the function with respect to the input. Therefore, the sensitivity of the explanation can be defined as: for any $j \in \{1, \ldots, d\}$,

$$[\nabla_{\mathbf{x}} \Phi(\mathbf{f}(\mathbf{x}))]_j = \lim_{\epsilon \to 0} \frac{\Phi(\mathbf{f}(\mathbf{x} + \epsilon\, \mathbf{e}_j)) - \Phi(\mathbf{f}(\mathbf{x}))}{\epsilon},$$

where $\mathbf{e}_j \in \mathbb{R}^d$ is the j-th coordinate basis vector, with j-th entry one and all others zero. It quantifies how the explanation changes as the input is varied infinitesimally. And as a scalar-valued summary of this sensitivity vector, a natural approach is to simply compute some norm of the sensitivity vector: $\|\nabla_{\mathbf{x}} \Phi(\mathbf{f}(\mathbf{x}))\|$. A slightly more robust variant would be a locally uniform bound:

$$\text{SENS}_{\text{GRAD}}(\Phi, \mathbf{f}, \mathbf{x}, r) = \sup_{\|\delta\| \leq r} \|\nabla_{\mathbf{x}} \Phi(\mathbf{x} + \delta)\|. \tag{2.3}$$

This is in turn related to local Lipschitz continuity [4] around $\mathbf{x}$:

$$\text{SENS}_{\text{LIPS}}(\Phi, \mathbf{f}, \mathbf{x}, r) = \sup_{\|\delta\| \leq r} \frac{\|\Phi(\mathbf{x}) - \Phi(\mathbf{x} + \delta))\|}{\|\delta\|}, \tag{2.4}$$

Thus if an explanation has locally uniformly bounded gradients, it is locally Lipshitz continuous as well. In this paper, we consider a closely related measure, we term *max-sensitivity*, that measures the maximum change in the explanation with a small perturbation of the input $\mathbf{x}$.

Definition 2. Given a black-box function $\mathbf{f}$, explanation functional Φ, and a given input neighborhood radius r, we define the max-sensitivity for explanation as:

$$\text{SENS}_{\text{MAX}}(\Phi, \mathbf{f}, \mathbf{x}, r) = \max_{\|\mathbf{y} - \mathbf{x}\| \leq r} \|\Phi(\mathbf{f}, \mathbf{y}) - \Phi(\mathbf{f}, \mathbf{x}))\|.$$

It can be readily seen that if an explanation is locally Lipshitz continuous, it has bounded max-sensitivity as well:

$$\text{SENS}_{\text{MAX}}(\Phi, \mathbf{f}, \mathbf{x}, r) := \max_{\|\delta\| \leq r} \|\Phi(\mathbf{f}, \mathbf{x} + \delta) - \Phi(\mathbf{f}, \mathbf{x}))\| \leq \text{SENS}_{\text{LIPS}}(\Phi, \mathbf{f}, \mathbf{x}, r)\, r, \tag{2.5}$$

The main attraction of the max-sensitivity measure is that it can be robustly estimated via Monte-Carlo sampling, as in our experiments. Can we then modify a given explanation so that it has lower sensitivity? If so, how much do we lower its sensitivity? There are two key objections to the very premise of these questions on how to lower sensitivity of an explanation. For the first objection, as we noted in the introduction, sensitivity provides only a partial measure of what is desired from an explanation. This can be seen from the fact that the optimal explanation that minimizes the above max-sensitivity measure is simply a constant explanation that just outputs a (potentially nonsensical) constant value for all possible test inputs. The second objection is that

natural explanations might have a certain amount of sensitivity by their very nature, either because the model is sensitive, or because the explanations themselves are constructed by measuring the sensitivities of the predictor function, so that their sensitivities in turn is likely to be more than that of the function. In which case, we might not want to lower their sensitivities, since it might affect the fidelity of the explanation to the predictor function, and perhaps degrade the explanation towards the vacuous constant explanation.

As one key contribution of the paper, we show that it is indeed possible to reduce sensitivity "responsibly" by ensuring that it also *lowers the infidelity*, as we detail in the next section. We start by relating the sensitivity of an explanation to its infidelity, and then show that appropriately reducing the sensitivity can achieve two ends: lowering sensitivity of course, but surprisingly, also lowering the infidelity itself.

2.3 Reducing Sensitivity and Infidelity by Smoothing Explanations

We then introduce a robust version of explanation fidelity which measures the maximum infidelity while adding a small perturbation to $\mathbf{x}$:

Definition 3. Given a black-box function INFD, explanation functional Φ, a random variable $\mathbf{I}$ which represents meaningful perturbation of interest, we define the robust fidelity of $\mathbf{x}$ as:

$$\begin{aligned}&\text{RINFD}(\Phi, \text{INFD}, \mathbf{x}) = \max_{\|\mathbf{u}\| \leq r} \text{INFD}(\Phi, \text{INFD}, \mathbf{x}+\mathbf{u}) \\ &= \max_{\|\mathbf{u}\| \leq r} \mathbb{E}_{\mathbf{I}}[\|\mathbf{I}^T \Phi(\text{INFD}, \mathbf{x}+\mathbf{u}) - (\text{INFD}(\mathbf{x}+\mathbf{u}) - \text{INFD}(\mathbf{x}+\mathbf{u}-\mathbf{I}))\|^2].\end{aligned}$$

We note that the optimal explanation that optimizes the robust infidelity equals to the optimal explanation that optimizes the infidelity. Therefore, by introducing the robust infidelity, we do not modify the optimal explanation but only capture a more robust measurement of the infidelity score. We introduce the following theorem that gives a lower bound for the robust infidelity that relates to the explanation sensitivity. The intuition is that by Definition 3, $\mathbf{I}^T \Phi(\text{INFD}, \mathbf{x}+\mathbf{u})$ and $\text{INFD}(\mathbf{x}+\mathbf{u}) - \text{INFD}(\mathbf{x}+\mathbf{u}-\mathbf{I})$ should be close for all $\mathbf{u}$. However, if $\mathbf{I}^T \Phi(\text{INFD}, \mathbf{x}+\mathbf{u})$ and $\text{INFD}(\mathbf{x}+\mathbf{u}) - \text{INFD}(\mathbf{x}+\mathbf{u}-\mathbf{I})$ have a very different sensitivity when perturbing $\mathbf{u}$, the difference will naturally be large for some $\mathbf{u}$.

Theorem 1. *Given a black-box function INFD, explanation functional Φ, a random variable $\mathbf{I}$, and let $A(\mathbf{x}) = \max_{\|\mathbf{u}\| \leq r} \mathbb{E}_{\mathbf{I}}[\|\mathbf{I}^T \Phi(INFD, \mathbf{x}+\mathbf{u}) - \mathbf{I}^T \Phi(INFD, \mathbf{x})\|]$, and $B(\mathbf{x}) = \max_{\|\mathbf{u}\| \leq r} \mathbb{E}_{\mathbf{I}}[\|INFD(\mathbf{x}+\mathbf{u}) - INFD(\mathbf{x})\|]$.*

$$RINFD(\mathbf{x}) \geq \left(\frac{A(\mathbf{x}) - B(\mathbf{x}) - B(\mathbf{x}-\mathbf{I})}{2}\right)^2.$$

We note $A(\mathbf{x})$ can be approximated by $\text{SENS}_{\text{MAX}}(\Phi, \text{INFD}, \mathbf{x}, r)$, which shows that $A(\mathbf{x})$ is related to the sensitivity of explanation Φ and $B(\mathbf{x})$ can be seen as the sensitivity of function INFD. When the explanation is much more sensitive than the model, the robust infidelity is lower bounded by the difference between explanation sensitivity and model sensitivity, which is clearly

undesirable. This suggests that we may improve the explanation sensitivity along with explanation infidelity.

We have shown that if the explanation sensitivity is much larger than the function sensitivity around some input $\mathbf{x}$, the infidelity measure in turn will necessarily be large for some point around $\mathbf{x}$ (that is, loosely, infidelity is lower bounded by the difference in sensitivity of the explanation and the function). Given that a large class of explanations are based on sensitivity of the function at the test input, and such sensitivities in turn can be more sensitive to the input than the function itself, does that mean that sensitivity-based explanations are just fated to have a large infidelity? In the following, we show that this need not be the case: by appropriately lowering the sensitivity of any given explanation, we not only reduce its sensitivity, but also its infidelity.

Given any kernel $k(\mathbf{x}, \mathbf{z})$ over the input domain with respect to which we desire smoothness, and some explanation functional $\Phi(\mathbf{f}, \mathbf{z})$, we can define a smoothed explanation as

$\Phi_k(\mathbf{f}, \mathbf{x}) := \int_{\mathbf{z}} \Phi(\mathbf{f}, \mathbf{z})\, k(\mathbf{x}, \mathbf{z}) d\mathbf{z}$. When $k(\mathbf{x}, \mathbf{z})$ is set to the Gaussian kernel, $\Phi_k(\mathbf{f}, \mathbf{x})$ becomes Smooth-Grad [148]. We now show that the smoothed explanation is less sensitive than the original sensitivity averaged around $\mathbf{x}$.

Theorem 2. *Given a black-box function* $\mathbf{f}$*, explanation functional* Φ*, the smoothed explanation functional* Φ_k*,*

$$\text{SENS}_{\text{MAX}}(\Phi_k, \mathbf{f}, \mathbf{x}, r) \leq \int_{\mathbf{z}} \text{SENS}_{\text{MAX}}(\Phi, \mathbf{f}, \mathbf{z}, r) k(\mathbf{x}, \mathbf{z}) d\mathbf{z}.$$

Thus, when the sensitivity SENS_{MAX} is large only along some directions $\mathbf{z}$, the averaged sensitivity could be much smaller than the worst case sensitivity over directions $\mathbf{z}$.

We now show that under certain assumptions, the infidelity of the smoothed explanation *actually decreases*. First, we introduce two relevant terms:

$$C_1 = \max_{\mathbf{x}} \frac{\int_{\mathbf{I}} \int_{\mathbf{z}} \|\mathbf{f}(\mathbf{z}) - \mathbf{f}(\mathbf{z} - \mathbf{I}) - [\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x} - \mathbf{I})]\|^2 k(\mathbf{x}, \mathbf{z}) d\mathbf{z} d\mu_{\mathbf{I}}}{\int_{\mathbf{I}} \int_{\mathbf{z}} \|\mathbf{I}^T \Phi(\mathbf{f}, \mathbf{z}) - [\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x} - \mathbf{I})]\|^2 k(\mathbf{x}, \mathbf{z}) d\mathbf{z} d\mu_{\mathbf{I}}}, \tag{2.6}$$

$$C_2 = \max_{\mathbf{x}} \frac{\int_{\mathbf{I}} \| \int_{\mathbf{z}} \{\mathbf{I}^T \Phi(\mathbf{f}, \mathbf{z}) - [\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x} - \mathbf{I})]\} k(\mathbf{x}, \mathbf{z}) d\mathbf{z}\|^2 d\mu_{\mathbf{I}}}{\int_{\mathbf{I}} \int_{\mathbf{z}} \|\mathbf{I}^T \Phi(\mathbf{f}, \mathbf{z}) - [\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x} - \mathbf{I})]\|^2 k(\mathbf{x}, \mathbf{z}) d\mathbf{z} d\mu_{\mathbf{I}}}. \tag{2.7}$$

We note that when the sensitivity of $\mathbf{f}$ is much smaller than the sensitivity of $\mathbf{I}^T \Phi(\mathbf{f}, \cdot)$, the numerator of the term C_1 will be much smaller than the denominator of C_1, so that the term C_1 will be small. The term C_2 is smaller than one by Jensen's inequality, but in practice it may be much smaller than one when $\mathbf{I}^T \Phi(\mathbf{f}, \mathbf{z}) - [\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x} - \mathbf{I})]$ have different signs for varying $\mathbf{z}$. We now present our theorem which relates the infidelity of the smoothed explanation to that of the original explanation.

Theorem 3. *Given a black-box function* $\mathbf{f}$*, explanation functional* Φ*, the smoothed explanation functional* Φ_k*, some perturbation of interest* $\mathbf{I}$*,* C_1, C_2 *defined in* (2.6) *and* (2.7) *where* $C_1 \leq \frac{1}{\sqrt{2}}$*,*

$$\mathit{INFD}(\Phi_k, \mathbf{f}, \mathbf{x}) \leq \frac{C_2}{1 - 2\sqrt{C_1}} \int_{\mathbf{z}} \mathit{INFD}(\Phi, \mathbf{f}, \mathbf{z}) k(\mathbf{x}, \mathbf{z}) d\mathbf{z}.$$

When $\frac{C_2}{1-2\sqrt{C_1}} \leq 1$, we show that the infidelity of Φ_k is less than the infidelity of Φ, as $\int_{\mathbf{z}} \text{INFD}(\Phi, \mathbf{f}, \mathbf{z}) k(\mathbf{x}, \mathbf{z}) d\mathbf{z}$ is usually very close to $\text{INFD}(\Phi, \mathbf{f}, \mathbf{z})$. This shows that smoothed explanation can be less sensitive and more faithful, which is validated in the experiments.

Datasets	MNIST	
Methods	SENS_{MAX}	INFD
Grad	0.86	4.12
Grad-SG	0.23	1.84
IG	0.77	2.75
IG-SG	0.22	1.52
GBP	0.85	4.13
GBP-SG	0.23	1.84
Noisy Baseline	0.35	0.51

(a) Results for local explanations on MNIST dataset.

Datasets	MNIST		Cifar-10		Imagenet	
Methods	SENS_{MAX}	INFD	SENS_{MAX}	INFD	SENS_{MAX}	INFD
Grad	0.56	2.38	1.15	15.99	1.16	0.25
Grad-SG	0.28	1.89	1.15	13.94	0.59	0.24
IG	0.47	1.88	1.08	16.03	0.93	0.24
IG-SG	0.26	1.72	0.90	15.90	0.48	0.23
GBP	0.58	2.38	1.18	15.99	1.09	0.15
GBP-SG	0.29	1.88	1.15	13.93	0.41	0.15
SHAP	0.35	1.20	0.93	5.78	–	–
Square	0.24	0.46	0.99	2.27	1.33	0.04

(b) Results for global explanations on MNIST, Cifar-10 and imagenet.

Table 2.1: Sensitivity and Infidelity for local and global explanations.

2.4 Experiments

Setup. We perform our experiments on randomly selected images from MNIST, CIFAR-10, and ImageNet. In our comparisons, we restrict local variants of the explanations to MNIST, since sensitivity of function values given pixel perturbations make more sense for grayscale rather than color images. To calculate our infidelity measure, we use the noisy baseline perturbation for local variants of the explanations, and the square removal for global variants of the explanations, and use Monte Carlo Sampling to estimate the measures. We use Grad, IG, GBP, and SHAP to denote vanilla gradient [142], integrated gradient [156], Guided Back-Propagation [150], and KernelSHAP [104] respectively, and add the postfix "-SG" when Smooth-Grad [148] is applied. We call the optimal explanation with respect to the perturbation Noisy Baseline and Square Removal as NB and Square for simplicity.

Explanation Sensitivity and Infidelity. We show results comparing sensitivity and infidelity for local explanations on MNIST and global explanations on MNIST, CIFAR-10, and ImageNet in Table 2.1. Recalling the discussion from Section 2.1.5, global explanations include a point-wise multiplication with the image minus baseline, but local explanations do not. We observe that the noisy baseline and square removal optimal explanations achieve the lowest infidelity, which is as expected, since they explicitly optimize the corresponding infidelity. We also observe that Smooth-Grad improves *both* sensitivity and infidelity for all base explanations across all datasets, which corroborates the analysis in section 2.3, and also addresses plausible criticisms of lowering sensitivity via smoothing: while one might expect such smoothing to increase infidelity, modest smoothing actually improves infidelity. We also perform a sanity check experiment when the perturbation follows that in SHAP (Defined in Prop.2.5), and we verify that SHAP has the lowest infidelity for this perturbation.

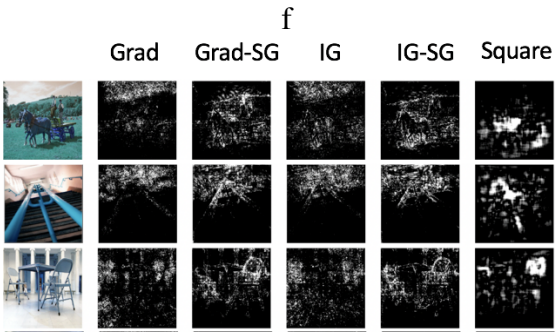


Figure 2.1: Examples of explanations on Imagenet.

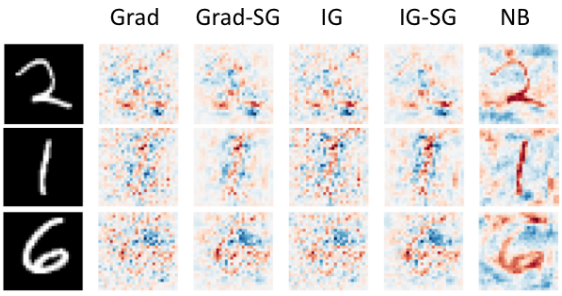


Figure 2.2: Examples of local explanations on MNIST.

Visualization. For a qualitative evaluation, we show several examples of global explanations on ImageNet, and local explanations on MNIST. The explanations optimizing our infidelity measure with respect to Square and Noisy Baseline (NB) perturbations, show a cleaner saliency map, highlighting the actual object being classified, when compared to then other explanations. For example, Square is the only explanation that highlights the whole bannister in the second image of Figure 2.1. For local examples on MNIST, NB clearly shows the digits, as well as regions that would increase the prediction score if brightened, such as the region on top of the number 6, which gives more insight into the behavior of the model. We also observe that SG provides a cleaner set of explanations, which validates the experimental results in [148], as well as our analysis in Section 2.3. We provide a more complete set of visualization results with higher resolution in the appendix.

Human Evaluation. We perform a controlled experiment to validate whether the infidelity measure aligns with human intuitions in a setting where we have an approximated ground truth feature for our model, following the setting of [88]. We create a dataset of two classes (bird and frog), with the image of the bird or frog in one half of the overall image, and just the caption in the other half (as shown in Figure 2.4). The images are potentially noisy with noise probability $p \in \{0, 0.6\}$: when $p = 0$, the image always agrees with the caption, and when $p = 0.6$, we randomize the image 60 percent of the time to a random image of another class. We train two models which both achieve testing accuracy above 0.95, where one model only relies on the image and the other only relies on the caption[2]. We then show the original input with aligned image and text, the prediction result, along with the corresponding explanations of the model (among Grad, IG, Grad-SG, and OPT) to humans, and test how often humans are able to infer the approximated ground truth feature (image or caption) the model relies on. The optimal explanation (OPT) is the explanation that minimizes our infidelity measure with respect to perturbation $\mathbf{I}$ defined as the right half or the left half of the image (since the location of the caption is in one half of the overall image in our case; but note that in more general settings, we could simply use a caption bounding box detector to specify our perturbations). Our human study includes 2 models, 4 explanations, and 16 test users, where each test user did a series of 8 tasks (2 models $\times$ 4 explanations) on

[2]When $p = 0$, the trained model solely relies on the image (accuracy for image only input is 0.9, but accuracy for caption only input is 0.5). When $p = 0.6$, the trained model only relies on the caption (the accuracy for caption only input is 0.98 but the accuracy for image only input is 0.5

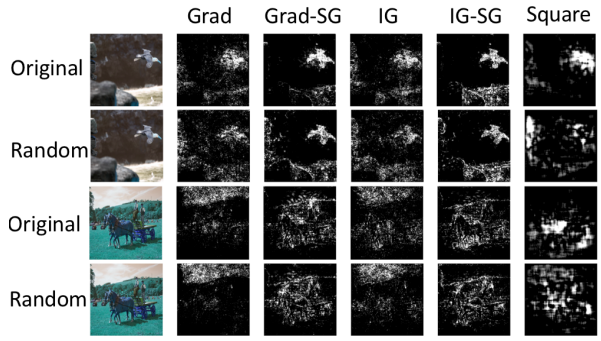


Figure 2.3: Examples of various explanations for the original model and the randomized model.

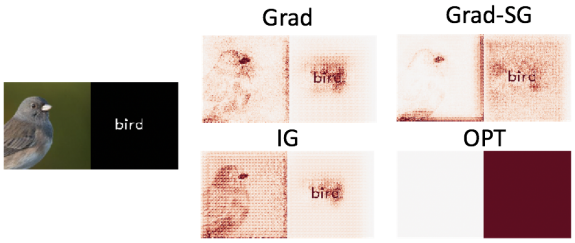


Figure 2.4: One example of explanations where the approximated ground truth is the right block (model focuses on the text). Some explanations focus on both text and image, so that just from these explanations, might be difficult to infer the ground truth feature used.

	Grad	Grad-SG	IG	IG-SG	Square
Corr	0.17	0.10	0.18	0.16	0.13
Corr (abs)	0.57	0.62	0.61	0.62	0.28

Table 2.2: Correlation of the explanation between the original model randomized model for the sanity check.

	Grad	Grad-SG	IG	OPT
Infid.	0.55	0.38	0.35	0.00
Acc.	0.47	0.50	0.53	0.88

Table 2.3: The infidelity and the accuracy human are able to predict the input blocked used based on the explanations.

random images. We report the average human accuracy and the infidelity measure for each explanation models in Table 2.3. We observe that unsurprisingly OPT has the best infidelity score by construction, and we also observe that the infidelity aligns with human evaluation result in general. This suggests that a faithful explanation communicates the important feature better in this setting, which validates the usefulness of the objective measure.

Sanity Check. Recent work in the interpretable machine learning literature [2, 38] has strongly argued for the importance of performing sanity checks on whether the explanation is at least loosely related to the model. Here, we conduct the sanity check proposed by Adebayo et al. [2], to check if explanations look different when the network being explained is randomly perturbed. One might expect that explanations that minimize infidelity will naturally be faithful to the model, and consequently pass this sanity check. We show visualizations for various explanations (with and without absolute values) of predictions by a pretrained Resnet-50 model and a randomized Resnet-50 model where the final fully connected layer is randomized in Figure 2.3. We also report the average rank correlation of the explanations for the original model and the randomized model in Table 2.2. All explanations without the absolute value pass the sanity check, but the rank correlation for explanations with the absolute value between the original model and the randomized model is high. In this case, Square has the lowest rank correlation and the visualizations for two models look the most distinct, which supports the hypothesis that an explanation with low infidelity is also faithful to the model.

Connection To Later Work The proposed infidelity is a general metric that connects the design of many feature attribution methods by the core design idea of feature attribution: that the explanation should be faithful to the model. For instance, one of our key design of a theoretic-motivated feature interaction Faith-Shap in later chapters is that the feature interaction should be faithful to the model for some non-degenerate perturbation function.

Chapter 3

Threading the needle for off-manifold and on-manifold value functions for Shapley Value

Shapley values [140] were originally proposed to measure the contributions of players in the context of cooperative games. They have since gained significant traction in the area of *explainable machine learning*, where they are used to measuring contributions of different features [31, 63, 102, 104, 119, 120], data points [50, 79], neurons [51], and concepts [182], towards the output of a learned model.

Despite their popularity, using Shapley values in an explainable machine learning context is not straightforward. In a cooperative game setting, the game is characterized by a set value function that takes as input a subset (also called a coalition) of players, and outputs the *value* or utility of this subset. Given this set value function, Shapley values then provide real-valued player-wise contributions. In a machine learning context, however, there is no well-defined set value function that takes as input a subset of features and outputs the value of this subset. Instead, we have the model p, data distribution p, and the test data point $\mathbf{x}$. A critical ingredient in order to derive Shapley values is to define a "value function" $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(S)$ which takes a set of features S as input, along with the model p, the test data point $\mathbf{x}$, and data distribution p. The value function describes how $\mathsf{p}(\mathbf{x})$ would change if only a set of features in $\mathbf{x}$ participates in an cooperative setting.

However, there is no consensus on how to specify the value function. Existing value functions can be roughly classified into two classes: so-called *off-manifold* value functions measure the utility of coalitions of features by perturbing features outside the coalition (and hence taking the inputs outside the data manifold), while so-called *on-manifold* value functions measure the utility of coalitions of features by marginalizing out features outside the coalition while respecting the data distribution/manifold. Both approaches come with their respective caveats [95]. Off-manifold value functions by construction evaluate the model on data that is not from the training data distribution; and hence are sensitive to how the model handles off-manifold test data [76, 147], and where machine learning models are typically unstable [46]. Moreover, the resulting explanations are very susceptible to manipulation: one can manipulate the model function only on off-manifold regions while preserving its behavior on-manifold, just to get the

desired explanation (e.g. to show that the model is unbiased, while maintaining biased behavior on-manifold) [147]. In other words, the off-manifold value functions respect the model function, but not the data distribution. This is problematic since these value functions evaluate the model outside the data manifold, where most ML training methods do not come with any guarantees on model performance.

On-manifold value functions on the other hand are in general computationally expensive, particularly if they involve conditional expectations due to marginalizing over out-of-coalition features while conditioning on the coalition features. Approximating these conditional expectations via empirical distributions could result in idiosyncratic issues, such as all features having the same attribution [155]. Aas et al. [1], Lundberg and Lee [104], Štrumbelj and Kononenko [151] compute the conditional value function via distributional assumptions, such as independence and Gaussian mixtures, which may not be flexible enough for complex data. Lundberg et al. [105] attempt to calculate the conditional expectation value function for tree ensembles, but which Sundararajan and Najmi [155] criticize as having unclear assumptions on the features. Frye et al. [46] propose approximating these conditional expectations via a supervised and an unsupervised approach. As we show however, the supervised approach converges to the empirical conditional value function, while the unsupervised approach generates samples whose coalition features need not be set to the conditioned values. Another principled approach of calculating the conditional expectation based value function is by using importance sampling on the joint distribution, but this is computationally expensive. Further, we show that conditional expectation based value functions, even when calculated accurately, will also be prone to adversarial manipulation off the data manifold. Another caveat with on-manifold value functions is that they result in Shapley values that violates natural axioms [155]. Moreover, they might respect the data distribution at the price of fidelity to the model function: a feature with no intervention effect can still receive non-zero feature importance, which was seen as a positive aspect by Adler et al. [3], but a negative aspect by Merrick and Taly [112], Sundararajan and Najmi [155].

So both on-manifold and off-manifold value functions, and their resulting Shapley value explanations, have their competing pros and cons, and Chen et al. [24] among others have suggested to "pick one's poison" when faced with a specific application. Kumar et al. [95] have even suggested that these two competing approaches present a serious concern with using Shapley values itself. But what if we can design value functions that respect both the model and the data manifold, and are not susceptible to adversarial manipulations in low density data regions? Surprisingly, we show that we indeed can. Towards this, we first propose a set of axioms that aim to formalize the desiderata in the question above, and show that there exists a unique value function that satisfies these axioms, which we call the Joint Baseline value function, satisfies these axioms. While the Shapley value by Joint Baseline value function does not satisfy the original set of axioms proposed in Sundararajan and Najmi [155], it does satisfy a new set of axioms which takes both data density and function model into account. Moreover, we show that this unique value function is robust to low density adversarial manipulations. We also show that the resulting Shapley value, Joint Baseline Shapley (JBshap), can be computed efficiently and scaled to high dimensional data such as images.

3.1 Different Value Functions for Shapley Value Explanations

3.1.1 Problem Definition

Given a machine learning model $\mathsf{p}(\cdot) : \mathsf{R}^d \to \mathsf{R}$ trained on d-dimensional data, data distribution p, and an *explicand* (i.e. test input) $\mathbf{x} \in \mathbb{R}^d$, we aim to attribute the model output $f(\mathbf{x})$ to the individual features. We use $\Phi_{\mathsf{p},p,i}(\mathbf{x})$ to denote this *attribution* to the i-th feature. In certain cases, there is also a given baseline value $\mathbf{x}' \in \mathbb{R}^d$, which is often set to 0 or the data mean. When applying Shapley values (which originate in cooperative game theory) to this attribution problem, a critical first step is to capture the key elements of $\mathsf{p}, p, \mathbf{x}$ into a set value function $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(S) : 2^d \to \mathbb{R}$, where $S \subseteq [d]$ is a subset of features. In this context, subsets of features can be thought of as subsets of players; in the sequel, we will often refer to the i-th feature as the i-th *player* to emphasize the connection to game theory at play with Shapley explanations. Given such a set value function $v(\cdot)$, Shapley value attributions to the i-th player can be computed as $\phi_i(v) = \sum_{S \subseteq [d]/i} \frac{s!(n-s-1)!}{n!}[v(S \cup i) - v(S)]$. In other words, the Shapley value of player i is the weighted average of its marginal contribution $v(S \cup i) - v(S)$ averaged over all possible subsets $S \subseteq [d]$ that do not include player i. The Shapley value has been shown to be the unique solution satisfying a set of reasonable axioms; see [140] for a further discussion and history. We will provide a more detailed discussion of such axioms in Sec. 3.3 and Sec. 3.4.

In the explainable AI context, using the set value function $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(\cdot)$, we can then compute the Shapley importance values $\Phi_{\mathsf{p},p,i}(\mathbf{x}) = \Phi_i(\mathbf{f}_{\mathbf{x},\mathsf{p},p}(\cdot))$, where Φ_i is the Shapley value attribution to the i-th player as detailed above. The critical question however is: how to specify the set value function $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(\cdot)$?

3.1.2 Notation for Existing On-Manifold and Off-Manifold Value Functions

Different choices of the set value function v would lead to different Shapley values $\phi_i(v)$, and indeed the proper selection of v has been of great interest within the explainable AI community. We first set up some notation. We let x_S denote the sub-vector of $\mathbf{x}$ corresponding to the feature set $S \subseteq [d]$, and $x_{\bar{S}}$ as the sub-vector corresponding to the complementary subset $\bar{S} \subseteq [d]$. Given a *baseline* $\mathbf{x}' \in \mathbb{R}^d$, we will be using the notation $f(x_S; x'_{\bar{S}})$ to refer to the function f with input sub-vectors x_S and $x'_{\bar{S}}$ corresponding to features in S and $\bar{S}$ respectively.

3.1.2.1 On-Manifold Value Functions

In this paper, we consider the prototypical on-manifold value function of Conditional Expectation, which in turn results in the Conditional Expectation Shapley value (*CES*) [104, 155]. We will be overloading notation and term the *value function* itself as CES.

CES: $\mathbf{f}^{\mathsf{C}}_{\mathbf{x},\mathsf{p},p}(S) = \mathsf{E}_{\mathbf{x}'_{\bar{S}} \sim p(\mathbf{x}'_{\bar{S}}|\mathbf{x}_S)}[\mathsf{p}(\mathbf{x}_S; \mathbf{x}'_{\bar{S}})] = \int_{\mathbf{x}'_{\bar{S}}} \mathsf{p}(\mathbf{x}_S; \mathbf{x}'_{\bar{S}}) p(\mathbf{x}'_{\bar{S}}|\mathbf{x}_S) d\mathbf{x}'_{\bar{S}}$. In other words, CES takes the expectation of $f(\mathbf{x}_S; \mathbf{x}'_{\bar{S}})$ over the conditional density of $\mathbf{x}'_{\bar{S}}$ given $\mathbf{x}_S$ (with respect to the joint density $p(\cdot)$).

3.1.2.2 Off-Manifold Value Functions

As discussed earlier, off-manifold value functions quantify the contribution of a coalition of features by perturbing or intervening on the off-coalition features. In this paper, we consider the prototypical off-manifold value functions of Baseline Shapley (*Bshap*) [104, 154]) and its randomized variant.

Bshap: $\mathbf{f}^{B}_{\mathbf{x},f,p}(S) := \mathsf{p}(\mathbf{x}_S; \mathbf{x}'_{\bar{S}})$. Bshap directly takes the value of f corresponding to $\mathbf{x}$, and ignores the data density model p.

RBshap: $\mathbf{f}^{RB}_{\mathbf{x},\mathsf{p},p}(S) = \mathsf{E}_{\mathbf{x}'_{\bar{S}}}[\mathsf{p}(\mathbf{x}_S; \mathbf{x}'_{\bar{S}})] = \int_{\mathbf{x}'_{\bar{S}}} \mathsf{p}(\mathbf{x}_S; \mathbf{x}'_{\bar{S}}) p_b(\mathbf{x}'_{\bar{S}}) d\mathbf{x}'_{\bar{S}}$, where p_b is the marginal density of $\mathbf{x}_{\bar{S}}$. RBshap is a natural extension of Bshap, where the baseline $\mathbf{x}'$ in Bshap is generalized to a distribution of baselines, which is represented by $p_b(\mathbf{x}')$.

3.2 Issues of Existing Value Functions

3.2.1 Off-Manifold Value function: Not "Respecting" the Data Manifold

The main difference between on-manifold value functions and off-manifold value functions is that off-manifold value functions do not take the data manifold (density) into account. The machine learning model p may not be trained to have meaningful prediction values when the input is $[\mathbf{x}_S; \mathbf{x}'_{\bar{S}}]$. For instance, for image inputs, $[\mathbf{x}_S; \mathbf{x}'_{\bar{S}}]$ would represent an image where certain random pixels are replaced by some baseline value (such as setting to black). [46] thus raise the concern that Shapley values calculated by off-manifold value functions could be over-reliant on function values in off-manifold regions, which is not where the model is trained on. Issues could arise when the model behavior is different on the data manifold and off the data manifold. For instance, if the image classifier tends to predict an image with many random black pixels as "dog", the Shapley explanation for an image of a dog may be strongly affected by how the model handles such artifacts. This problem is emphasized by Slack et al. [147] as they show that they can train a new model that only differs to the original model in off-manifold regions, and they are able to manipulate the Shapley explanations for off-manifold value functions. We further discuss this issue in Sec. 3.2.3.

3.2.2 On-Manifold Value function: Difficulty to Calculate Conditional Value

In contrast to the robustness challenges of the previous section, the main difficulty with conditional expectation (CES) based value functions is computational. Many variants have been proposed to simplify the computational burden of calculating the CES value, however, they may come with the cost of its new issues. We discuss some key variants and their respective issues.

3.2.2.1 CES-Empirical

One variant is called CES-Empirical, which uses the empirical probability to calculate the conditioned value. To formally define CES-Empricial, let $p^E(\mathbf{x}) = \frac{1}{m}\mathbb{1}[\mathbf{x} \text{ in dataset}]$ denote the empirical distribution, given m points in the dataset. In this case, $\mathbf{f}^{CE}_{\mathbf{x},\mathsf{p},p^E}(S) = \mathsf{E}_{\mathbf{x}'_{\bar{S}} \sim p^E(\mathbf{x}'_{\bar{S}}|\mathbf{x}_S)}[\mathsf{p}(\mathbf{x}_S; \mathbf{x}'_{\bar{S}})] = \frac{\sum_{i=1}^m \mathbb{1}[\mathbf{x}^i_S = \mathbf{x}_S]\mathsf{p}(\mathbf{x}^i)}{\sum_{i=1}^m \mathbb{1}[\mathbf{x}^i_S = \mathbf{x}_S]}$ will be the average prediction of the points in the dataset where its feature set S is equal to $\mathbf{x}_S$.

Although CES-Empirical can be easy to calculate, the caveats of CES-Empirical have been discussed in a line of recent work; we summarize the key ones below. The main criticism of CES-Empirical is that when an explicand has feature values that are unique in the dataset (which is likely to be the case with a continuous data distribution), each feature will get the same Shapley value importance even if the model is not symmetric in all features [155].

3.2.2.2 CES-Supervised

Another popular CES version is called CES-Supervised [46]. The method is motivated by the observation that the conditional expectation $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(S) = \mathsf{E}_{\mathbf{x}'_{\bar{S}} \sim p(\mathbf{x}'_{\bar{S}}|\mathbf{x}_S)}[\mathsf{p}(\mathbf{x}_S; \mathbf{x}'_{\bar{S}})]$ can be seen to minimize the MSE loss $\text{mse}(v(\mathbf{x}_S)) := \mathsf{E}_{\mathbf{x}'_{\bar{S}} \sim p(\mathbf{x}'_{\bar{S}}|\mathbf{x}_S)}[\mathsf{p}(\mathbf{x}_S; \mathbf{x}'_{\bar{S}}) - v(\mathbf{x}_S)]^2$. Thus, Frye et al. [46] propose to learn the value function by a surrogate model g by minimizing the MSE loss:

$$\mathbf{f}^{CS}_{\mathbf{x},\mathsf{p},p^E}(S) = \arg\min_g \mathsf{E}_{\mathbf{x} \sim p^E} \mathsf{E}_{S \sim \text{shapley}}[(\mathsf{p}(\mathbf{x}) - g(\mathbf{x}_S))^2],$$

and the conditioned value $\mathbf{f}^{CS}_{\mathbf{x},\mathsf{p},p^E}(S)$ can be obtained by the surrogate model g by $g(\mathbf{x}_S)$. Frye et al. [46] propose to use a neural network with masked inputs as the surrogate model.

In order to calculate CES-supervised, Frye et al. [46] follow the standard approach of empirical risk minimization, and optimize the empirical MSE loss:

$$\mathsf{E}_{\mathbf{x} \sim p^E(\mathbf{x})} \mathsf{E}_{S \sim \text{shapley}}[(\mathsf{p}(\mathbf{x}) - \mathbf{f}^{CS}_{\mathbf{x}_S,\mathsf{p},p})^2]. \tag{3.1}$$

One issue of CES-Supervised is that it requires retraining the model completely, where the sample space is all possible subsets for every input. The training of the model g is even more difficult than the training of the original model p.

Another issue is that the resulting optimal ERM CES-supervised estimator is exactly the CES-Empirical estimate, as we show in the following theorem.

Proposition 1. The global minimizer to (3.1), where $p^E(\mathbf{x}) = \frac{1}{m}\mathbb{1}[\mathbf{x} \text{ in dataset}]$ is exactly equal to $f^{CE}_{\mathbf{x},\mathsf{p},p^E}(S)$. Thus, $\mathbf{f}^{CS}_{\mathbf{x}_S,\mathsf{p},p} = f^{CE}_{\mathbf{x},\mathsf{p},p^E}(S)$ when the empirical distribution is used.

The proof follows from a simple analysis of the stationary conditions of the objectives above. As a consequence of this surprising theorem, all pitfalls of CES-Empirical still carry over to CES-supervised when the empirical distribution is used. Even although Frye et al. [46] use a masked encoder model to learn $\mathbf{f}^{CS}_{\mathbf{x}_S,\mathsf{p},p}$, when the encoder is sufficiently flexible (e.g. large neural models) this still converges to CES-Empirical, and not the true conditioned value. One evidence is that CES-supervised is invariant to the behavior of p off the data points in the dataset. However, the true conditioned expectation should be dependent on the behavior of p off the data manifold.

We note that when the learning of the surrogate model has strong regularization, CES-SUP may not necessarily converge to CES-Empirical, but the fact that CES-SUP is invariant to the behavior of p off the data points validates that it does not capture the real conditioned value. Another issue of CES-Supervised, which is demonstrated in the experiment section, is that the empirical MSE loss can still be low even if the training objective is altered, and thus leaving CES-supervised to be prone to model-based attack.

3.2.2.3 CES-Sample

Suppose we have access to the conditioned probability $p(\mathbf{x}'_{\bar{S}}|\mathbf{x}_S)$, we can use Monte-Carlo sampling to estimate $\mathbf{f}^{CSam}_{\mathbf{x},\mathsf{p},p}(S) = \int_{\mathbf{x}'_{\bar{S}}} \mathsf{p}(\mathbf{x}_S;\mathbf{x}'_{\bar{S}})p(\mathbf{x}'_{\bar{S}}|\mathbf{x}_S)d\mathbf{x}'_{\bar{S}}$. To obtain the conditioned probability, Frye et al. [46] proposed a Masked variational auto-encoder approach, which results in samples that does not respect the conditioned value. An alternative is to sample conditioned probability that satisfies the conditioned constraint is to use importance-sampling, by first sampling uniformly from all $\mathbf{x}'_{\bar{S}}$, and average the value of $\mathsf{p}(\mathbf{x}_S;\mathbf{x}'_{\bar{S}})p(\mathbf{x}'_{\bar{S}}|\mathbf{x}_S)$ for each sampled point. The main downside of CES-Sample is that it is computationally expensive to calculate $\mathbf{f}^{CSam}_{\mathbf{x},\mathsf{p},p^E}(S)$, each of which may take around 10 to 100 accesses to the model p; in contrast, the computation of most other value functions require single access to the model p. Thus, CES-sample may be $10\times$ to $100\times$ more expensive to calculate compared to other value functions.

3.2.3 Off-Manifold and On-Manifold Value function: Sensitivity to Perturbation In Low-Density Regions

For off-manifold value functions, it is shown that one can manipulate the Shapley value explanations by perturbing the functions on off-manifold regions [147]. The methodology of Slack et al. [147] is that given an original black-box model f which has explanation $\phi(f)$, one may perturb f to f' by only changing the behavior in low-density regions. In practice, we usually only check the test accuracy and test prediction, and the difference of f and f' may not be detected in several cases. However, the explanation $\phi(f)$ and $\phi(f')$ may differ drastically. This undermines the original explanation $\phi(f)$, as there are now ground truth for how the model f should behave off-manifold. While previous works [46] claim that on-manifold value function is robust to such manipulation, we show that one can also manipulate (the Shapley values corresponding to) on-manifold value functions by perturbing the functions on low density regions (while fixing the model behavior on high density regions).

We first formalize a definition of robustness to off-manifold manipulations, and discuss the sensitivity (robustness) of existing value functions to off-manifold manipulations.

Definition 1. Given any two models $\mathsf{p}_1(\mathbf{x})$, $\mathsf{p}_2(\mathbf{x})$ and any probability measure $p(\mathbf{x})$, if $\max_{\mathbf{x}} |\mathsf{p}_1(\mathbf{x}) - \mathsf{p}_2(\mathbf{x})|p(\mathbf{x}) \leq \epsilon$ always entails $|\mathbf{f}_{\mathbf{x},\mathsf{p}_1,p}(S) - \mathbf{f}_{\mathbf{x},\mathsf{p}_2,p}(S)| \leq T\epsilon$ for any S, we term the value function $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(S)$ as strong T-robust to off-manifold perturbations.

The premise $\max_{\mathbf{x}} |\mathsf{p}_1(\mathbf{x}) - \mathsf{p}_2(\mathbf{x})|p(\mathbf{x}) \leq \epsilon$ bounds the maximum perturbation on low density regions. For instance, if we require that $\max_{\mathbf{x}} |\mathsf{p}_1(\mathbf{x}) - \mathsf{p}_2(\mathbf{x})|p(\mathbf{x}) \leq 0.01$, then in regions with density less than 0.01, we can perturb the function values by at most 1.0. However, we show that both Bshap and CES are not strong T-robust.

Proposition 2. Bshap and CES are not strong T-robust to off-manifold perturbations for $|T| < \infty$.

The key reason that Bshap and CES does not satisfy strong T-robustness is that the value function $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(S)$ for Bshap and CES can be determined by the behavior of p on low density regions. To address this issue, we show that the strong- T-robustness can be obtained by taking the data density into account in the calculation of value functions.

We empirically evaluate the sensitivity of Bshap and CES when the function is perturbed in low-density regions in practice in Sec. 3.6.

3.3 Axioms for Value Functions & A New Value Function

In this section, we argue that it is possible to get the best of, and avoiding the pitfalls of, both worlds — of on-manifold and off-manifold value functions — by taking both the function and the data distribution into account.

We first propose a set of axioms on set value functions that takes these desiderata into account.

1. **Linearity (over functions and distributions) (Lin.)**: For any functions $\mathsf{p}, \mathsf{p}_1, \mathsf{p}_2, p, p_1, p_2$, $\alpha_1 \mathbf{f}_{\mathbf{x},\mathsf{p}_1,p}(S) + \alpha_2 \mathbf{f}_{\mathbf{x},\mathsf{p}_2,p}(S) = \mathbf{f}_{\mathbf{x},\alpha_1\mathsf{p}_1+\alpha_2\mathsf{p}_2,p}(S)$, for $\alpha_1, \alpha_2 \in \mathbb{R}$ and $\alpha_1 \mathbf{f}_{\mathbf{x},\mathsf{p},\alpha_1 p_1}(S) + \alpha_2 \mathbf{f}_{\mathbf{x},\mathsf{p},p_1}(S) = \mathbf{f}_{\mathbf{x},\mathsf{p},\alpha_1 p_1+\alpha_2 p_2}(S)$ for $\alpha_1, \alpha_2 \geq 0$.
2. **Symmetry (over functions and distributions) (Sym.)**: for all $i, j \not\subseteq S$, p, p, $\mathbf{x}$, $\mathbf{x}'$ all being symmetric in the i-th and j-th dimensions implies $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(i \cup S) = \mathbf{f}_{\mathbf{x},\mathsf{p},p}(j \cup S)$. That is, if both the function p and the probability p are symmetric with respect to two features, then so should the value function $\mathbf{f}$.
3. **Dummy Player (over functions and distributions) (Dum.)**: p, p being invariant in the i-th dimension implies $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(S) = \mathbf{f}_{\mathbf{x},\mathsf{p},p}(i \cup S)$. That is, a completely irrelevant feature in both the function value and data density should also be irrelevant in the value function.
4. **Null Player (over functions and distributions) (Null)**: if p', p' satisfies $\mathsf{p}(\mathbf{x}') = \mathsf{p}'(\mathbf{x}'), p(\mathbf{x}') = p'(\mathbf{x}')$, then $\mathbf{f}_{\mathsf{p},p,\mathbf{x}}(\emptyset) = \mathbf{f}_{\mathsf{p}',p',\mathbf{x}}(\emptyset)$. That is, the value function when no players are in the set function solely depends on the function value and data density of the baseline.
5. **Efficiency (over functions and distributions) (Eff.)**: $\mathbf{f}_{\mathsf{p},p,\mathbf{x}}[d] - \mathbf{f}_{\mathsf{p},p,\mathbf{x}}[\emptyset] = \mathsf{p}(x)p(\mathbf{x}) - \mathsf{p}(x')p(\mathbf{x}')$. The difference between the value function of all players and no players is equal to the total joint density. Note that this axiom is not needed in the uniqueness of JBshap.
6. **Set Relevance (over functions and distributions) (Set.)**: If $\mathsf{p}_1(\mathbf{x}_S, \bar{x}_{\bar{S}}) = \mathsf{p}_2(\mathbf{x}_S, \bar{x}_{\bar{S}})$ and $p_1(\mathbf{x}_S, \bar{x}_{\bar{S}}) = p_2(\mathbf{x}_S, \bar{x}_{\bar{S}})$ for any $\bar{x}$, $\mathbf{f}_{\mathsf{p}_1,p_1,\mathbf{x}}(S) = \mathbf{f}_{\mathsf{p}_2,p_2,\mathbf{x}}(S)$. The set relevance axiom states that the value function is determined by how p, p behaves when the feature set S is fixed to be equal to $\mathbf{x}_S$.
7. **Strong-T-Robustness (over functions and distributions) (Rob.)**: For any functions $\mathsf{p}_1, \mathsf{p}_2, p$, if $\max_{\mathbf{x}} |\mathsf{p}_1(\mathbf{x}) - \mathsf{p}_2(\mathbf{x})| p(\mathbf{x}) \leq \epsilon$, then $|\mathbf{f}_{\mathbf{x},\mathsf{p}_1,p}(S) - \mathbf{f}_{\mathbf{x},\mathsf{p}_2,p}(S)| \leq T\epsilon$ for some constant T. This axiom has been discussed in Sec. 3.2.3.

As discussed earlier, Shapley values originated in a cooperative game context, as the unique *attribution function* that assigns attributions to individual players, given a set value function that specifies utilities of sets/coalitions of players. Our set of axioms above on the other hand are for the set value function itself: on what properties such set-value functions should satisfy. Our linearity axiom states that for a linear combination of models should have a corresponding linear combination of set-value functions. Similarly a convex combination (i.e. mixture) of distributions should have a corresponding convex combination of value functions.

Note that Shapley values themselves satisfy a linearity axiom, that states that a linear combination of set-value functions should have a linear combination of corresponding Shapley values. By combining the linearity axioms for set-value functions we specify here, with the linearity axiom of Shapley value feature attributions, it can be seen that we get that a combined linearity axiom: that for linear (convex) combinations p (p), the Shapley attributions should in turn be linear (convex) combinations of the corresponding Shapley values. We formalize the complete set of transitive axioms in Sec. 3.4.

The set relevance axiom states that the value function is determined by the behavior of p and p when the feature set of S is set to the value of $\mathbf{x}$. The rationale is that the value function identifies the contribution of $\mathbf{x}_S$ in $\mathsf{p}(\mathbf{x})$ and $p(\mathbf{x})$, and thus should be only relevant to $\mathsf{p}(\bar{\mathbf{x}})$ and $p(\bar{\mathbf{x}})$ when $\bar{\mathbf{x}}_S = \mathbf{x}_S$. In other words, how $\mathsf{p}(\bar{\mathbf{x}}), p(\bar{\mathbf{x}})$ behaves when $\bar{\mathbf{x}}_S$ is not equal to $\mathbf{x}_S$ should not affect the value function $\mathbf{f}_{\mathsf{p},p,\mathbf{x}}(S)$.

We next show the following uniqueness result:

Theorem 4. $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(S)$ *satisfies (Lin.), (Sym.), (Dum.), (Null), (Eff.), (Set.), (Rob.) axioms if and only* $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(S) = \mathsf{p}(\mathbf{x}_S, \mathbf{x}'_{\bar{S}})p(\mathbf{x}_S, \mathbf{x}'_{\bar{S}})$.

This result specifies the unique value function that satisfies natural axioms which take both the data and model into account. We term this unique value function the *Joint Baseline Shapley value* (JBshap):

Definition 2. (JBshap):

$$\mathbf{f}^{J}_{\mathbf{x},\mathsf{p},p}(S) = \mathsf{p}(\mathbf{x}_S; \mathbf{x}'_{\bar{S}})p(\mathbf{x}_S; \mathbf{x}'_{\bar{S}}) \tag{3.2}$$

An interesting fact is that for the uniqueness result, the efficiency axiom is not needed. Suppose the model $f(\mathbf{x}_S; \mathbf{x}'_S)$ converges to the conditional distribution $p(y|\mathbf{x}_S, \mathbf{x}'_S)$, then from (3.2), it can be seen that $v^{J}_{\mathbf{x},f,p}(S) \to p(y|\mathbf{x}_S, \mathbf{x}'_S)p(\mathbf{x}_S; \mathbf{x}'_S) = p(y, \mathbf{x}_S, \mathbf{x}'_S)$. In other words, as the model itself converges to the Bayes optimal classifier, JBshap assigns a value to $\mathbf{x}_S$ that is consistent with the joint density of the observation y and $(\mathbf{x}_S, \mathbf{x}'_S)$. We highlight that JBshap resembles on-manifold value function since it respects the data manifold, but can also be considered as off-manifold since it takes an interventional approach similar to off-manifold value functions.

A natural extension of JBshap is to average over all possible baseline values $\mathbf{x}'$, and we obtain RJBshap:

Definition 3. (RJBshap):

$$\mathbf{f}^{RJ}_{\mathbf{x},\mathsf{p},p}(S) = \int_{\mathbf{x}'_{\bar{S}}} [\mathsf{p}(\mathbf{x}_S; \mathbf{x}'_{\bar{S}})p(\mathbf{x}_S; \mathbf{x}'_{\bar{S}})p_b(\mathbf{x}')]d\mathbf{x}'_{\bar{S}},$$

where $p_b(\cdot)$ is the prior probability for a set of baseline values. We point out that RJBshap still satisfies the axioms (Lin.), (Sym.), (Dum.), (Null), (Eff.), (Rob), with the slight modification of (Sym.), (Null), (Eff.) by replacing the baseline $\mathbf{x}'$ in JBshap by the random baseline $\mathbf{x}'$ with distribution $p_b(\mathbf{x}')$

It is easy to observe that JBshap is equal to Bshap when the function $\mathsf{p}(\mathbf{x})$ is replaced by $\mathsf{p}(\mathbf{x})p(\mathbf{x})$ since $\mathbf{f}^{J}_{\mathbf{x},\mathsf{p},p}(S) = \mathbf{f}^{B}_{\mathbf{x},\mathsf{p}\cdot p,p}(S)$. The clear difference is that JBshap takes account of the data density $p(\cdot)$ and thus will not be effected by off-manifold regions $\mathbf{x}$ since $p(\mathbf{x})$ will be small. We can also relate RJBshap and CES, by noting that CES value function can be written as $\mathbf{f}^{C}_{\mathbf{x},\mathsf{p},p}(S) = \mathsf{E}_{\mathbf{x}'_{\bar{S}} \sim p(\mathbf{x}'_{\bar{S}}|\mathbf{x}_S)}[\mathsf{p}(\mathbf{x}_S; \mathbf{x}'_{\bar{S}})] = \int_{\mathbf{x}'_{\bar{S}}} \mathsf{p}(\mathbf{x}_S; \mathbf{x}'_{\bar{S}})p(\mathbf{x}'_{\bar{S}}; \mathbf{x}_S)/p(\mathbf{x}_S)d\mathbf{x}'_{\bar{S}}$. When the baseline follows an uniform solution for all possible values (such that p_b is a constant), CES corresponds to RJBshap divided by $p(\mathbf{x}_S)/C_0$, where C_0 is the constant value of p_b. However, we point out that the calculation of $p(\mathbf{x}_S)$ requires marginalizing over $|\bar{S}|$ variables to obtain $p(\mathbf{x}_S)$, which is computationally difficult. Thus, RJBshap is much more computational tractable compared to CES. By not dividing over $p(\mathbf{x}_S)$, both JBshap and RJBshap are strong T-robust to off-manifold manipulations, in contrast to CES.

3.4 Translative Relation For Shapley value Axioms

In this section, we better motivate the axioms for the value functions in Sec. 3.3 by introducing its relation to the axioms of Shapley value. Recall the set of axioms we introduced in Sec. 3.3, we

rename their abbreviations here for simpler presentations. Lin. in main text is renamed as (L-DS), Sym. in main text is renamed as (S-DS), Dum. in main text is renamed as (D-DS), Null in main text is renamed as (N-DS), Eff. in main text is renamed as (E-DS), Set. in main text is renamed as (Set-DS), and the renaming makes the axioms easier to distinguish to other versions of the respective axioms.

1. **Linearity (over functions and distributions) (L-DS)**: For any functions $\mathsf{p}, \mathsf{p}_1, \mathsf{p}_2, p, p_1, p_2$, $\alpha_1 \mathbf{f}_{\mathbf{x},\mathsf{p}_1,p}(S) + \alpha_2 \mathbf{f}_{\mathbf{x},\mathsf{p}_2,p}(S) = \mathbf{f}_{\mathbf{x},\alpha_1\mathsf{p}_1+\alpha_2\mathsf{p}_2,p}(S)$, for $\alpha_1, \alpha_2 \in \mathbb{R}$ and $\alpha_1 \mathbf{f}_{\mathbf{x},\mathsf{p},\alpha_1 p_1}(S) + \alpha_2 \mathbf{f}_{\mathbf{x},\mathsf{p},p_1}(S) = \mathbf{f}_{\mathbf{x},\mathsf{p},\alpha_1 p_1+\alpha_2 p_2}(S)$ for $\alpha_1, \alpha_2 \geq 0; \alpha_1 + \alpha_2 = 1$.
2. **Symmetry (over functions and distributions) (S-DS)**: for all $i, j \not\subseteq S$, p, p, $\mathbf{x}$, $\mathbf{x}'$ all being symmetric in the i-th and j-th dimensions implies $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(i \cup S) = \mathbf{f}_{\mathbf{x},\mathsf{p},p}(j \cup S)$.
3. **Dummy Player (over functions and distributions) (D-DS)**: p, p being invariant in the i-th dimension implies $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(S) = \mathbf{f}_{\mathbf{x},\mathsf{p},p}(i \cup S)$.
4. **Null Player (over functions and distributions) (N-DS)**: if p', p' satisfies $\mathsf{p}(\mathbf{x}') = \mathsf{p}'(\mathbf{x}'), p(\mathbf{x}') = p'(\mathbf{x}')$, then $\mathbf{f}_{\mathsf{p},p,\mathbf{x}}(\emptyset) = \mathbf{f}_{\mathsf{p}',p',\mathbf{x}}(\emptyset)$.
5. **Efficiency (over functions and distributions) (A-DS)**: $\mathbf{f}_{\mathsf{p},p,\mathbf{x}}[d] - \mathbf{f}_{\mathsf{p},p,\mathbf{x}}[\emptyset] = \mathsf{p}(x)p(\mathbf{x}) - \mathsf{p}(x')p(\mathbf{x}')$.
6. **Set Relevance (over functions and distributions) (Set-DS)**: If $\mathsf{p}_1(\mathbf{x}_S, \bar{x}_{\bar{S}}) = \mathsf{p}_2(\mathbf{x}_S, \bar{x}_{\bar{S}})$ and $p_1(\mathbf{x}_S, \bar{x}_{\bar{S}}) = p_2(\mathbf{x}_S, \bar{x}_{\bar{S}})$ for any $\bar{x}$, $\mathbf{f}_{\mathsf{p}_1,p_1,\mathbf{x}}(S) = \mathbf{f}_{\mathsf{p}_2,p_2,\mathbf{x}}(S)$.
7. **Strong-T-Robustness (over functions and distributions) (Rob-DS)**: For any functions $\mathsf{p}_1, \mathsf{p}_2, p$, if $\max_{\mathbf{x}} |\mathsf{p}_1(\mathbf{x}) - \mathsf{p}_2(\mathbf{x})| p(\mathbf{x}) \leq \epsilon$, then $|\mathbf{f}_{\mathbf{x},\mathsf{p}_1,p}(S) - \mathbf{f}_{\mathbf{x},\mathsf{p}_2,p}(S)| \leq T\epsilon$.

We then introduce the following definitions for a set of axioms for a value function. Define the set $[d] := \{1, 2, \ldots, d\}$. We first start from axioms with only a single input function f:

1. **Linearity (over functions) (L-IS)**: $\mathbf{f}_{\mathbf{x},\mathsf{p}_1,p}(S) + \mathbf{f}_{\mathbf{x},\mathsf{p}_2,p}(S) = \mathbf{f}_{\mathbf{x},\mathsf{p}_1+\mathsf{p}_2,p}(S)$.
2. **Symmetry (over functions) (S-IS)**: for all $i, j \not\subseteq S$, p, $\mathbf{x}$, all being symmetric in the i-th and j-th dimensions implies $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(i \cup S) = \mathbf{f}_{\mathbf{x},\mathsf{p},p}(j \cup S)$.
3. **Dummy Player (over functions) (D-IS)**: p being invariant in the i-th dimension implies $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(S) = \mathbf{f}_{\mathbf{x},\mathsf{p},p}(i \cup S)$.
4. **Null Player (over functions) (N-IS)**: if $\mathsf{p}(\mathbf{x}) = \mathsf{p}'(\mathbf{x})$, then $\mathbf{f}_{\mathsf{p},p,\mathbf{x}}(\emptyset) = \mathbf{f}_{\mathsf{p}',p,\mathbf{x}}(\emptyset)$.
5. **Efficiency (over functions) (E-IS)**: $\mathbf{f}_{\mathsf{p},p,\mathbf{x}}[d] - \mathbf{f}_{\mathsf{p},p,\mathbf{x}}[\emptyset] = \mathsf{p}(x) - \mathsf{p}(x')$.
6. **Set Relevance (over functions) (Set-IS)**: If $\mathsf{p}_1(\mathbf{x}_S, \bar{x}_{\bar{S}}) = \mathsf{p}_2(\mathbf{x}_S, \bar{x}_{\bar{S}})$ for any $\bar{x}$, $\mathbf{f}_{\mathsf{p}_1,p,\mathbf{x}}(S) = \mathbf{f}_{\mathsf{p}_2,p,\mathbf{x}}(S)$.

The difference is that these set of axioms disregard the data distribution p.

We then define the following axioms between the **S**et function v and **E**xplanation ϕ as Axioms-SE. This is a classic set of axioms in game theory and has been widely discussed. See, e.g., the work of Lundberg and Lee [104], Shapley [140], Sundararajan and Najmi [155].

1. **Linearity from set to explanation (L-SE)**: $\Phi(\mathbf{f}_1) + \Phi(\mathbf{f}_2) = \Phi(\mathbf{f}_1 + \mathbf{f}_2)$ for any two value functions $\mathbf{f}_1$ and $\mathbf{f}_2$.
2. **Symmetry from set to explanation (S-SE)**: $\mathbf{f}(S \cup i) = \mathbf{f}(S \cup j)$ for any $S \subseteq [d] \backslash \{i, j\}$, $\Phi_i(\mathbf{f}) = \Phi_j(\mathbf{f})$.

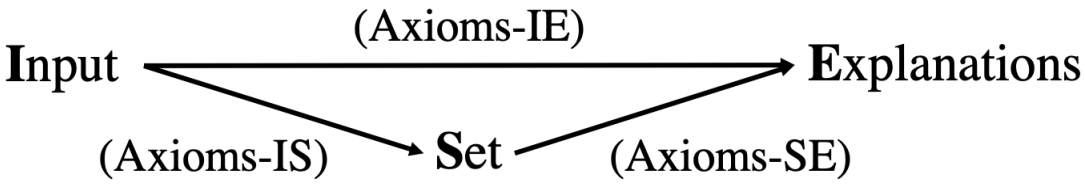


Figure 3.1: An illustration of the relation between the three sets of axioms. Axioms-IS and Axioms-SE satisfy a transfer property: broadly speaking, a pair v, ϕ satisfies Axioms-IE if v satisfies Axioms-IS and ϕ satisfies Axioms-SE.

3. **Dummy Player from set to explanation (D-SE)**: $\mathbf{f}(S \cup i) = \mathbf{f}(S)$ for any $S \subseteq [d] \backslash \{i\}$ implies $\Phi_i(\mathbf{f}) = 0$.
4. **Efficiency from set to explanation (E-SE)**: $\sum_i \Phi_i(\mathbf{f}) = \mathbf{f}([d]) - \mathbf{f}(\emptyset)$.

The Shapley value is well-known to be the unique function that satisfies the above four axioms. However, this set of axioms assumes a set function. On the other hand, the machine learning model takes a real-valued input instead of a set.

Thus, we propose the following set of axioms between the **I**nput functions (f and p) and **E**xplanation ϕ, which we term Axioms-IE. Axioms-IE can be viewed as a parallel to the axioms proposed by Sundararajan and Najmi [155]; the main difference is that our axioms take into account the in-data classifier p.

1. **Linearity from double input to explanation (L-DE)**: $\Phi_{\mathsf{p}_1,p}(\mathbf{x}) + \Phi_{\mathsf{p}_2,p}(\mathbf{x}) = \Phi_{\mathsf{p}_1+\mathsf{p}_2,p}(\mathbf{x})$ and $\Phi_{\mathsf{p},p_1}(\mathbf{x}) + \Phi_{\mathsf{p},p_2}(\mathbf{x}) = \Phi_{\mathsf{p},p_1+p_2}(\mathbf{x})$, for any functions $\mathsf{p}, \mathsf{p}_1, \mathsf{p}_2, p, p_1, p_2$.
2. **Symmetry from double input to explanation (S-DE)**: if p, p, $\mathbf{x}', \mathbf{x}$ are symmetric in the i-th and j-th dimensions, then $\Phi_{\mathsf{p},p,i}(\mathbf{x}) = \Phi_{\mathsf{p},p,j}(\mathbf{x})$.
3. **Dummy Player from double input to explanation (D-DE)**: both p, p being invariant in the i-th dimension implies $\Phi_{\mathsf{p},p,i}(\mathbf{x}) = 0$. In other words, if feature i is not used in the model, then
4. **Efficiency from double input to explanation (E-DE)**: $\sum_i \Phi_{\mathsf{p},p,i}(\mathbf{x}) = \mathsf{p}(x)p(\mathbf{x}) - \mathsf{p}(x')p(\mathbf{x}')$.

We remark that when ignoring p, this recovers a version of Axioms-IE proposed by Sundararajan and Najmi [155]: when ignoring p, it should be quite evident that value functions that depend on p (such as CES and JBshap) would easily fail to satisfy linearity and dummy defined by [155]. However, we argue that (CD-DE) is the form of dummy player axiom satisfied by CES and (D-DE) is the form of dummy player axiom satisfied by JBshap. This suggests that previous discussions on this topic may need restatements based on our proposed Axioms-IE, which takes into account the dependencies of p.

The intricate relation between Axioms-IS, Axioms-SE and Axioms-IE is captured in the following theorem.

Theorem 5. *If set function $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(\cdot)$ satisfies Axioms-IS, and explanation $\phi_{\mathbf{x},\mathsf{p},p}(\cdot)$ satisfies Axioms-SE, then $(\mathbf{f}_{\mathbf{x},\mathsf{p},p}(\cdot), \Phi(\mathbf{f}_{\mathbf{x},\mathsf{p},p}(\cdot)))$ satisfies Axioms-IE. Similarly, if set function $\mathbf{f}_{\mathbf{x},\mathsf{p},p}(\cdot)$ satisfies (L-DS), (S-DS), (D-DS), (M-DS), (N-DS), and (E-DS), and explanation $\phi_{\mathbf{x},\mathsf{p},p}(\cdot)$ is the Shapley value, then $(\mathbf{f}_{\mathbf{x},\mathsf{p},p}(\cdot), \Phi(\mathbf{f}_{\mathbf{x},\mathsf{p},p}(\cdot)))$ satisfies (L-DE), (S-DE), (D-DE), (E-DE).*

Here, theorem states the translative relationship between set of axioms from input to set, set to explanations, and input to explanations, as shown in Fig. 3.1.

This further motivates the design of axioms shown in Sec.3.3.

We remark that these sets of axioms can be useful to explanations not limited to the Shapley value. For instance, the Banzhaf value (SE version) shares a set of axioms with the Shapley value (SE version), together with our proposed (DS version of axioms), we may get resulting (DE) axioms for the Banzhaf value also. However, since this paper is focused to the Shapley value as an explanation, we do not further discuss explanations other than the Shapley value.

3.5 Estimating P(x)

In our discussion of off and on-manifold value functions, we discussed that the key caveat of on-manifold value function of CES in particular is that it requires expectation (or sampling) with respect to the conditional distributions $p(\mathbf{x}'_{\bar{S}}|\mathbf{x}_S)$. But if JBshap also requires computing the joint distribution, is it more computational efficient compared to CES at all?

We emphasize that JBshap is much more tractable computationally, since instead of the difficult conditional expectations, we only need to compute the much more amenable *joint density* $p(\mathbf{x}'_{\bar{S}}; \mathbf{x}_S)$. Moreover, we only need estimates of the data density on the space $\mathbf{x}'_{\bar{S}}; \mathbf{x}_S$ instead of the entire domain R^d. Towards this, we could use noise-contrastive estimation [66] to estimate the data density on space in the form $\mathbf{x}'_{\bar{S}}; \mathbf{x}_S$. The idea of noise-contrastive estimation is to estimate the data density of an unknown distribution by comparing it to a self-specified noise distribution. In this case, the event space where we need to query the data density $p(\mathbf{x}'_{\bar{S}}; \mathbf{x}_S)$ is the subspace $[\mathbf{x}'_{\bar{S}}; \mathbf{x}_S]$ for all possible $\mathbf{x}$ and S. We thus specify a self specified noise Q to have an constant probability for every possible $[\mathbf{x}'_{\bar{S}}; \mathbf{x}_S]$.

To estimate $p(\mathbf{x}'_{\bar{S}}; \mathbf{x}_S)$, we can then train an Out-of-Distribution (OOD) classifier model $\text{OOD}(\mathbf{x})$ that outputs 1 when the input $\mathbf{x} \in D$ comes from the true data and outputs 0 when the input $\mathbf{x} \in Q$ is self-generated noise, and control the input to the training of the OOD classifier to have balanced true and noise data. When the OOD classifier converges, $\text{OOD}(\mathbf{x}) = p(\mathbf{x} \in D|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{x}\in D)}{p(\mathbf{x}|\mathbf{x}\in D)+p(\mathbf{x}|\mathbf{x}\in Q)}$. By simple algebra, we can then get $p(\mathbf{x}|\mathbf{x} \in D) = p(\mathbf{x}|\mathbf{x} \in Q) \times \frac{\text{OOD}(\mathbf{x})}{1-\text{OOD}(\mathbf{x})}$. We emphasize that since we do not require sampling from $p(\mathbf{x}'_{\bar{S}}; \mathbf{x}_S)$, and the subspace of $\mathbf{x}'_{\bar{S}}; \mathbf{x}_S$ is much more tractable compared to the real space R^d, the estimation of $p(\mathbf{x}'_{\bar{S}}; \mathbf{x}_S)$ is a much easier problem than estimating $p(\mathbf{x})$ for general $\mathbf{x}$. Furthermore, estimating even the full joint $p(\mathbf{x})$, is much easier than computing the conditional expectations $p(\mathbf{x}'_{\bar{S}}|\mathbf{x}_S)$ (to provide a sense of the difficulty of the latter: even for simpler probabilistic graphical models, it is always tractable to compute the unnormalized joint density, while intractable in general to compute conditional marginals.)

While the calculation of JBshap only requires learning $p(\mathbf{x})$ by training an OOD classifier, the calculation of CES either requires learning $p(\mathbf{x})$ and sampling from it for each query (CES-Sample), or retraining a new model that estimates the conditioned value (CES-Supervised). Thus, the computation for calculating JBshap is much more affordable compared to the calculation of CES value functions, as we verify in the experiments.

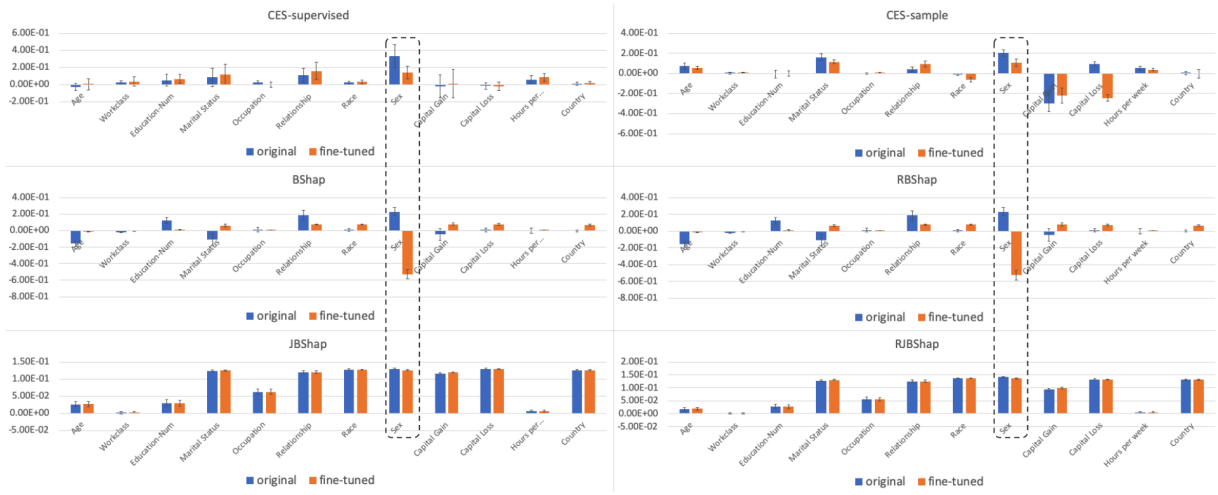


Figure 3.2: Global Shapley values for different value functions on the UCI Adult dataset; with two bars for each feature: on an original model (blue, left) and a fine-tuned model (orange, right). The importance for "sex" feature (which is boxed) of RBshap, Bshap, and CES is significantly reduced after the fine-tuning, while the importance for "sex" feature of JBshap is almost unchanged.

3.6 Experiments

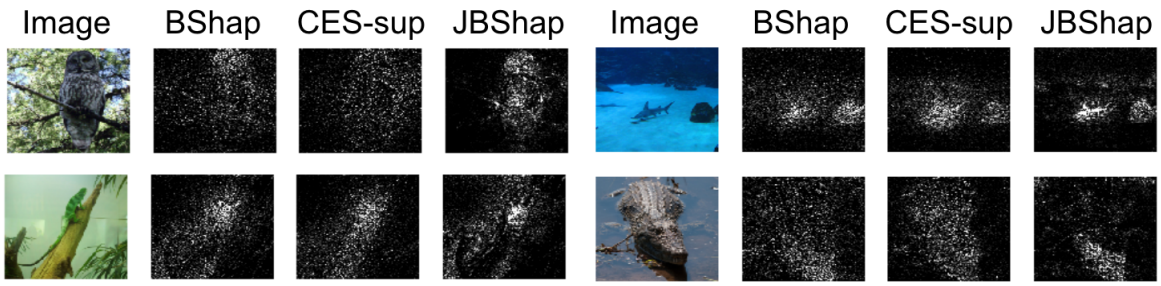


Figure 3.3: Visualization of Shapley values for JBshap, Bshap, CES-Supervised on Imagenet.

3.6.1 Robustness to off-manifold manipulation.

Recall that in Sec. 3.2.3, we have shown that both Bshap and CES are not strong $T-$robust, and thus are prone to manipulations in low-density regions. To evaluate the robustness to off-manifold manipulations in practice, we perform an empirical study on the UCI Census Income data dataset [40] for value functions Bshap, RBshap, JBshap, RJBshap, CES-Supervised, CES-Sample. Given a pretrained biased model, we want to test whether we are able to hide the dependency of a biased feature in the Shapley value by fine-tuning the model only on off-manifold regions.

We trained a four-layer neural network to predict whether an individual's income exceeds \$50k based on demographic features in the data based on the UCI income Data. To create a biased model, we add 0.1 times the feature value of "sex" to the individual's income following the setting

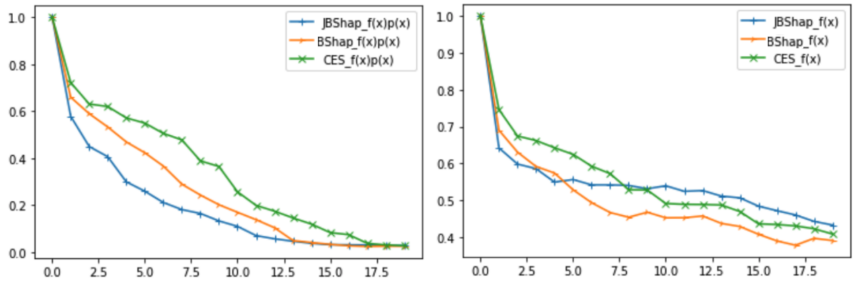


Figure 3.4: Deletion curve for JBShap, BShap, and CES-supervised on Imagenet for joint density (left) and model output (right).

of [147]. We then train a "fine-tuned network" such that the dependency of the feature "sex" is set to the negative off the data manifold (while it is positive on the data manifold), while making the same prediction as the original biased model on over 99% of the testing points. To validate that the bias is not hidden in the 1% testing point with different predictions, we find that the average difference (L1 Loss) in the prediction probability for data where the fine-tuned model and the original model disagrees (and agrees) is 0.07 (0.01), demonstrating that the model prediction for fine-tuned model and original model are close for all test data points.

For CES-Supervised, we train a "fine-tuned network" for the surrogate model g, with the constraint that the fine-tuned network behaves similar to the original surrogate model on the on-manifold points and has a similar empirical MSE loss as the original surrogate model (validation MSE loss for the fine-tuned model decreased from 0.0278 ± 0.001 to 0.0277 ± 0.001, while training MSE Loss increased from 0.0136 ± 0.001 to 0.0139 ± 0.001). To determine off-manifold data, we train an OOD detector and use noise-contrastive estimation to obtain $p(\mathbf{x})$ to calculate JBshap and CES-Sample. For numerical stability, we clip the output the of the OOD classifier between the range $[0.01, 0.99]$. For CES-Sample, we sample 100 points per value function with $p(\mathbf{x})$. We then average the Shapley value based on different value functions on 100 testing points with feature of "sex" having value 1 to obtain the global Shapley value. We do 5 separate runs with different data splits (via random seeds), we report the global Shapley value for different value functions in the top figure with error bars representing the standard deviation. We normalize the global Shapley value such that the sum of absolute value of Shapley value for all features sum up to one.

We present our results in Figure 3.2; each feature has two bars, each corresponding to the average normalized Shapley values, of the original model (left) and the fine-tuned model (right). We observe that the importance of the feature "sex" of the fine-tuned model is strongly reduced for existing value functions Bshap, RBshap and has a negative value for the fine-tuned model. For CES-Sample and CES-Supervised, the respective Shapley value of the feature "sex" is not perturbed as severely as Bshap and RBshap, but they are also reduced more than half compared to the original. On the other hand, the Shapley value for JBshap and RJBshap are only slightly reduced for the fine-tuned model. This verifies our theoretical analysis that JBshap is robust to off-manifold manipulations, in contrast to existing value functions. We also note that the Shapley value of JBshap for all features have standard deviation less than 0.01, demonstrating the robustness of JBshap over different OOD classifiers and data splits.

3.6.2 Visualization on High Dimensional Data.

In this section, we visualize the Shapley values for different value functions on image data. We perform experiments on Imagenet subset with the first 50 classes (the total number of classes is 1000). We first fine-tune a Resnet-18 model that reaches 80% top-1 accuracy as our base model. We then train a masked surrogate model with the same architecture to calculate CES-Supervised, and we train an OOD classifier to estimate $p(\mathbf{x})$ by noise contrastive estimation to calculate JBshap. To calculate the Shapley value, we use the permutation sampling approach introduced in Castro et al. [19] with 10 permutations per image. We compare the Shapley values from different value functions: Bshap, JBshap, CES-Supervised. Note that we omit other variants of value function since they require more than one function pass during the calculation of the value function, which will make the computation infeasible on high-dimension. The training of the surrogate model takes around 2976 seconds, while the training of the OOD classifier only take around 74 seconds. The cost of the training of the surrogate model will become even more costly if the model is trained on the full set of imagenet, not just a subset of imagenet. Thus, the additional training time for JBshap is much lower than the additional training time for CES-Supervised. The Shapley value computation for each image takes around 1800 seconds, which is equal for Bshap, JBshap, and CES-supervised. All computations are done on a single 1080-Ti GPU. The visualization results are shown in Figure 3.3, and we can observe that while the Shapley value for different value functions all focus on the object in the image, the Shapley value for JBshap concentrates on a smaller part of the image. For instance, in the bottom right image, Bshap and CES-supervised focuses on the whole alligator, while JBshap focuses more specifically on the head on the alligator. One possible reason is that Bshap and CES-supervised captures pixels that contributes to the model prediction, while JBshap focuses on pixels that contributes to both the data density of the image and the model prediction.

Quantitative evaluation on ImageNet We perform an evaluation for the Shapley values based on different variants of value function, where we choose the well-known deletion evaluation criteria for our results in ImageNet over the average of 50 data points, where we remove the top pixels by Shapley value and evaluate how the function decreases (lower is better) [9, 136] in Fig. 3.4, where we show the resulting joint density $f(x)p(x)$ in the left and model value $f(x)$ in the right after removing top pixels, where the x-axis is the percent of pixels. The area under curve (AUC) (where smaller is better) of $f(x)p(x)$ for JBShap, Bshap, CES is $0.207, 0.27, 0.348$ respectively, and the AUC for $f(x)$ for JBShap, Bshap, CES is $0.549, 0.504, 0.556$ respectively. Thus, we hypothesize that JBShap highlights features that affect both $f(x)$ and $p(x)$, and Bshap highlights pixels that only affect $f(x)$. We also report the rank correlation version of sensitivity-n [5], which measures the correlation between the function drop and sum of Shapley values for random sets of pixels, where the random sets of pixels is set to $1-20$ percent of pixels, and the average sensitivity-n rank correlation for JBShap, Bshap, CES, are $0.182 \pm 0.007, 0.132 \pm 0.004, 0.119 \pm 0.01$ respectively.

3.7 Conclusion

In this work, we discussed the issues of interventional and conditional value functions, including different implementation variants of conditional value functions. To address these issues, we propose JBshap, which uniquely satisfies a set of axioms considering both the model and data manifold, while being robust to manipulation and computationally efficient.

Chapter 4

Using Robustness analysis to Evaluate and Design Feature Set Explanations

There is an increasing interest in machine learning models to be credible, fair, and more generally *interpretable* [39]. Researchers have explored various notions of model interpretability, ranging from trustability [129], fairness of a model [191], to characterizing the model's weak points [89, 179]. Even though the goals of these various model interpretability tasks vary, the vast majority of them use so called *feature based explanations*, that assign importance to individual features [5, 10, 11, 20, 104, 129, 142, 144, 156, 187, 193]. There have also been a slew of recent *evaluation* measures for feature based explanations, such as completeness [156], sensitivity-n [5], infidelity [181], causal local explanation metric [126], and most relevant to the current paper, removal- and preservation-based criteria [30, 45, 124, 136]. A common thread in all these evaluation measures is that for a good feature based explanation, the most salient features are necessary, in that removing them should lead to a large difference in prediction score, and are also sufficient in that removing non-salient features should not lead to a large difference in prediction score.

Thus, common evaluations and indeed even methods for feature based explanations involve measuring the function difference after "removing features", which in practice is done by setting the feature value to some reference value (also called baseline value sometimes). However, this would favor feature values that are far way from the baseline value (since this corresponds to a large perturbation, and hence is likely to lead to a function value difference), causing an intrinsic bias for these methods and evaluations. For example, if we set the feature value to black in RGB images, this introduces a bias favoring bright pixels: explanations that optimize such evaluations often omit important dark objects such as a dark-colored dog. An alternative approach to "remove features" is to sample from some predefined distribution or a generative model [20]. This nevertheless in turn incurs the bias inherent to the generative model, and accurate generative models that approximate the data distribution well might not be available in all domains.

In this work, instead of defining prediction changes with "removal" of features (which introduces biases as we argued), we alternatively consider the use of small but *adversarial perturbations*. It is natural to assume that adversarial perturbations on irrelevant features should be ineffective, while those on relevant features should be effective. We can thus measure the necessity of a set of relevant features, provided by an explanation, by measuring the consequences

Figure 4.1: Illustration of our explanation highlighting both pertinent positive and negative features that support the prediction of "2". The blue circled region corresponds to pertinent positive features that when its value is perturbed (from white to black) will make the digit resemble "7"; while the green and yellow circled region correspond to pertinent negative features that when turned on (black to white) will shape the digit into "0","8", or "9".

of adversarially perturbing their feature values: if the features are indeed relevant, this should lead to an appreciable change in the predictions. Complementarily, we could measure the sufficiency of the set of relevant features via measuring consequences of adversarially perturbing its complementary set of irrelevant features: if the perturbed features are irrelevant, this should not lead to an appreciable change in the predictions. We emphasize that by our definition of "important features", our method may naturally identify both pertinent positive and pertinent negative features [35] since both pertinent positive and pertinent negative features are the most susceptible to adversarial perturbations, and we demonstrate the idea in Figure 4.1. While exactly computing such an effectiveness measure is NP-hard [83], we can leverage recent results from test-time robustness [18, 108], which entail that perturbations computed by adversarial attacks can serve as reasonably tight upper bounds for our proposed evaluation. Given this adversarial effectiveness evaluation measure, we further design feature based explanations that optimize this evaluation measure.

To summarize our contributions:

- We define new evaluation criteria for feature based explanations by leveraging robustness analysis involving small adversarial perturbations. These reduce the bias inherent in other recent evaluation measures that focus on "removing features" via large perturbations to some reference values, or sampling from some reference distribution.
- We design efficient algorithms to generate explanations that optimize the proposed criteria by incorporating game theoretic notions, and demonstrate the effectiveness and interpretability of our proposed explanation on image and language datasets: via our proposed evaluation metric, additional objective metrics, as well as qualitative results and a user study.[1]

4.1 Robustness Analysis for Evaluating Explanation Set

4.1.1 Problem Notation

We consider the setting of a general K-way classification problem with input space $\mathcal{X} \subseteq \mathbb{R}^d$, output space $\mathcal{Y} = \{1, \ldots, K\}$, and a predictor function $f : \mathcal{X} \to \mathcal{Y}$ where $f(\boldsymbol{x})$ denotes the output class for some input example $\boldsymbol{x} = [x_1, \ldots, x_d] \in \mathcal{X}$. Then, for a particular prediction $f(\boldsymbol{x}) = y$, a common goal of feature based explanations is to extract a compact set of relevant features with respect to the prediction. We denote the set of relevant features provided by an explanation as $S_r \subseteq U$ where $U = \{1, \ldots, d\}$ is the set of all features, and use $\overline{S_r} = U \setminus S_r$, the

[1] Code available at https://github.com/ChengYuHsieh/explanation_robustness.

complementary set of S_r, to denote the set of irrelevant features. We further use $\boldsymbol{x}_S$ to denote the features within $\boldsymbol{x}$ that are restricted to the set S.

4.1.2 Evaluation through Robustness Analysis

A common thread underlying evaluations of feature based explanations [124, 136], even ranging over axiomatic treatments [104, 156], is that the importance of a set of features corresponds to the change in prediction of the model when the features are removed from the original input. Nevertheless, as we discussed in previous sections, operationalizing such a removal of features, for instance, by setting them to some reference value, introduces biases (see Section 4.3 and Section 4.4.2 for formal discussion and empirical results on the impact of reference values). To finesse this, we leverage adversarial robustness, but to do so in this context, we rely on two key intuitive assumptions that motivate our method:

Assumption 1: When the values of the important features are anchored (fixed), perturbations restricted to the complementary set of features has a weaker influence on the model prediction.

Assumption 2: When perturbations are restricted to the set of important features, fixing the values of the rest of the features, even small perturbations could easily change the model prediction.

Based on these two assumptions, we propose a new framework leveraging the notion of adversarial robustness on feature subsets, as defined below, to evaluate feature based explanations.

Definition 4. Given a model f, an input $\boldsymbol{x}$, and a set of features $S \subseteq U$ where U is the set of all features, the minimum adversarial perturbation norm on $\boldsymbol{x}_S$, which we will also term *Robustness-S* of $\boldsymbol{x}$ is defined as:

$$\epsilon^*_{\boldsymbol{x}_S} = g(f, \boldsymbol{x}, S) = \left\{ \min_{\delta} \|\delta\|_p \text{ s.t. } f(\boldsymbol{x}+\delta) \neq y,\ \delta_{\overline{S}} = 0 \right\}, \tag{4.1}$$

where $y = f(\boldsymbol{x})$, $\overline{S} = U \setminus S$ is the complementary set of features, and $\delta_{\overline{S}} = 0$ means that the perturbation is constrained to be zero along features in $\overline{S}$.

Suppose that a feature based explanation partitions the input features of $\boldsymbol{x}$ into a relevant set S_r, and an irrelevant set $\overline{S_r}$, Assumption 1 implies that the quality of the relevant set can be measured by $\epsilon^*_{\boldsymbol{x}_{\overline{S_r}}}$ – by keeping the relevant set unchanged, and measuring the adversarial robustness norm by perturbing only the irrelevant set. Specifically, from Assumption 1, a larger coverage of pertinent features in set S_r entails a higher robustness value $\epsilon^*_{\boldsymbol{x}_{\overline{S_r}}}$. On the other hand, from Assumption 2, a larger coverage of pertinent features in set S_r would in turn entail a smaller robustness value $\varepsilon^*_{\boldsymbol{x}_{S_r}}$, since only relevant features are perturbed. More formally, we propose the following twin criteria for evaluating the quality of S_r identified by any given feature based explanation.

Definition 5. Given an input $\boldsymbol{x}$ and a relevant feature set S_r, we define *Robustness-*$\overline{S_r}$ and *Robustness-*S_r of the input $\boldsymbol{x}$ as the following:

$$\textit{Robustness-}\overline{S_r} = \varepsilon^*_{\boldsymbol{x}_{\overline{S_r}}}. \quad \textit{Robustness-}S_r = \varepsilon^*_{\boldsymbol{x}_{S_r}}.$$

Following our assumptions, a set S_r that has larger coverage of relevant features would yield *higher* Robustness-$\overline{S_r}$ and *lower* Robustness-S_r.

Evaluation for Feature Importance Explanations. While Robustness-$\overline{S_r}$ and Robustness-S_r are defined on sets, general feature attribution based explanations could also easily fit into the evaluation framework. Given any feature attribution method that assigns importance score to each feature, we can sort the features in descending order of importance weights, and provide the top-K features as the relevant set S_r. The size of K (or $|S_r|$), can be specified by the users based on the application. An alternative approach that we adopt in our experiments is to vary the size of set K and plot the corresponding values of Robustness-$\overline{S_r}$ and Robustness-S_r over different values of K. With a graph where the $X-$axis is the size of K and the $Y-$axis is Robustness-$\overline{S_r}$ or Robustness-S_r, we are then able to plot an evaluation curve for an explanation and in turn compute its the area under curve (AUC) to summarize its performance. A larger (smaller) AUC for Robustness-$\overline{S_r}$ (Robustness-S_r) indicates a better feature attribution ranking. Formally, given a curve represented by a set of points $\mathcal{C} = \{(x_0, y_0), \ldots, (x_n, y_n)\}$ where $x_{i-1} < x_i$, we calculate the AUC of the curve by: $AUC(\mathcal{C}) = \sum_{i=1}^{n}(y_i + y_{i-1})/2 * (x_i - x_{i-1})$.

Relation to Insertion and Deletion Criteria. We relate the proposed criteria to a set of commonly adopted evaluation metrics: the *Insertion* and *Deletion* criteria [124, 136, 152]. The Insertion score measures the model's function value when only the top-relevant features, given by an explanation, are presented in the input while the others are removed (usually by setting them to some reference value representing feature missingness). The Deletion score, on the other hand, measures the model's function value when the most relevant features are masked from the input. As in our evaluation framework, we could plot the evaluation curves for Insertion (Deletion) score by progressively increasing the number of top-relevant features. A larger (smaller) AUC under Insertion (Deletion) then indicates better explanation, as the identified relevant features could indeed greatly influence the model prediction. We note that optimizing the proposed Robustness-$\overline{S_r}$ and Robustness-S_r could roughly be seen as optimizing a lower bound for the Insertion and Deletion score respectively. This follows from the intuition: Robustness-$\overline{S_r}$ considers features that when anchored, would make the prediction most robust to "adversarial perturbation". Since adversarial perturbation is the worst case of "any arbitrary perturbations", the prediction will also be robust to different removal techniques (which essentially correspond to different perturbations) considered in the evaluation of Insertion score; The same applies to the connection between Robustness-S_r and Deletion score. We shall see in the experiment section that explanation optimizing our robustness measurements enjoys competitive performances on the Insertion / Deletion criteria.

Untargeted v.s. Targeted Explanation. Definition 4 corresponds to the untargeted adversarial robustness – a perturbation that changes the predicted class to any label other than y is considered as a successful attack. Our formulation can also be extended to **targeted adversarial robustness**, where we replace (4.1) by:

$$\epsilon^*_{x_S,t} = \left\{ \min_{\delta} \|\delta\|_p \text{ s.t. } f(\boldsymbol{x} + \delta) = t; \delta_{\overline{S}} = 0 \right\}, \tag{4.2}$$

where t is the targeted class. Using this definition, our approach will try to address the question "Why is this example classified as y instead of t" by highlighting the important features that

contrast between class y and t. Further examples of the "targeted explanations" are in the experiment section.

Robustness Evaluation on Feature Subset. It is known that computing the exact minimum distortion distance in modern neural networks is intractable [83], so many different methods have been developed to estimate the value. Adversarial attacks, such as C&W [18] and projected gradient descent (PGD) attack [108], aim to find a feasible solution of (4.1), which leads to an upper bound of $\epsilon^*_{x_S}$. They are based on gradient based optimizers which are usually efficient. On the other hand, neural network verification methods aim to provide a lower bound of $\epsilon^*_{x_S}$ to ensure that the model prediction will not change within certain perturbation range [48, 146, 171, 172, 174, 188, 189]. The proposed framework can be combined with any method that aims to approximately compute (4.1), including attack, verification, and some other statistical estimations for more discussions on estimating adversarial robustness for different types of model). However, for simplicity we only choose to evaluate (4.1) by the state-of-the-art PGD attack [108], since the verification methods are too slow and often lead to much looser estimation as reported in some recent studies [134]. Our additional constraint restricting perturbation to only be on a subset of features specifies a set that is simple to project onto, where we set the corresponding coordinates to zero at each step of PGD.

4.2 Extracting Relevant Features through Robustness Analysis

Our adversarial robustness based evaluations allow us to evaluate any given feature based explanation. Here, we set out to design new explanations that explicitly optimize our evaluation measure. We focus on feature set based explanations, where we aim to provide an important subset of features S_r. Given our proposed evaluation measure, an optimal subset of feature S_r would aim to maximize (minimize) Robustness-$\overline{S_r}$ (Robustness-S_r), under a cardinality constraint on the feature set, leading to the following set of optimization problems:

$$\underset{S_r \subseteq U}{\text{maximize}} \; g(f, \boldsymbol{x}, \overline{S_r}) \;\; \text{s.t.} \;\; |S_r| \leq K \tag{4.3}$$

$$\underset{S_r \subseteq U}{\text{minimize}} \; g(f, \boldsymbol{x}, S_r) \;\; \text{s.t.} \;\; |S_r| \leq K \tag{4.4}$$

where K is a pre-defined size constraint on the set S_r, and $g(f, \boldsymbol{x}, S)$ computes the the minimum adversarial perturbation from (4.1), with set-restricted perturbations.

It can be seen that this sets up an adversarial game for (4.3) (or a co-operative game for (4.4)). In the adversarial game, the goal of the feature set explainer is to come up with a set S_r such that the minimal adversarial perturbation is as large as possible, while the adversarial attacker, given a set S_r, aims to design adversarial perturbations that are as small as possible. Conversely in the co-operative game, the explainer and attacker cooperate to minimize the perturbation norm. Directly solving these problems in (4.3) and (4.4) is thus challenging, which is exacerbated by the discrete input constraint that makes it intractable to find the optimal solution. We therefore propose a greedy algorithm in the next section to estimate the optimal explanation sets.

4.2.1 Greedy Algorithm to Compute Optimal Explanations

We first consider a greedy algorithm where, after initializing S_r to the empty set, we iteratively add to S_r the most promising feature that optimizes the objective at each local step until S_r reaches the size constraint. We thus sequentially solve the following sub-problem at every step t:

$$\arg\max_i \; g(f, \boldsymbol{x}, \overline{S_r^t \cup i}), \text{ or } \arg\min_i \; g(f, \boldsymbol{x}, S_r^t \cup i), \; \forall i \in \overline{S_r^t} \tag{4.5}$$

where S_r^t is the relevant set at step t, and $S_r^0 = \emptyset$. We repeat this subprocedure until the size of set S_r^t reaches K. A straightforward approach for solving (4.5) is to exhaustively search over every single feature. We term this method **Greedy**. While the method eventually selects K features for the relevant set S_r, it might lose the sequence in which the features were selected. One approach to encode this order would be to output a feature explanation that assigns higher weights to those features selected earlier in the greedy iterations.

4.2.2 Greedy by Set Aggregation Score

The main downside of using the greedy algorithm to optimize the objective function is that it ignores the interactions among features. Two features that may seem irrelevant when evaluated separately might nonetheless be relevant when added simultaneously. Therefore, in each greedy step, instead of considering how each individual feature will marginally contribute to the objective $g(\cdot)$, we propose to choose features based on their expected marginal contribution when added to the union of S_r and a random subset of unchosen features. To measure such an aggregated contribution score, we draw from cooperative game theory literature [41, 67] to reduce this to a linear regression problem. Formally, let S_r^t and $\overline{S_r^t}$ be the ordered set of chosen and unchosen features at step t respectively, and $\mathcal{P}(\overline{S_r^t})$ be all possible subsets of $\overline{S_r^t}$. We measure the expected contribution that including each unchosen feature to the relevant set would have on the objective function by learning the following regression problem:

$$\boldsymbol{w}^t, c^t = \arg\min_{\boldsymbol{w}, c} \sum_{S \in \mathcal{P}(\overline{S_r^t})} ((\boldsymbol{w}^T b(S) + c) - v(S_r^t \cup S))^2, \tag{4.6}$$

where $b : \mathcal{P}(\overline{S_r^t}) \to \{0,1\}^{|\overline{S_r^t}|}$ is a function that projects a set into its corresponding binary vector form: $b(S)[j] = \mathbb{I}(\overline{S_r^t}[j] \in S)$, and $v(\cdot)$ is set to be the objective function in (4.3) or (4.4): $v(S_r) = g(f, \boldsymbol{x}, \overline{S_r})$ for optimizing (4.3); $v(S_r) = g(f, \boldsymbol{x}, S_r)$ for optimizing (4.4). We note that $\boldsymbol{w}^t$ corresponds to the well-known Banzhaf value [13] when $S_r^t = \emptyset$, which can be interpreted as the importance of each player by taking coalitions into account [41]. Hammer and Holzman [67] show that the Banzhaf value is equivalent to the optimal solution of linear regression with pseudo-Boolean functions as targets, which corresponds to (4.6) with $S_r^t = \emptyset$. At each step t, we can thus treat the linear regression coefficients $\boldsymbol{w}^t$ in (4.6) as each corresponding feature's expected marginal contribution when added to the union of S_r and a random subset of unchosen features.

We thus consider the following set-aggregated variant of our greedy algorithm in the previous section, which we term **Greedy-AS**. In each greedy step t, we choose features that are expected to contribute most to the objective function, i.e. features with highest (for (4.3)) or lowest (for

(4.4)) aggregation score (Banzhaf value), rather than simply the highest marginal contribution to the objective function as in vanilla greedy. This allows us to additionally consider the interactions among the unchosen features when compared to vanilla greedy. The chosen features each step are then added to S_r^t and removed from $\overline{S_r^t}$. When S_r^t is not $\emptyset$, the solution of (4.6) can still be seen as the Banzhaf value where the players are the unchosen features in $\overline{S_r^t}$, and the value function computes the objective when a subset of players is added into the current set of chosen features S_r^t. We solve the linear regression problem in (4.6) by sub-sampling to lower the computational cost, and we validate the effectiveness of Greedy and Greedy-AS in the experiment section. [2]

4.3 Bias for Reference-Value Methods

In Sec. 4.3.1, we show that many explanations are biased to reference value (IG, SHAP, LRP, DeepLift). For example, if the reference value is a zero vector (which means black pixels in image), then any black pixel will get a zero attribution value no matter if the object of interest is actually back. We note that the Expected Gradient [44] is not biased to reference value by this theoretic definition since the baseline is actually a distribution. However, for feature values that are close to the distribution of , the attribution score will be lower (but not 0 as our theoretic definition), when feature values are far from the distribution of , the the attribution score will be larger, which still has some biased involve. We leave further investigation of the problem to future work to use more advanced analysis to quantify such a bias for explanations when the baseline follows a distribution.

In Sec. 4.3.2, we added theoretical analysis that when a feature is equal to the replaced reference value, no matter how important it actually is, it will not contribute to the Deletion score nor the Insertion score. However, the lower the deletion score is better, and the higher the insertion score is better, and choosing an important feature that corresponds to the reference value will inevitably not improve the Deletion score and Insertion score. The reference values such as blurring out or adding noise may still make the original features unchanged (such as when the main part of the image is already blurred, blurring the image will not change the image, and thus the main part of the image is biased to blurred baseline).

4.3.1 Bias for Reference-Value Based Explanations

Theorem 6. *For any reference-value based explanation with the form* $\phi(f,x,x') = (x-x') \otimes K(f,x,x')$ *for any meta-function K, we know that if* $x_i = x_i'$, $\phi_i(f,x) = 0$. *That is, when ever a feature value coincides with the baseline value* x'*, it will get 0 attribution even if it is the main feature contributing to the prediction. We call such explanations biased to the reference value* x'.

Corollary 1. *IG, LRP, DeepLift are biased to the reference value* x'.

Following the definition in [5], let $\frac{\partial^g x_c}{\partial x_i} = \sum_{p \in P_{ic}} \prod w_p \prod g(z)_p$ where P_{ic} is the set of all paths that connect an unit i to unit c in a deep neural network, w_p are the weights existing in path p, z be the value of an unit before nonlinear activation, and $g()$ be any generic function. Note that

[2] We found that concurrent to our work, greedy with choosing the players with the highest restricted Banzhaf was used in Elkind et al. [43].

if $g() = f'()$, the definition corresponds to the standard partial derivative of unit c to unit i.

$$\phi_i^{IG}(f,x) = (x_i - x_i') \cdot K(f,x,x'), \text{ where } K(f,x,x') = \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} d\alpha \tag{4.7}$$

$$\phi_i^{LRP}(f,x) = (x_i - x_i') \cdot K(f,x,x'), \text{ where } x' = 0 \text{ and } K(f,x,x') = \frac{\partial^g f(x)}{\partial x_i} \text{ with } g = \frac{f(z)}{z} \tag{4.8}$$

$$\phi_i^{DeepLift}(f,x) = (x_i - x_i') \cdot K(f,x,x'), \text{ where } K(f,x,x') = \frac{\partial^g f(x)}{\partial x_i} \text{ with } g = \frac{f(z) - f(z')}{z - z'} \tag{4.9}$$

For example, if the reference value x' is a zero vector (which means black pixels in image), then any black pixel will get a zero attribution value no matter if the object of interest is actually back.

Theorem 7. *For any average reference-value based explanation with the form $\phi(f,x,x') = \mathsf{E}_{x' \sim p_b}[(x - x') \otimes K(f,x)]$ for some meta function K, if $x_i = \mathsf{E}_{x' \sim p_b}[x_i']$, $\phi_i(f,x) = 0$. That is, they are biased to the reference value $\mathsf{E}_{x' \sim p_b}[x']$.*

Corollary 2. *Averaging LRP over multiple baselines are bias to the reference value $\mathsf{E}_{x' \sim p_b}[x']$.*

$$\phi_i^{avgLRP}(f,x,x') = \mathsf{E}_{x' \sim p_b}[(x_i - x_i') \cdot K(f,x)], \quad \text{where } K(f,x) = \frac{\partial^g f(x)}{\partial x_i} \text{ with } g = \frac{f(z)}{z} \tag{4.10}$$

Theorem 8. *The baseline Shapley value [155] is biased to reference value x'.*

$$\phi_i^{BShap}(f,x,x') = \sum_{S \subseteq N \setminus i} \frac{|S|! \cdot (|N| - |S| - 1)!}{|N|!} (v(S \cup i) - v(S)) \tag{4.11}$$

where $v(s) = f(x_S; x'_{N \setminus S})$ and N is the set of all features.

These above analysis shows that more explanations are also biased to reference value. We note that by Expected Gradient [44] is not biased to reference value by this theoretic definition since the baseline x' is actually a distribution. However, for feature values x_i that are close to the distribution of x_i', the attribution score will be lower (but not 0 as our theoretic definition), when feature values x_i are far from the distribution of x_i', the the attribution score will be larger, which still shows some level of bias. We leave this to future work to use more advanced analysis techniques to quantify such a biased for explanations when the baseline follows a distribution.

4.3.2 Bias for Reference-value Based Evaluations

Definition 6. (Informal) Given a data point x, a reference point x', the deletion score of a model f given a set of important features S (and denote the complement of S as $\bar{S}$), which we abbreviate

as $\text{DEL}(x, x', S)$, is defined as $f(x'_S; x_{\bar{S}})$. Here, x' can be a fixed value or a random value, or even a blurred component (by a slight abuse of notations). Similarly, we define the insertion score for a data point x, a reference point x', important set of features S, and a model f as $\text{INSR}(x, x', S) = f(x_S; x'_{\bar{S}})$.

Note that the lower the DEL score is better, and the higher the INSR score is better.

Theorem 9. *If $x_i = x'_i$, then $DEL(x, x', S) = DEL(x, x', S \cup i)$ for $i \not\subseteq S$. Similarly, if $x_i = x'_i$, $INSR(x, x', S) = INSR(x, x', S \cup i)$ for $i \not\subseteq S$*

The implication of Thm. 9 is that if a feature happens to correspond to the reference value, no matter how important it actually is, it will not contribute to the DEL score nor the INSR score. However, the lower the DEL score is better, and the higher the INSR score is better, and choosing an important feature that corresponds to the reference value will inevitably not improve the DEL score and INSR score. Thus, DEL score and INSR score will deem feature with the same value as the reference value as non-important even if the features are actually crucial.

4.4 Experiments

In this section, we first evaluate different model interpretability methods on the proposed criteria. We justify the effectiveness of the proposed Greedy-AS. We then move onto further validating the benefits of the explanations extracted by Greedy-AS through comparisons to various existing methods both quantitatively and qualitatively. Finally, we demonstrate the flexibility of our method with the ability to provide targeted explanations as mentioned in Section 4.1.2. We perform the experiments on two image datasets, MNIST [98] and ImageNet [33], as well as a text classification dataset, Yahoo! Answers [190]. On MNIST, we train a convolutional neural network (CNN) with 99% testing accuracy. On ImageNet, we deploy a pre-trained ResNet model obtained from the Pytorch library. On Yahoo! Answers, we train a BiLSTM sentence classifier which attains testing accuracy of 71%.

Setup. In the experiments, we consider $p = 2$ for $\|\cdot\|_p$ in (4.1) and (4.2). We note that (4.1) is defined for a single data example. Given n multiple examples $\{\boldsymbol{x}_i\}_{i=1}^n$ with their corresponding relevant sets provided by some explanation $\{S_i\}_{i=1}^n$, we compute the overall Robustness-$\overline{S_r}$ (-S_r) by taking the average: for instance, $\frac{1}{n}\sum_{i=1}^n g(f, \boldsymbol{x}_i, \overline{S_i})$ for Robustness-$\overline{S_r}$. We then plot the evaluation curve as discussed in Section 4.1.2 and report the AUC for different explanation methods. For all quantitative results, we report the average over 100 random examples. For the baseline methods, we include vanilla gradient (Grad) [142], integrated gradient (IG) [156], and expected gradient (EG) [44, 152] from gradient-based approaches; leave-one-out (LOO) [101, 187], SHAP [104] and black-box meaningful perturbation (BBMP) (only for image examples) [45] from perturbation-based approaches [5]; counterfactual explanation (CFX) proposed by Wachter et al. [168]; Anchor [130] for text examples; and a Random baseline that ranks feature importance randomly. Following common setup [5, 156], we use zero as the reference value for all explanations that require baseline.

Table 4.1: AUC of Robustness-$\overline{S_r}$ and Robustness-S_r for various explanations on different datasets. The higher the better for Robustness-$\overline{S_r}$; the lower the better for Robustness-S_r.

Datasets	Explanations	Grad	IG	EG	SHAP	LOO	BBMP	CFX	Random	Greedy-AS
MNIST	Robustness-$\overline{S_r}$	88.00	85.98	93.24	75.48	74.14	78.58	69.88	64.44	**98.01**
	Robustness-S_r	91.72	91.97	91.05	101.49	104.38	176.61	102.81	193.75	**82.81**
ImageNet	Robustness-$\overline{S_r}$	27.13	26.01	26.88	18.25	22.29	21.56	27.12	17.98	**31.62**
	Robustness-S_r	45.53	46.28	48.82	60.02	58.46	158.01	46.10	56.11	**43.97**
Yahoo!Answer	Robustness-$\overline{S_r}$	1.97	1.86	1.96	1.81	1.74	-	1.95	1.71	**2.13**
	Robustness-S_r	2.91	3.14	2.99	3.34	4.04	-	2.96	7.64	**2.41**

4.4.1 Robustness Analysis on Model Interpretability Methods

Here we compare Greedy-AS and various existing explanation methods under the proposed evaluation criteria Robustness-$\overline{S_r}$ and Robustness-S_r. We list the results in Table 4.1.

Comparisons between Different Explanations. Furthermore from Table 4.1, we observe that the proposed Greedy-AS consistently outperforms other explanation methods on both criteria. On one hand, this suggests that the proposed algorithm indeed successfully optimizes towards the criteria; on the other hand, this might indicate the proposed criteria do capture different characteristics of explanations which most of the current explanations do not possess. Another somewhat interesting finding from the table is that while Grad has generally been viewed as a baseline method, it nonetheless performs competitively on the proposed criteria. We conjecture the phenomenon results from the fact that Grad does not assume any reference value as opposed to other baselines such as LOO which sets the reference value as zero to mask out the inputs. Indeed, it might not be surprising that Greedy-AS achieves the best performances on the proposed criteria since it is explicitly designed for so. To more objectively evaluate the usefulness of the proposed explanation, we demonstrate different advantages of our method by comparing Greedy-AS to other explanations quantitatively on existing commonly adopted measurements, and qualitatively through visualization in the following subsections.

4.4.2 Evaluating Greedy-AS

The Insertion and Deletion Metric. To further justify the proposed explanation not only performs well on the very metric it optimizes, we evaluate our method on the suite of quantitative measurements mentioned above: the Insertion and Deletion criteria [124, 136, 152]. Recall that evaluations on Insertion and Deletion score require specifying a reference value to represent feature missingness. Here, we first focus on the results when the reference values are randomly sampled from an uniform distribution, i.e., $\mathcal{U}(0,1)$ for image inputs and random word vector for text inputs, and we shall discuss the impact on varying such reference value shortly. We plot the evaluation curves and report corresponding AUCs in Table 4.4. On these additional two criteria, we observe that Greedy-AS performs favorably against other explanations. The results further validate the benefits of the proposed criteria where optimizing Robustness-$\overline{S_r}$ (-S_r) has tight connection to optimizing the Insertion (Deletion) score. We note that on ImageNet, SHAP obtains a better performance under the Deletion criterion. We however suspect such performance comes from adversarial artifacts (features vulnerable to perturbations while not being semantically

meaningful) since SHAP seems rather noisy on ImageNet (as shown in Figure 4.3), and the Deletion criterion has been observed to favor such artifacts in previous work [20, 30]. We note that although Greedy-AS exploits regions that are most susceptible to adversarial attacks, such regions may still be meaningful as shown in our visualization result.

Impact and Potential Bias of Reference Value. From Table 4.4, one might wonder why explanations like IG and SHAP would suffer relatively low performance although some of these methods (e.g., SHAP) are intentionally designed for optimizing Insertion- and Deletion-like measurements. We anticipate that such inferior performances are due to the mismatch between the intrinsic reference values used in the explanations and the ones used in the evaluation (recall that we set the intrinsic reference value to zero for all explanation methods, but utilize random value for Insertion and Deletion). To validate the hypothesis, we evaluate all explanations on Insertion and Deletion criteria with different reference values (0, 0.25, 0.5, 0.75, 1), and we show empirical analysis on how different reference values would affect the evaluation results of different explanation methods in Table 4.2 and Table 4.3.

Table 4.2: AUC of the Insertion and Deletion criteria with different reference values for various explanations on MNIST. The higher the better for Insertion; the lower the better for Deletion.

Reference Values	Explanations	Grad	IG	SHAP	LOO	BBMP	CFX	EG	Random	Greedy-AS
Reference value = rand	Insertion	174.18	177.12	125.93	121.99	108.97	102.05	228.64	51.71	**270.75**
rand $\sim$ Unifrom(0, 1)	Deletion	153.58	150.90	213.32	274.77	587.08	137.69	113.21	312.07	**94.24**
Reference value = 0	Insertion	393.25	**802.57**	656.55	706.63	212.12	81.59	450.37	120.65	502.32
	Deletion	214.81	134.64	**66.36**	119.91	362.95	840.78	138.57	453.01	252.44
Reference value = 0.25	Insertion	293.92	424.45	412.87	453.29	142.00	77.33	341.05	78.73	**455.74**
	Deletion	162.24	133.06	115.88	204.75	442.23	354.05	161.28	426.21	**90.89**
Reference value = 0.5	Insertion	218.18	317.55	136.51	189.87	42.76	86.80	382.31	49.15	**479.92**
	Deletion	209.83	258.50	271.11	424.66	549.11	**90.50**	195.47	392.69	155.59
Reference value = 0.75	Insertion	220.90	204.92	79.28	125.73	21.97	163.46	285.66	67.35	**325.47**
	Deletion	305.17	305.38	481.67	712.46	708.93	**89.38**	250.57	451.18	176.85
Reference value = 1	Insertion	234.61	206.54	89.83	128.71	25.31	229.02	276.48	81.44	**313.66**
	Deletion	364.01	372.18	647.68	956.88	823.90	**101.31**	310.50	495.91	223.99

We validate that zero-baseline SHAP and IG perform much stronger when the reference value used in Insertion / Deletion is closer to zero (matching the intrinsically-used baseline) and perform significantly worse when the reference value is set to 0.75, 1, or $\mathcal{U}(0,1)$. On the other hand, we observe EG that does not rely on a single reference point (but instead averaging over multiple baselines) performs much more stably across different reference values. Finally, we see that Greedy-AS performs stably among the top across different reference values, which could be the merits of not assuming any baselines (but instead consider the worst-case perturbation). These empirical results reveal potential risk of evaluations (and indeed explanations) that could largely be affected by the change of baseline values.

4.4.3 Qualitative Results

Image Classification. To complement the quantitative measurements, we show several visualization results on MNIST and ImageNet in Figure 4.2 and Figure 4.3. On MNIST, we observe

Table 4.3: AUC of the Insertion and Deletion criteria with different reference values for various explanations on ImageNet. The higher the better for Insertion; the lower the better for Deletion.

Reference Values	Explanations	Grad	IG	SHAP	LOO	BBMP	CFX	EG	Random	Greedy-AS
Reference value = rand	Insertion	86.16	109.94	28.06	63.90	135.98	97.33	150.81	31.73	**183.66**
rand $\sim$ Unifrom(0, 1)	Deletion	276.78	256.51	**143.27**	290.10	615.13	281.12	244.88	314.82	219.52
Reference value = 0	Insertion	125.32	**214.85**	85.57	61.37	138.00	135.31	183.47	69.57	182.61
	Deletion	313.04	260.75	**181.11**	262.78	665.49	333.98	288.52	312.66	234.34
Reference value = 0.25	Insertion	291.06	**329.24**	39.74	163.81	243.92	287.58	245.46	103.63	293.30
	Deletion	408.95	395.37	328.50	365.03	686.75	421.56	433.78	516.56	**322.50**
Reference value = 0.5	Insertion	301.78	343.77	45.77	178.18	250.25	270.03	**348.66**	129.65	305.87
	Deletion	431.17	412.70	345.84	388.64	689.08	421.89	420.31	546.30	**321.11**
Reference value = 0.75	Insertion	222.63	217.26	29.39	131.92	243.44	215.11	248.74	87.38	**286.40**
	Deletion	364.28	340.24	279.51	325.71	701.97	375.87	329.11	402.97	**262.36**
Reference value = 1	Insertion	109.89	136.10	55.29	89.26	135.06	109.44	170.96	59.23	**189.96**
	Deletion	245.57	218.09	207.57	234.28	658.54	329.20	203.84	217.39	**191.96**

Table 4.4: AUC of the Insertion and Deletion criteria for various explanations on different datasets. The higher the better for Insertion; the lower the better for Deletion.

Datasets	Explanations	Grad	IG	EG	SHAP	LOO	BBMP	CFX	Random	Greedy-AS
MNIST	Insertion	174.18	177.12	228.64	125.93	121.99	108.97	102.05	51.71	**270.75**
	Deletion	153.58	150.90	113.21	213.32	274.77	587.08	137.69	312.07	**94.24**
ImageNet	Insertion	86.16	109.94	150.81	28.06	63.90	135.98	97.33	31.73	**183.66**
	Deletion	276.78	256.51	244.88	**143.27**	290.10	615.13	281.12	314.82	219.52
Yahoo!Answers	Insertion	0.06	0.06	0.20	0.07	0.18	-	0.05	0.10	**0.21**
	Deletion	2.57	2.96	2.07	2.23	2.07	-	2.35	2.63	**1.56**

that existing explanations tend to highlight mainly on the white pixels in the digits; among which SHAP and LOO show less noisy explanations comparing to Grad and IG. On the other hand, the proposed Greedy-AS focuses on both the "crucial positive" (important white pixels) as well as the "pertinent negative" (important black regions) that together support the prediction. For example, in the first row, a 7 might have been predicted as a 4 or 0 if the pixels highlighted by Greedy-AS are set to white. Similarly, a 1 may be turned to a 4 or a 7 given additional white pixels to its left, and a 9 may become a 7 if deleted the lower circular part of its head. From the results, we see that Greedy-AS focuses on *"the region where perturbation on its current value will lead to easier prediction change"*, which includes both the crucial positive and pertinent negative pixels. Such capability of Greedy-AS is also validated by its superior performance on the proposed robustness criteria, on which methods like LOO that highlights only the white strokes of digits show relatively low performance. The capability of capturing pertinent negative features has also been observed in explanations proposed in some recent work [10, 35, 118]. From the visualized ImageNet examples shown in Figure 4.3, we observe that our method provides more compact explanations that focus mainly on the actual objects being classified. For instance, in the first image, our method focuses more on the face of the Maltese while others tend to have noisier results; in the last image, our method focuses on one of the Japanese Spaniel whereas others highlight both the dogs and some noisy regions.

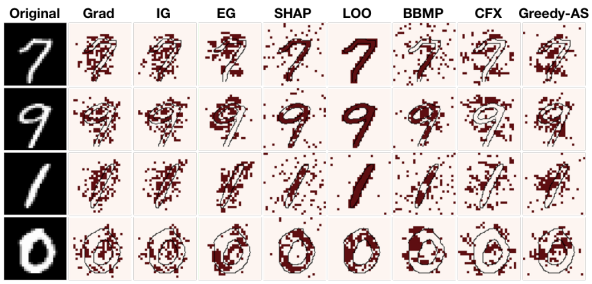


Figure 4.2: Visualization on top 20 percent relevant features provided by different explanations on MNIST. We see Greedy-AS highlights both crucial positive and pertinent negative features supporting the prediction.

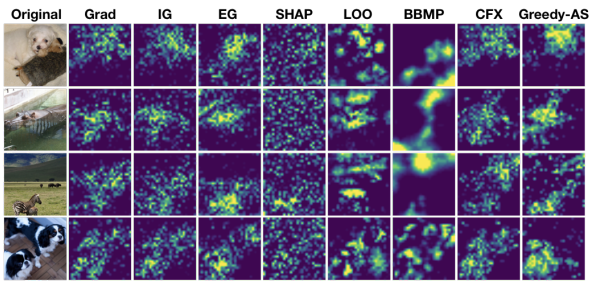


Figure 4.3: Visualization of different explanations on ImageNet, where the predicted class for each input is “Maltese", “hippopotamus", “zebra", and “Japanese Spaniel". Greedy-AS focuses more compactly on objects.

Input	Ronaldinho and kaka are my favorite players out there. why did they replace them? I completely missed that part. Do they say why the switched them?
Grad	Ronaldinho and kaka are my favorite players out there. why did they replace them? I completely missed that part. Do they say why the switched them?
IG	Ronaldinho and kaka are my favorite players out there. why did they replace them? I completely missed that part. Do they say why the switched them?
EG	Ronaldinho and kaka are my favorite players out there. why did they replace them? I completely missed that part. Do they say why the switched them?
SHAP	Ronaldinho and kaka are my favorite players out there. why did they replace them? I completely missed that part. Do they say why the switched them?
LOO	Ronaldinho and kaka are my favorite players out there. why did they replace them? I completely missed that part. Do they say why the switched them?
CFX	Ronaldinho and kaka are my favorite players out there. why did they replace them? I completely missed that part. Do they say why the switched them?
Greedy-AS	Ronaldinho and kaka are my favorite players out there. why did they replace them? I completely missed that part. Do they say why the switched them?
Anchor	Ronaldinho and kaka are my favorite players out there. why did they replace them? I completely missed that part. Do they say why the switched them?

Most Relevant — Less Relevant

Figure 4.4: Explanations on a text classification model which correctly predicts the label “sport". Unlike most other methods, the top-5 relevant keywords highlighted by Greedy-AS are all related to the concept “sport".

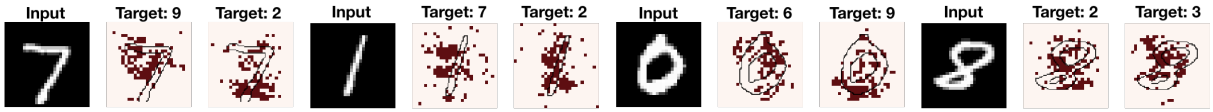


Figure 4.5: Visualization of targeted explanation. For each input, we highlight relevant regions explaining why the input is not predicted as the target class. We see the explanation changes in a semantically meaningful way as the target class changes.

Text Classification. Here we demonstrate how our explanation method could be applied to text classification models. We showcase an example in Figure 4.4. We see that the top-5 keywords highlighted by Greedy-AS are all relevant to the label “sport”, and Greedy-AS is less likely to select stop words as compared to other methods. Additionally, we conduct an user study where we observe that Greedy-AS generates explanation that matches user intuition the most.

Targeted Explanation Analysis. In section 4.1.2, we discussed about the possibility of using targeted adversarial perturbation to answer the question of *“why the input is predicted as A but not B”*. In Figure 4.5, for each input digit, we provide targeted explanation towards two different target classes. Interestingly, as the target class changes, the generated explanation varies in an interpretatble way. For example, in the first image, we explain why the input digit 7 is not classified

as a 9 (middle column) or a 2 (rightmost column). The resulting explanation against 9 highlights the upper-left part of the 7. Semantically, this region is indeed pertinent to the classification between 7 and 9, since turning on the highlighted pixel values in the region (currently black in the original image) will then make the 7 resemble a 9. However, the targeted explanation against 2 highlights a very different but also meaningful region, which is the lower-right part of the 7; since adding a horizontal stroke on the area would turn a 7 into a 2.

Chapter 5

Representer Point Framework

As machine learning systems start to be more widely used, we are starting to care not just about the accuracy and speed of the predictions, but also *why* it made its specific predictions. While we need not always care about the why of a complex system in order to trust it, especially if we observe that the system has high accuracy, such trust typically hinges on the belief that some other expert has a richer understanding of the system. For instance, while we might not know exactly how planes fly in the air, we trust some experts do. In the case of machine learning models however, even machine learning experts do not have a clear understanding of why say a deep neural network makes a particular prediction. Our work proposes to address this gap by focusing on improving the understanding of experts, in addition to lay users. In particular, expert users could then use these explanations to further fine-tune the system (e.g. dataset/model debugging), as well as suggest different approaches for model training, so that it achieves a better performance.

Our key approach to do so is via a representer theorem for deep neural networks, which might be of independent interest even outside the context of explainable ML. We show that we can decompose the pre-activation prediction values into a linear combination of training point activations, with the weights corresponding to what we call representer values, which can be used to measure the importance of each training point has on the learned parameter of the model. Using these representer values, we select representer points – training points that have large/small representer values – that could aid the understanding of the model's prediction.

Such representer points provide a richer understanding of the deep neural network than other approaches that provide influential training points, in part because of the meta-explanation underlying our explanation: a positive representer value indicates that a similarity to that training point is *excitatory*, while a negative representer value indicates that a similarity to that training point is *inhibitory*, to the prediction at the given test point. It is in these inhibitory training points where our approach provides considerably more insight compared to other approaches: specifically, what would cause the model to *not* make a particular prediction? In one of our examples, we see that the model makes an error in labeling an antelope as a deer. Looking at its most inhibitory training points, we see that the dataset is rife with training images where there are antelopes in the image, but also some other animals, and the image is labeled with the other animal. These thus contribute to inhibitory effects of small antelopes with other big objects: an insight that as machine learning experts, we found deeply useful, and which is difficult to obtain via other explanatory approaches. We demonstrate the utility of our class of *representer point*

explanations through a range of theoretical and empirical investigations.

5.1 Related Work

There are two main classes of approaches to explain the prediction of a model. The first class of approaches point to important input features. Ribeiro et al. [129] provide such feature-based explanations that are model-agnostic; explaining the decision locally around a test instance by fitting a local linear model in the region. Ribeiro et al. [130] introduce Anchors, which are locally sufficient conditions of features that "holds down" the prediction so that it does not change in a local neighborhood. Such feature based explanations are particularly natural in computer vision tasks, since it enables visualizing the regions of the input pixel space that causes the classifier to make certain predictions. There are numerous works along this line, particularly focusing on gradient-based methods that provide saliency maps in the pixel space [10, 142, 144, 156].

The second class of approaches are sample-based, and they identify training samples that have the most influence on the model's prediction on a test point. Among model-agnostic sample-based explanations are prototype selection methods [14, 86] that provide a set of "representative" samples chosen from the data set. Kim et al. [87] provide criticism alongside prototypes to explain what are not captured by prototypes. Usually such prototype and criticism selection is model-agnostic and used to accelerate the training for classifications. Model-aware sample-based explanation identify influential training samples which are the most helpful for reducing the objective loss or making the prediction. Recently, Koh and Liang [89] provide tractable approximations of influence functions that characterize the influence of each sample in terms of change in the loss. Anirudh et al. [6] propose a generic approach to influential sample selection via a graph constructed using the samples.

Our approach is based on a representer theorem for deep neural network predictions. Representer theorems [137] in machine learning contexts have focused on non-parametric regression, specifically in reproducing kernel Hilbert spaces (RKHS), and which loosely state that under certain conditions the minimizer of a loss functional over a RKHS can be expressed as a linear combination of kernel evaluations at training points. There have been recent efforts at leveraging such insights to compositional contexts [16, 166], though these largely focus on connections to non-parametric estimation. Bohn et al. [16] extend the representer theorem to compositions of kernels, while Unser [166] draws connections between deep neural networks to such deep kernel estimation, specifically deep spline estimation. In our work, we consider the much simpler problem of explaining pre-activation neural network predictions in terms of activations of training points, which while less illuminating from a non-parametric estimation standpoint, is arguably much more explanatory, and useful from an explainable ML standpoint.

5.2 Decomposing Testing Point Value by Training Data

5.2.1 Problem Setup

Consider a classification problem, of learning a mapping from an input space $\mathcal{X} \subseteq \mathbb{R}^d$ (e.g., images) to an output space $\mathcal{Y} \subseteq \mathbb{R}$ (e.g., labels), given training points $\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_n$, and corre-

sponding labels $\mathbf{y}_1, \mathbf{y}_2, ...\mathbf{y}_n$. We consider a neural network as our prediction model, which takes the form $\hat{\mathbf{y}}_i = \sigma(\mathbf{f}(\mathbf{x}_i, \Theta)) \subseteq \mathbb{R}^c$, where $\mathbf{f}(\mathbf{x}_i, \Theta) = \Theta_1 \mathbf{f}_i \subseteq \mathbb{R}^c$ and $\mathbf{f}_i = \mathbf{f}_2(\mathbf{x}_i, \Theta_2) \subseteq \mathbb{R}^f$ is the last intermediate layer feature in the neural network for input $\mathbf{x}_i$. Note that c is the number of classes, f is the dimension of the feature, Θ_1 is a matrix $\subseteq \mathbb{R}^{c \times f}$, and Θ_2 is all the parameters to generate the last intermediate layer from the input $\mathbf{x}_i$.

Thus $\Theta = \{\Theta_1, \Theta_2\}$ are all the parameters of our neural network model. The parameterization above connotes splitting of the model as a feature model $\mathbf{f}_2(\mathbf{x}_i, \Theta_2)$ and a prediction network with parameters Θ_1. Note that the feature model $\mathbf{f}_2(\mathbf{x}_i, \Theta_2)$ can be arbitrarily deep, or simply the identity function, so our setup above is applicable to general feed-forward networks.

5.2.2 Completeness for Training Data Importance: Decomposition of f(x) by Training Data Contribution

Our goal is to understand to what extent does one particular training point $\mathbf{x}_i$ affect the prediction $\hat{\mathbf{y}}_t$ of a test point $\mathbf{x}_t$ as well as the learned weight parameter Θ. Let $L(\mathbf{x}, \mathbf{y}, \Theta)$ be the loss, and $\frac{1}{n}\sum_i^n L(\mathbf{x}_i, \mathbf{y}_i, \Theta)$ be the empirical risk. To indicate the form of a representer theorem, suppose we solve for the optimal parameters $\Theta^* = \arg\min_\Theta \left\{ \frac{1}{n}\sum_i^n L(\mathbf{x}_i, \mathbf{y}_i, \Theta) + g(||\Theta||) \right\}$ for some non-decreasing g. We would then like our pre-activation predictions $\mathbf{f}(\mathbf{x}_t, \Theta)$ to have the decomposition: $\mathbf{f}(\mathbf{x}_t, \Theta^*) = \sum_i^n \alpha_i k(\mathbf{x}_t, \mathbf{x}_i)$. Given such a representer theorem, $\alpha_i k(\mathbf{x}_t, \mathbf{x}_i)$ can be seen as the contribution of the training data $\mathbf{x}_i$ on the testing prediction $\mathbf{f}(\mathbf{x}_t, \Theta)$.

5.2.3 A Representer Theorem that Satisfies the Decomposition

However, such representer theorems have only been developed for non-parametric predictors, specifically where $\mathbf{f}$ lies in a reproducing kernel Hilbert space. Moreover, unlike the typical RKHS setting, finding a global minimum for the empirical risk of a deep network is difficult, if not impossible, to obtain. In the following, we provide a representer theorem that addresses these two points: it holds for deep neural networks, and for any stationary point solution.

Theorem 10. *Let us denote the neural network prediction function by* $\hat{\mathbf{y}}_i = \sigma(\mathbf{f}(\mathbf{x}_i, \Theta))$*, where* $\mathbf{f}(\mathbf{x}_i, \Theta) = \Theta_1 \mathbf{f}_i$ *and* $\mathbf{f}_i = \mathbf{f}_2(\mathbf{x}_i, \Theta_2)$*. Suppose* Θ^* *is a stationary point of the optimization problem:* $\arg\min_\Theta \left\{ \frac{1}{n}\sum_i^n L(\mathbf{x}_i, \mathbf{y}_i, \Theta)) + g(||\Theta_1||) \right\}$*, where* $g(||\Theta_1||) = \lambda ||\Theta_1||^2$ *for some* $\lambda > 0$*. Then we have the decomposition:*

$$\mathbf{f}(\mathbf{x}_t, \Theta^*) = \sum_i^n k(\mathbf{x}_t, \mathbf{x}_i, \alpha_i),$$

where $\alpha_i = \frac{1}{-2\lambda n} \frac{\partial L(\mathbf{x}_i, \mathbf{y}_i, \Theta)}{\partial \mathbf{f}(\mathbf{x}_i, \Theta)}$ *and* $k(\mathbf{x}_t, \mathbf{x}_i, \alpha_i) = \alpha_i \mathbf{f}_i^T \mathbf{f}_t$*, which we call a representer value for* $\mathbf{x}_i$ *given* $\mathbf{x}_t$*.*

Proof. Note that for any stationary point, the gradient of the loss with respect to Θ_1 is equal to 0. We therefore have

$$\frac{1}{n}\sum_{i=1}^n \frac{\partial L(\mathbf{x}_i, \mathbf{y}_i, \Theta)}{\partial \Theta_1} + 2\lambda \Theta_1^* = 0 \quad \Rightarrow \quad \Theta_1^* = -\frac{1}{2\lambda n}\sum_{i=1}^n \frac{\partial L(\mathbf{x}_i, \mathbf{y}_i, \Theta)}{\partial \Theta_1} = \sum_{i=1}^n \alpha_i \mathbf{f}_i^T \qquad (5.1)$$

where $\alpha_i = -\frac{1}{2\lambda n}\frac{\partial L(\mathbf{x}_i,\mathbf{y}_i,\Theta)}{\partial \mathbf{f}(\mathbf{x}_i,\Theta)}$ by the chain rule. We thus have that

$$\mathbf{f}(\mathbf{x}_t,\Theta^*) = \Theta_1^*\mathbf{f}_t = \sum_{i=1}^{n} k(\mathbf{x}_t,\mathbf{x}_i,\alpha_i), \tag{5.2}$$

where $k(\mathbf{x}_t,\mathbf{x}_i,\alpha_i) = \alpha_i \mathbf{f}_i^T\mathbf{f}_t$ by simply plugging in the expression (5.1) into (5.2). ■

We note that α_i can be seen as the resistance for training example feature $\mathbf{f}_i$ towards minimizing the norm of the weight matrix Θ_1. Therefore, α_i can be used to evaluate the importance of the training data $\mathbf{x}_i$ have on Θ_1. Note that for any class j, $\mathbf{f}(\mathbf{x}_t,\Theta^*)_j = \Theta_{1j}^*\mathbf{f}_t = \sum_{i=1}^{n} k(\mathbf{x}_t,\mathbf{x}_i,\alpha_i)_j$ holds by (5.2). Moreover, we can observe that for $k(\mathbf{x}_t,\mathbf{x}_i,\alpha_i)_j$ to have a significant value, two conditions must be satisfied: (a) α_{ij} should have a large value, and (b) $\mathbf{f}_i^T\mathbf{f}_t$ should have a large value. Therefore, we interpret the pre-activation value $\mathbf{f}(\mathbf{x}_t,\Theta)_j$ as a weighted sum for the feature similarity $\mathbf{f}_i^T\mathbf{f}_t$ with the weight α_{ij}. When $\mathbf{f}_t$ is close to $\mathbf{f}_i$ with a large positive weight α_{ij}, the prediction score for class j is increased. On the other hand, when $\mathbf{f}_t$ is close to $\mathbf{f}_i$ with a large negative weight α_{ij}, the prediction score for class j is then decreased.

We can thus interpret the training points with negative representer values as inhibitory points that suppress the activation value, and those with positive representer values as excitatory examples that does the opposite. We demonstrate this notion with examples further in Section 5.3.2. We note that such excitatory and inhibitory points provide a richer understanding of the behavior of the neural network: it provides insight both as to why the neural network prefers a particular prediction, as well as *why it does not*, which is typically difficult to obtain via other sample-based explanations.

5.2.4 Setting 1: Training an Interpretable Model by Imposing L2 Regularization.

Theorem 10 works for any model that performs a linear matrix multiplication before the activation σ, which is quite general and can be applied on most neural-network-like structures. By simply introducing a L2 regularizer on the weight with a fixed $\lambda > 0$, we can easily decompose the pre-softmax prediction value as some finite linear combinations of a function between the test and train data. We now state our main algorithm. First we solve the following optimization problem:

$$\Theta^* = \arg\min_{\Theta} \frac{1}{n}\sum_{i}^{n} L(\mathbf{y}_i,\mathbf{f}(\mathbf{x}_i,\Theta)) + \lambda||\Theta_1||^2. \tag{5.3}$$

Note that for the representer point selection to work, we would need to achieve a stationary point with high precision. In practice, we find that using a gradient descent solver with line search or LBFGS solver to fine-tune after converging in SGD can achieve highly accurate stationary point. Note that we can perform the fine-tuning step only on Θ_1, which is usually efficient to compute. We can then decompose $\mathbf{f}(\mathbf{x}_t,\Theta) = \sum_i^n k(\mathbf{x}_t,\mathbf{x}_i,\alpha_i)$ by Theorem 10 for any arbitrary test point $\mathbf{x}_t$, where $k(\mathbf{x}_t,\mathbf{x}_i,\alpha_i)$ is the contribution of training point $\mathbf{x}_i$ on the pre-softmax prediction $\mathbf{f}(\mathbf{x}_t,\Theta)$. We emphasize that imposing L2 weight decay is a common practice to avoid overfitting for deep neural networks, which does not sacrifice accuracy while achieving a more interpretable model.

5.2.5 Setting 2: Generating Representer Points for a Given Pre-trained Model.

We are also interested in finding representer points for a given model $\Phi(\Theta_{given})$ that has already been trained, potentially without imposing the L2 regularizer. While it is possible to add the L2 regularizer and retrain the model, the retrained model may converge to a different stationary point, and behave differently compared to the given model, in which case we cannot use the resulting representer points as explanations. Accordingly, we learn the parameters Θ while imposing the L2 regularizer, but under the additional constraint that $\Phi(\mathbf{x}_i, \Theta)$ be close to $\Phi(\mathbf{x}_i, \Theta_{given})$. In this case, our learning objective becomes $\mathbf{f}(\mathbf{x}_i, \Theta_{given})$ instead of y_i, and our loss $L(\mathbf{x}_i, y_i, \Theta)$ can be written as $L(\mathbf{f}(\mathbf{x}_i, \Theta_{given}), \mathbf{f}(\mathbf{x}_i, \Theta))$.

Definition 7. We say that a convex loss function $L(\mathbf{f}(\mathbf{x}_i, \Theta_{given}), \mathbf{f}(\mathbf{x}_i, \Theta))$ is "suitable" to an activation function σ, if it holds that for any $\Theta^* \in \arg\min_\Theta L(\mathbf{f}(\mathbf{x}_i, \Theta_{given}), \mathbf{f}(\mathbf{x}_i, \Theta))$, we have $\sigma(\Phi(\mathbf{x}_i, \Theta^*)) = \sigma(\Phi(\mathbf{x}_i, \Theta_{given}))$.

Assume that we are given such a loss function L that is "suitable to" the activation function σ. We can then solve the following optimization problem:

$$\Theta^* \in \arg\min_\Theta \left\{ \frac{1}{n} \sum_i^n L(\mathbf{f}(\mathbf{x}_i, \Theta_{given}), \mathbf{f}(\mathbf{x}_i, \Theta)) + \lambda ||\Theta_1||^2 \right\}. \tag{5.4}$$

The optimization problem can be seen to be convex under the assumptions on the loss function. The parameter $\lambda > 0$ controls the trade-off between the closeness of $\sigma(\Phi(\mathbf{X}, \Theta))$ and $\sigma(\Phi(\mathbf{X}, \Theta_{given}))$, and the computational cost. For a small λ, $\sigma(\Phi(\mathbf{X}, \Theta))$ could be arbitrarily close to $\sigma(\Phi(\mathbf{X}, \Theta_{given}))$, while the convergence time may be long. We note that the learning task in Eq. (5.4) can be seen as learning from a teacher network Θ_{given} and imposing a regularizer to make the student model Θ capable of generating representer points. In practice, we may take Θ_{given} as an initialization for Θ and perform a simple line-search gradient descent with respect to Θ_1 in (5.4). In our experiments, we discover that the training for (5.4) can converge to a stationary point in a short period of time, as demonstrated in Section 5.3.5.

We now discuss our design for the loss function that is mentioned in (5.4). When σ is the softmax activation, we choose the softmax cross-entropy loss, which computes the cross entropy between $\sigma(\Phi(\mathbf{x}_i, \Theta_{given}))$ and $\sigma(\Phi(\mathbf{x}_i, \Theta))$ for $L_{\text{softmax}}(\mathbf{f}(\mathbf{x}_i, \Theta_{given}), \mathbf{f}(\mathbf{x}_i, \Theta))$. When σ is ReLU activation, we choose $L_{\text{ReLU}}(\mathbf{f}(\mathbf{x}_i, \Theta_{given}), \mathbf{f}(\mathbf{x}_i, \Theta)) = \frac{1}{2} \max(\mathbf{f}(\mathbf{x}_i, \Theta), 0) \odot \mathbf{f}(\mathbf{x}_i, \Theta) - \max(\mathbf{f}(\mathbf{x}_i, \Theta_{given}), 0) \odot \mathbf{f}(\mathbf{x}_i, \Theta)$, where $\odot$ is the element-wise product. In the following Proposition, we show that L_{softmax} and L_{ReLU} are convex, and satisfy the desired suitability property in Definition 7. The proof is provided in the supplementary material.

Proposition 6. *The loss functions $L_{softmax}$ and L_{ReLU} are both convex in Θ_1. Moreover, $L_{softmax}$ is "suitable to" the softmax activation, and L_{ReLU} is "suitable to" the ReLU activation, following Definition 7.*

As a sanity check, we perform experiments on the CIFAR-10 dataset [93] with a pre-trained VGG-16 network [143]. We first solve (5.4) with loss $L_{\text{softmax}}(\mathbf{f}(\mathbf{x}_i, \Theta), \mathbf{f}(\mathbf{x}_i, \Theta_{given}))$ for $\lambda = 0.001$, and then calculate $\mathbf{f}(\mathbf{x}_t, \Theta^*) = \sum_{i=1}^n k(\mathbf{x}_t, \mathbf{x}_i, \alpha_i)$ as in (5.2) for all train and test points. We note that the computation time for the whole procedure only takes less than a minute, given the pre-trained model. We compute the Pearson correlation coefficient between the actual output

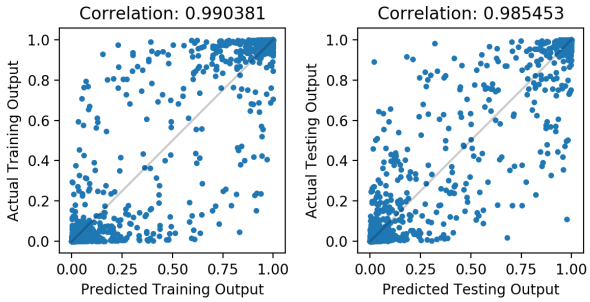


Figure 5.1: Pearson correlation between the actual and approximated softmax output (expressed as a linear combination) for train (left) and test (right) data in CIFAR-10 dataset. The correlation is almost 1 for both cases.

$\sigma(\Phi(\mathbf{x}_t, \Theta))$ and the predicted output $\sigma(\sum_{i=1}^{n} k(\mathbf{x}_t, \mathbf{x}_i, \alpha_i))$ for multiple points and plot them in Figure 5.1. The correlation is almost 1 for both train and test data, and most points lie at the both ends of $y = x$ line.

We note that Theorem 10 can be applied to any hidden layer with ReLU activation by defining a sub-network from input $\mathbf{x}$ and the output being the hidden layer of interest. The training could be done in a similar fashion by replacing L_{softmax} with L_{ReLU}. In general, any activation can be used with a derived "suitable loss".

5.3 Experiments

We perform a number of experiments with multiple datasets and evaluate our method's performance and compare with that of the influence functions.[1] The goal of these experiments is to demonstrate that selecting the representer points is efficient and insightful in several ways. Additional experiments discussing the differences between our method and the influence function are included in the supplementary material.

5.3.1 Dataset Debugging

To evaluate the influence of the samples, we consider a scenario where humans need to inspect the dataset quality to ensure an improvement of the model's performance in the test data. Real-world data is bound to be noisy, and the bigger the dataset becomes, the more difficult it will be for humans to look for and fix mislabeled data points. It is crucial to know which data points are more important than the others to the model so that prioritizing the inspection can facilitate the debugging process.

To show how well our method does in dataset debugging, we run a simulated experiment on CIFAR-10 dataset [97] with a task of binary classification with logistic regression for the classes

[1] Source code available at `github.com/chihkuanyeh/Representer_Point_Selection`.

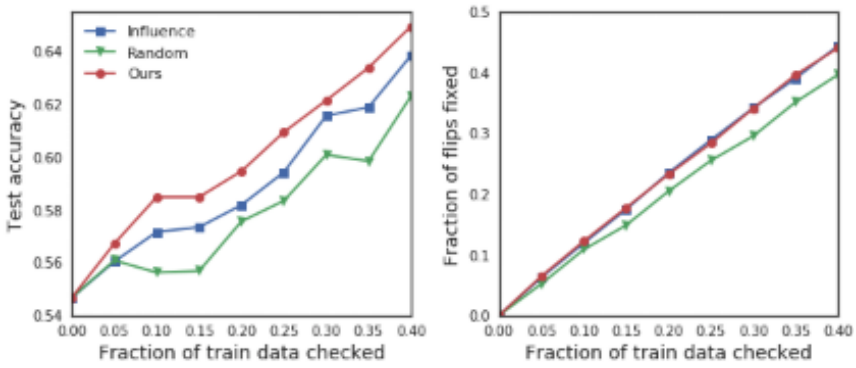


Figure 5.2: Dataset debugging performance for several methods. By inspecting the training points using the representer value, we are able to recover the same amount of mislabeled training points as the influence function (right) with the highest test accuracy compared to other methods (left).

automobiles and horses. The dataset is initially corrupted, where 40 percent of the data has the labels flipped, which naturally results in a low test accuracy of 0.55. The simulated user will check some fraction of the train data based on the order set by several metrics including ours, and fix the labels. With the corrected version of the dataset, we retrain the model and record the test accuracies for each metrics. For our method, we train an explainable model by mimimizing (5.3) as explained in section 5.2.4. The L2 weight decay is set to $1e^{-2}$ for all methods for fair comparison. All experiments are repeated for 5 random splits and we report the average result. In Figure 5.2 we report the results for four different metrics: "ours" picks the points with bigger $|\alpha_{ij}|$ for training instance i and its corresponding label j; "influence" prioritizes the training points with bigger influence function value; and "random" picks random points. We observe that our method recovers the same amount of training data as the influence function while achieving higher testing accuracy. Nevertheless, both methods perform better than the random selection method.

5.3.2 Excitatory (Positive) and Inhibitory (Negative) Examples

We visualize the training points with high representer values (both positive and negative) for some test points in Animals with Attributes (AwA) dataset [175] and compare the results with those of the influence functions. We use a pre-trained Resnet-50 [72] model and fine-tune on the AwA dataset to reach over 90 percent testing accuracy. We then generate representer points as described in section 5.2.5. For computing the influence functions, just as described in [89], we froze all top layers of the model and trained the last layer. We report top three points for two test points in the following Figures 5.3 and 5.4. In Figure 5.3, which is an image of three grizzly bears, our method correctly returns three images that are in the same class with similar looks, similar to the results from the influence function. The positive examples excite the activation values for a particular class and supports the decision the model is making. For the negative examples, just like the influence functions, our method returns images that look like the test image but are labeled as a different class. In Figure 5.4, for the image of a rhino the influence function could not recover useful training points, while ours does, including the similar-looking elephants or zebras which

might be confused as rhinos, as negatives. The negative examples work as inhibitory examples for the model – they suppress the activation values for a particular class of a given test point because they are in a different class despite their striking similarity to the test image. Such inhibitory points thus provide a richer understanding, even to machine learning experts, of the behavior of deep neural networks, since they explicitly indicate training points that lead the network away from a particular label for the given test point. More examples can be found in the supplementary material.

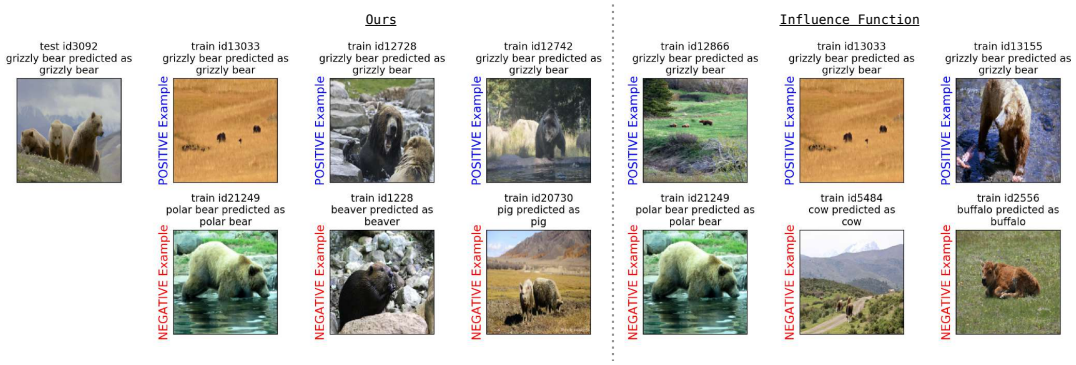


Figure 5.3: Comparison of top three positive and negative influential training images for a test point (left-most column) using our method (left columns) and influence functions (right columns).

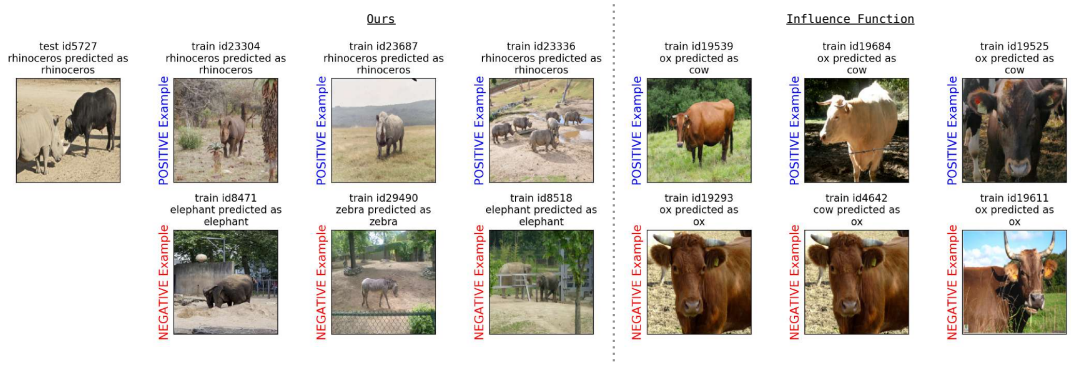


Figure 5.4: Here we can observe that our method provides clearer positive and negative examples while the influence function fails to do so.

5.3.3 Understanding Misclassified Examples

The representer values can be used to understand the model's mistake on a test image. Consider a test image of an antelope predicted as a deer in the left-most panel of Figure 5.5. Among 181 test images of antelopes, the total number of misclassified instances is 15, among which 12 are misclassified as deer. All of those 12 test images of antelopes had the four training images shown in Figure 5.5 among the top inhibitory examples. Notice that we can spot antelopes even in the images labeled as zebra or elephant. Such noise in the labels of the training data confuses the

model – while the model sees elephant *and* antelope, the label forces the model to focus on just the elephant. The model thus learns to inhibit the antelope class given an image with small antelopes and other large objects. This insight suggests for instance that we use multi-label prediction to train the network, or perhaps clean the dataset to remove such training examples that would be confusing to humans as well. Interestingly, the model makes the same mistake (predicting deer instead of antelope) on the second training image shown (third from the left of Figure 5.5), and this suggests that for the training points, we should expect most of the misclassifications to be deer as well. And indeed, among 863 training images of antelopes, 8 are misclassified, and among them 6 are misclassified as deer.

Figure 5.5: A misclassified test image (left) and the set of four training images that had the most negative representer values for almost all test images in which the model made the same mistakes. The negative influential images all have antelopes in the image despite the label being a different animal.

5.3.4 Sensitivity Map Decomposition

From Theorem 10, we have seen that the pre-softmax output of the neural network can be decomposed as the weighted sum of the product of the training point feature and the test point feature, or $\mathbf{f}(\mathbf{x}_t, \Theta^*) = \sum_i^n \alpha_i \mathbf{f}_i^T \mathbf{f}_t$. If we take the gradient with respect to the test input $\mathbf{x}_t$ for both sides, we get $\frac{\partial \mathbf{f}(\mathbf{x}_t, \Theta^*)}{\partial \mathbf{x}_t} = \sum_i^n \alpha_i \frac{\partial \mathbf{f}_i^T \mathbf{f}_t}{\partial \mathbf{x}_t}$. Notice that the LHS is the widely-used notion of sensitivity map (gradient-based attribution), and the RHS suggests that we can decompose this sensitivity map into a weighted sum of sensitivity maps that are native to each i-th training point. This gives us insight into how sensitivities of training points contribute to the sensitivity of the given test image.

In Figure 5.6, we demonstrate two such examples, one from the class zebra and one from the class moose from the AwA dataset. The first column shows the test images whose sensitivity maps we wish to decompose. For each example, in the following columns we show top four influential representer points in the the top row, and visualize the decomposed sensitivity maps in the bottom. We used SmoothGrad [148] to obtain the sensitivity maps.

For the first example of a zebra, the sensitivity map on the test image mainly focuses on the face of the zebra. This means that infinitesimally changing the pixels around the face of the zebra would cause the greatest change in the neuron output. Notice that the focus on the head of the zebra is distinctively the strongest in the fourth representer point (last column) when the training image manifests clearer facial features compared to other training points. For the rest of the training images that are less demonstrative of the facial features, the decomposed sensitivity maps

accordingly show relatively higher focus on the background than on the face. For the second example of a moose, a similar trend can be observed – when the training image exhibits more distinctive bodily features of the moose than the background (first, second, third representer points), the decomposed sensitivity map highlights the portion of the moose on the test image more compared to training images with more features of the background (last representer point). This provides critical insight into the contribution of the representer points towards the neuron output that might not be obvious just from looking at the images itself.

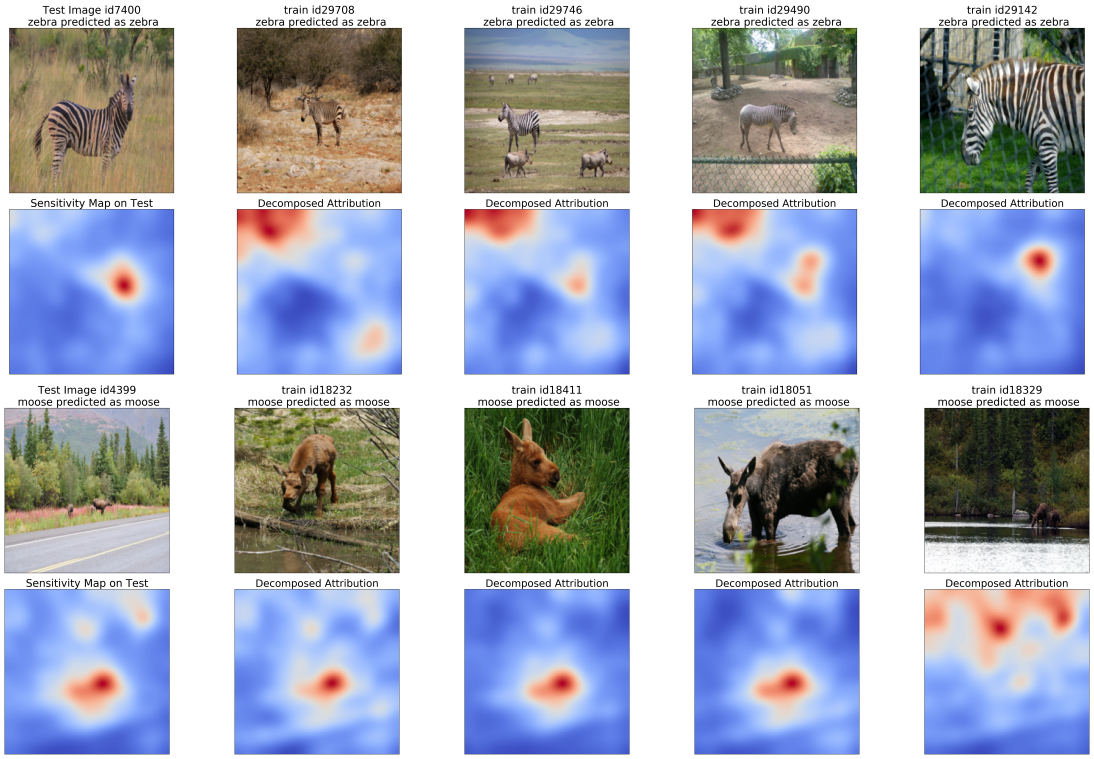


Figure 5.6: Sensitivity map decomposition using representer points, for the class zebra (above two rows) and moose (bottom two rows). The sensitivity map on the test image in the first column can be readily seen as the weighted sum of the sensitivity maps for each training point. The less the training point displays spurious features from the background and more of the features related to the object of interest, the more focused the decomposed sensitivity map corresponding to the training point is at the region the test sensitivity map mainly focuses on.

5.3.5 Computational Cost and Numerical Instabilities

Computation time is particularly an issue for computing the influence function values [89] for a large dataset, which is very costly to compute for each test point. We randomly selected a subset of test points, and report the comparison of the computation time in Table 5.1 measured on CIFAR-10 and AwA datasets. We randomly select 50 test points to compute the values for all train data, and recorded the average and standard deviation of computation time. Note that the influence function does not need the fine-tuning step when given a pre-trained model, hence the values being

0, while our method first optimizes for Θ^* using line-search then computes the representer values. However, note that the fine-tuning step is a one time cost, while the computation time is spent for every testing image we analyze. Our method significantly outperforms the influence function, and such advantage will favor our method when a larger number of data points is involved. In particular, our approach could be used for *real-time explanations* of test points, which might be difficult with the influence function approach.

	Influence Function		Ours	
Dataset	Fine-tuning	Computation	Fine-tuning	Computation
CIFAR-10	0	267.08 ± 248.20	7.09 ± 0.76	0.10 ± 0.08
AwA	0	172.71 ± 32.63	12.41 ± 2.37	0.19 ± 0.12

Table 5.1: Time required for computing an influence function / representer value for all training points and a test point in seconds. The computation of Hessian Vector Products for influence function alone took longer than our combined computation time.

While ranking the training points according to their influence function values, we have observed numerical instabilities, more discussed in the supplementary material. For CIFAR-10, over 30 percent of the test images had all zero training point influences, so influence function was unable to provide positive or negative influential examples. The distribution of the values is demonstrated in Figure 5.7, where we plot the histogram of the maximum of the absolute values for each test point in CIFAR-10. Notice that over 300 testing points out of 1,000 lie in the first bin for the influence functions (right). We checked that all data in the first bin had the exact value of 0. Roughly more than 200 points lie in range $[10^{-40}, 10^{-28}]$, the values which may create numerical instabilities in computations. On the other hand, our method (left) returns non-trivial and more numerically stable values across all test points.

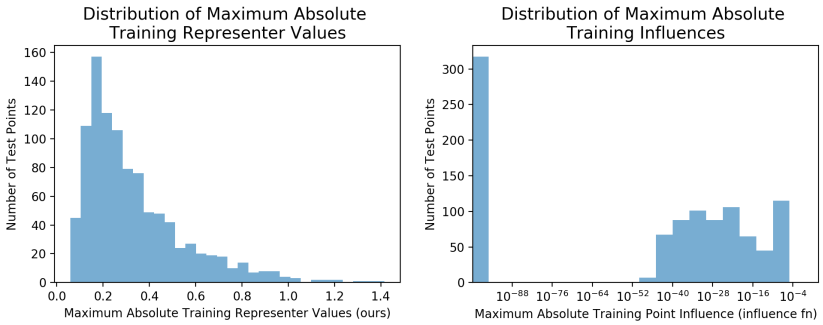


Figure 5.7: The distribution of influence/representer values for a set of randomly selected 1,000 test points in CIFAR-10. While ours have more evenly spread out larger values across different test points (left), the influence function values can be either really small or become zero for some points, as seen in the left-most bin (right).

Chapter 6

Completeness of Concepts – How Sufficient are Concepts to Explain a Model

The lack of explainability of deep neural networks (DNNs) arguably hampers their full potential for real-world impact. Explanations can help domain experts better understand rationales behind the model decisions, identify systematic failure cases, and potentially provide feedback to model builders for improvements. Most commonly-used methods for DNNs explain each prediction by quantifying the importance of each input feature [104, 129]. One caveat with such explanations is that they typically focus on the local behavior for each data point, rather than globally explaining how the model reasons. Besides, the weighted input features are not necessarily the most intuitive explanations for human understanding, particularly when using low-level features such as raw pixel values. In contrast, human reasoning often comprise "concept-based thinking," extracting similarities from numerous examples and grouping them systematically based on their resemblance [8, 160]. It is thus of interest to develop such "concept-based explanations" to characterize the global behavior of a DNN in a way understandable to humans, explaining how DNNs use concepts in arriving at particular decisions.

A few recent studies have focused on bringing such concept-based explainability to DNNs, largely based on the common implicit assumption that the concepts lie in low-dimensional subspaces of some intermediate DNN activations. Via supervised training based on labeled concepts, TCAV [88] trains linear concept classifiers to derive concept vectors, and uses how sensitive predictions are to these vectors (via directional derivatives) to measure the importance of a concept with respect to a specific class. Zhou et al. [192] considers the decomposition of model predictions in terms of projections onto concept vectors. Instead of human-labeled concept data, Ghorbani et al. [54] employs k-means clustering of super-pixel segmentations of images to discover concepts. Bouchacourt and Denoyer [17] proposes a Bayesian generative model involving concept vectors. One drawback of these approaches is that they do not take into account *how much* each concept plays a role in the prediction. In particular, selecting a set of concepts salient to a particular class does not guarantee that these concepts are sufficient in explaining the prediction. The notion of sufficiency is also referred to as "completeness" of explanations, as in [55, 177]. This motivates the following key questions: Is there an unsupervised approach to extract concepts that are sufficiently predictive of a DNN's decisions? If so, how can we measure this sufficiency?

In this paper, we propose such a completeness score for concept-based explanations. Our metric can be applied to a set of concept vectors that lie in a subspace of some intermediate DNN activations, which is a general assumption in previous work in this context [88, 192]. Intuitively speaking, a set of "complete" concepts can fully explain the prediction of the underlying model. By further assuming that for a complete set of concepts, the projections of activations onto the concepts are a sufficient statistic for the prediction of the model, we may measure the "completion" of the concepts by the accuracy of the model just given these concept based sufficient statistics. For concept discovery, we propose a novel algorithm, which could also be viewed as optimizing a surrogate likelihood of the concept-based data generation process, motivated by topic modeling [15]. To ensure that the discovered complete concepts are also coherent (distinct from other concepts) and semantically meaningful, we further introduce an interpretability regularizer.

Beyond concept discovery, we also propose a score, *ConceptSHAP*, for quantification of concept attributions as contextualized importance. ConceptSHAP uniquely satisfies a key set of axioms involving the contribution of each concept to the completeness score [104, 140]. We also propose a class-specific version of ConceptSHAP that decomposes it with respect to each class in multi-class classification. This can be used to find class-specific concepts that contribute the most to a specific class. To verify the effectiveness of our *automated* completeness-aware concept discovery method, we create a synthetic dataset with apriori-known ground truth concepts. We show that our approach outperforms all compared methods in correct retrieval of the concepts as well as in terms of its coherency via a user study. Lastly, we demonstrate how our concept discovery algorithm provides additional insights into the behavior of DNN models on both image and language real-world datasets.

6.1 Defining Completeness of Concepts

Problem setting: Consider a set of n training examples $\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^n$, corresponding labels $y^1, y^2, ..., y^n$ and a given pre-trained DNN model that predicts the corresponding y from the input $\mathbf{x}$. We assume that the pre-trained DNN model can be decomposed into two functions: the first part $\Phi(\cdot)$ maps input $\mathbf{x}^i$ into an intermediate layer $\Phi(\mathbf{x}^i)$, and the second part $h(\cdot)$ maps the intermediate layer $\Phi(\mathbf{x}^i)$ to the output $h(\Phi(\mathbf{x}^i))$, which is a probability vector for each class, and $h_y(\Phi(\mathbf{x}))$ is the probability of data $\mathbf{x}$ being predicted as label y by the model p. For DNNs that build up by processing parts of input at a time, such as those composed of convolutional layers, we can additionally assume that $\Phi(\mathbf{x}^i)$ is the concatenation of $[\phi(\mathbf{x}^i_1), ..., \phi(\mathbf{x}^i_T)]$, such that $\Phi(\cdot) \in \mathbb{R}^{(T \cdot d)}$, and $\phi(\cdot) \in \mathbb{R}^d$. Here, $\mathbf{x}^i_1, \mathbf{x}^i_2, ..., \mathbf{x}^i_T$ denote different, potentially overlapping parts of the input for $\mathbf{x}^i$, such as a segment of an image or a sub-sentence of a text. These parts for example, can be chosen to correspond to the receptive field of the neurons at the intermediate layer $\Phi(\cdot)$. We will use these $\mathbf{x}^i_t$ to relate discovered concepts. As an illustration of such parts, consider the fifth convolution layer of a VGG-16 network with input shape 224×224 have the size $7 \times 7 \times 512$. If we treat this layer as $\Phi(\mathbf{x}^i)$, $\phi(\mathbf{x}^i_1)$ corresponds to the first 512 dimensions of the intermediate layer (with size $7 \times 7 \times 512$), and $\Phi(\mathbf{x}^i) = [\phi(\mathbf{x}^i_1), ..., \phi(\mathbf{x}^i_{49})]$. Here, each $\mathbf{x}^i_j$ corresponds to a 164×164 square in the input image (with effective stride 16), which is the receptive field of convolution layer 5 of VGG-16 [7]. We note that when the receptive field of $\phi(\cdot)$ is equal to the entire input size, such as for multi-layer perceptrons, we may simply choose

$T = 1$ so that $\mathbf{x}^i_{1:T} = \mathbf{x}^i$ and $\Phi(\mathbf{x}^i) = \phi(\mathbf{x}^i_1)$. Thus, our method can also be generally applied to any DNN with an arbitrary structure besides convolutional layers. To choose the intermediate layer to apply concepts, we follow previous works on concept explanations [54, 88] by starting from the layer closest to the prediction until we reached a layer that user is happy with, as higher layers encodes more abstract concepts with larger receptive field, and lower layers encodes more specific concepts with smaller receptive field.

Suppose that there is a set of m concepts denoted by unit vectors[1] $\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_m$ that represent linear directions in the activation space $\mathbf{f}(\cdot) \in \mathbb{R}^d$, given by a concept discovery algorithm. For each part of data point $\mathbf{x}_t$ (We omit i for notational simplicity), The inner product between the data and concept vector is viewed as the closeness of the input $\mathbf{x}_t$ and the concept $\mathbf{c}$ following [54, 88]. If $\langle \phi(\mathbf{x}_t), c_j \rangle$ is large, then we know that $\mathbf{x}_t$ is close to concept j. However, when $\langle \phi(\mathbf{x}_t), c_j \rangle$ is less than some threshold, the dot product value is not semantically meaningful other than the the input is not close to the concept. Based on this motivation, we define the *concept product* for part of data $\mathbf{x}_t$ as $v_\mathbf{c}(\mathbf{x}_t) := \text{TH}(\langle \phi(\mathbf{x}_t), c_j \rangle, \beta)_{j=1}^m \in \mathbb{R}^m$, where TH is a threshold which trims value less than β to 0. We normalize the concept product to unit norm for numerical stability, and aggregate upon all parts of data to obtain the *concept score* for input $\mathbf{x}$ as $v_\mathbf{c}(\mathbf{x}) = (\frac{v_c(x_t)}{\|v_c(x_t)\|_2})_{t=1}^T \in \mathbb{R}^{T \cdot m}$.

We assume that for "sufficient" concepts, the concept scores should be sufficient statistics for the model output, and thus we may evaluate the completeness of concepts by how well we can recover the prediction given the concept score. Let $g : \mathbb{R}^{T \cdot m} \to \mathbb{R}^{T \cdot d}$ denote any mapping from the concept score to the activation space of $\Phi(\cdot)$. If concept scores $v_\mathbf{c}(\cdot)$ are sufficient statistics for the model output, then there exists some mapping g_p such that $h(g_\mathsf{p}(v_\mathbf{c}(\mathbf{x}))) \approx \mathsf{p}(\mathbf{x})$. We can now formally define the completeness core for a set of concept vectors $\mathbf{c}_1, ..., \mathbf{c}_m$:

Definition 8. Completeness Score: Given a prediction model $\mathsf{p}(\mathbf{x}) = h(\mathbf{f}(\mathbf{x}))$, a set of concept vectors $\mathbf{c}_1, ..., \mathbf{c}_m$, we define the completeness score $\eta_\mathsf{p}(\mathbf{c}_1, ..., \mathbf{c}_m)$ as:

$$\eta_\mathsf{p}(\mathbf{c}_1, ..., \mathbf{c}_m) = \frac{\sup_g \mathbb{P}_{\mathbf{x},y \sim V}[y = \arg\max_{y'} h_{y'}(g(v_\mathbf{c}(\mathbf{x})))] - a_r}{\mathbb{P}_{\mathbf{x},y \sim V}[y = \arg\max_{y'} \mathsf{p}_{y'}(\mathbf{x})] - a_r}, \tag{6.1}$$

where V is the set of validation data and $\sup_g \mathbb{P}_{\mathbf{x},y \sim V}[y = \arg\max_{y'} h_{y'}(g(v_\mathbf{c}(\mathbf{x})))]$ is the best accuracy by predicting the label just given the concept scores $v_\mathbf{c}(\mathbf{x})$, and a_r is the accuracy of random prediction to equate the lower bound of completeness score to 0. When the target y is multi-label, we may generalize the definition of completeness score by replacing the accuracy with the binary accuracy, which is the accuracy where each label is treated as a binary classification.

To calculate the completeness score, we can set g to be a DNN or a simple linear projection, and optimize using stochastic gradient descent. In our experiments, we simply set g to be a two-layer perceptron with 500 hidden units. We note that we approximate $\mathsf{p}(\mathbf{x}_t)$ by $h(g_\mathsf{p}(v_\mathbf{c}(\mathbf{x}_t)))$, but not an arbitrary neural network $h_g(v_\mathbf{c}(\mathbf{x}_t))$ for two benefits: (a) the measure of completeness considers the architecture and parameter of the given model to be explained (b) the computation is much more efficient since we only need to optimize the parameters of g, instead of the whole backbone h_g. The completeness score measures how "sufficient" are the concept scores as a sufficient statistic of the model, based on the assumption that the concept scores of "complete" concepts are sufficient statistics of the model prediction $\mathsf{p}(\cdot)$. By measuring the accuracy achieved

[1] We apply additional normalization to $\phi(\cdot)$ so it has unit norm and keep the notation for simplicity.

by the concept score, we are effectively measuring how "complete" the concepts are. We note that the completeness score can also be used to measure how sufficient concepts can explain a dataset independent of the model, by replacing $\phi(\cdot), h(\cdot)$ with identical functions, and $\mathsf{p}(\mathbf{x})$ with y. Below is an illustrative example on why we need the completeness score:

Example 1. Consider a simplified scenario where we have the input $\mathbf{x} \in \mathbb{R}^m$, and the intermediate layer Φ is the identity function. In this case, the m concepts $\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_m$ are the one-hot encoding of each feature in $\mathbf{x}$. Assume that the concepts $\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_m$ follow independent Bernoulli distribution with $p = 0.5$, and the model we attempt to explain is $f(\mathbf{x}) = \mathbf{c}_1$ XOR $\mathbf{c}_2$... XOR $\mathbf{c}_m$. The ground truth concepts that are sufficient to the model prediction should then be $\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_m$. However, if we have the information on $\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_{m-1}$ but do not have information on $\mathbf{c}_m$, we may have at most 0.5 probability to predict the output of the model, which is the same as the accuracy of random guess. In this case, $\eta_{\mathsf{p}}(\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_{m-1}) = 0$. On the other hand, given $\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_m$, $\eta_{\mathsf{p}}(\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_m) = 1$.

The completeness score offers a way to assess the 'sufficiency' of the discovered concepts to "explain" reasoning behind a model's decision. Not only the completeness score is useful in evaluating a proposed concept discovery method, but it can also shed light on how much of the learned information by DNN may not be 'understandable' to humans. For example, if the completeness score is very high, but discovered concepts aren't making cohesive sense to humans, this may mean that the DNN is basing its decisions on other concepts that are potentially hard to explain.

6.2 Discovering Completeness-aware Interpretable Concepts

6.2.1 Limitations of existing methods

Our goal is to discover a set of maximally-complete concepts under the definition 8, where each concept is interpretable and semantically-meaningful to humans. We first discuss the limitations of recent notable works related to concept discovery and then explain how we address them. TCAV and ACE are concept discovery methods that use training data for specific concepts and use trained linear concept classifier to derive concept vectors. They quantify the saliency of a concept to a class using 'TCAV score', based on the similarity of the loss gradients to the concept vectors. This score implicitly assumes a first-order relationship between the concepts and the model outputs. Regarding labeling of the concepts, TCAV relies on human-defined labels, while ACE uses automatically-derived image clusters by k-means clustering of super-pixel segmentations. There are two main caveats to these approaches. The first is that while they may retrieve an important set of concepts, there is no guarantee on how 'complete' the concepts are in explain the model – e.g., one may have 10 concepts with high TCAV scores, but they may still be very insufficient in understanding the predictions. Besides, human-suggested exogenous concept data might even encode confirmation bias. The second caveat is that their saliency scores may fail to capture concepts that have non-linear relationships with the output due to first-order assumption. The concepts in Example 1 might not be retrieved by the TCAV score since XOR is not a linear relationship. Overall, our completeness score complements previous works in concept discovery by adding a criterion to determine whether a set of concepts are sufficient to explain the model. The discussion of our relation to PCA is in the Appendix.

6.2.2 Our method

The goal of our method is to obtain concepts that are *complete* to the model. We consider the case where each data point $\mathbf{x}^i$ has parts $\mathbf{x}^i_{1:T}$, as described above. We assume that input data has spatial dependency, which can help learning coherent concepts. Thus, we encourage proximity between each concept and its nearest neighbors patches. Note that the assumption works well with images and language, as we will demonstrate in the result section. We aim that the concepts would obtain consistent nearest neighbors that only occur in parts of the input, e.g. head of animals or the grass in the background so that the concepts are pertained to certain spacial regions. By encouraging the closeness between each concept and its nearest neighbors, we aim to obtain consistent nearest neighbors to enhance interpretability. Lastly, we optimize the completeness terms to encourage the *completeness* of the discovered concepts.

Learning concepts: To optimize the completeness of the discovered concepts, we optimize the surrogate loss for the completeness term for both concept vectors $\mathbf{c}_{1:m}$ and the mapping function g:

$$\arg\max_{\mathbf{c}_{1:m}, g} \log \mathbb{P}[h_y(g(v_{\mathbf{c}}(\mathbf{x})))] \tag{6.2}$$

An interpretation for finding the underlying concepts whose concept score maximizes the recovered prediction score is analogous to treating the prediction of DNNs as a topic model. By assuming the data generation process of $(\mathbf{x}, y)$ follows the probabilistic graphical model $\mathbf{x}_t \to \mathbf{z}_t$ and $\mathbf{z}_{1:T} \to y$, such that the concept assignment $\mathbf{z}_t$ is generated by the data, and the overall concept assignment $\mathbf{z}_{1:T}$ determines the label y. The log likelihood of the data $\log P[y|\mathbf{x}]$ can be estimated by $\log P[y|\mathbf{x}] = \log \int_z P[y|z]P[z|\mathbf{x}] \approx \log P[y|E[z|\mathbf{x}]]$, by replacing the sampling by a deterministic average. We note that $v_{\mathbf{c}}(\mathbf{x}_{1:T})$ resembles $E[z|\mathbf{x}]$ and $P(y|h(g(v_{\mathbf{c}}(\mathbf{x}))))$ resembles $P[y|E[z|\mathbf{x}]]$, and as in supervised topic modeling [111], we jointly optimize the latent "topic" and the prediction model, but in an end-to-end fashion to maintain efficiency instead of EM update.

To enhance the interpretability of our concepts beyond "topics", we further design a regularizer to encourage the spacial dependency (and thus coherency) of concepts. Intuitively, we require that the top-K nearest neighbor training input patches of each concept to be sufficiently close to the concept, and different concepts are as different as possible. This formulation encourages the top-K nearest neighbors of the concepts would be coherent, and thus allows explainability by ostensive definition. K is a hyperparameter that is usually chosen based on domain knowledge of the desired frequency of concepts. In our results, we fix K to be half of the average class size in our experiments. When using batch update, we find that picking $K = (\text{batch size} \cdot \text{average class ratio})/2$ works well in our experiments, where average class ratio $=$ average instance of each class/total number of instances. That is, the regularizer term tries to maximize $\Phi(\mathbf{x}^i_t) \cdot \mathbf{c}_k$ while minimizing $\mathbf{c}_j \cdot \mathbf{c}_k$. $\Phi(\mathbf{x}^i_t) \cdot \mathbf{c}_k$ is the similarity between the t^{th} patch of the i^{th} example and $\mathbf{c}_j \cdot \mathbf{c}_k$ is the similarity between the j^{th} concept vector and the k^{th} concept vector. By averaging over all concepts, and defining $T_{\mathbf{c}_k}$ as the set of top-K nearest neighbors of $\mathbf{c}_k$, the final regularization term is

$$R(\mathbf{c}) = \lambda_1 \frac{\sum_{k=1}^{m} \sum_{\mathbf{x}^b_a \in T_{\mathbf{c}_k}} \Phi(\mathbf{x}^b_a) \cdot \mathbf{c}_k}{mK} - \lambda_2 \frac{\sum_{j \neq k} \mathbf{c}_j \cdot \mathbf{c}_k}{m(m-1)}.$$

By adding the regularization term to (6.2), the final objective becomes

$$\arg\max_{\mathbf{c}_{1:m},g} \log P(h_y(g(v_{\mathbf{c}}(\mathbf{x}_{1:T})))) + R(\mathbf{c}), \tag{6.3}$$

for which we use stochastic gradient descent to optimize variables $\mathbf{c}_{1:m}, g$ jointly. When the optimization converges, g is a (local) optimal value given $\mathbf{c}_{1:m}$. Since only concept vectors $\mathbf{c}_{1:m}$, and the mapping function g is optimized in the process, the optimization process converges much faster compared to training the model from scratch. The computational cost for discovering concepts and calculating conceptSHAP is about 3 hours for AwA dataset and less than 20 minutes for the toy dataset and IMDB, using a single 1080 Ti GPU, which can be further accelerated with parallelism. The choice of which layer to apply h_y, g and the corresponding architecture are further discussed in the appendix.

6.2.3 ConceptSHAP: How important is each concept?

Given a set of concept vectors $C_S = \{\mathbf{c}_1, \mathbf{c}_2, ...\mathbf{c}_m\}$ with a high completeness score, we would like to evaluate the importance of each individual concept by quantifying how much each individual concept contributes to the final completeness score. Let $\mathbf{s}_i$ denote the importance score for concept $\mathbf{c}_i$, such that $\mathbf{s}_i$ quantifies how much of the completeness score $\eta(C_S)$ is contributed by $\mathbf{c}_i$. Motivated by its successful applications in quantifying attributes for complex systems, we adapt Shapley values [140] to fairly assign the importance of each concept (which we call ConceptSHAP):

Definition 9. Given a set of concepts $C_S = \{\mathbf{c}_1, \mathbf{c}_2, ...\mathbf{c}_m\}$ and some completeness score η, we define the ConceptSHAP $\mathbf{s}_i$ for concept $\mathbf{c}_i$ as

$$\mathbf{s}_i(\eta) = \sum_{S \subseteq C_s \setminus \mathbf{c}_i} \frac{(m - |S| - 1)!|S|!}{m!} [\eta(S \bigcup \{\mathbf{c}_i\}) - \eta(S)],$$

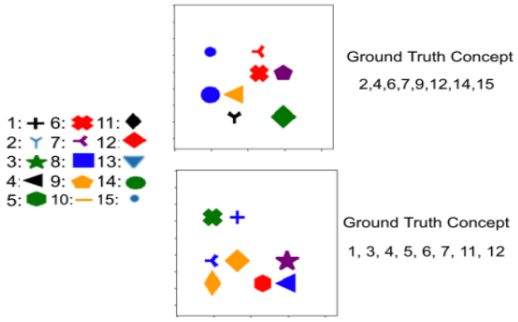
The main benefit of Shapley for importance scoring is that it uniquely satisfies the set of desired axioms: efficiency, symmetry, dummy, and additivity. As these axioms are widely discussed in previous works [104, 140], we leave the definitions and proof to Appendix.

Per-class saliency of concepts: Thus far, conceptSHAP measures the global attribution (i.e., contribution to completeness when all classes are considered). However, per-class saliency, how much concepts contribute to prediction of a particular class, might be informative in many cases. To obtain the concept importance score for each class, we define the completeness score with respect to the class by considering data points that belong to it, which is formalized as:

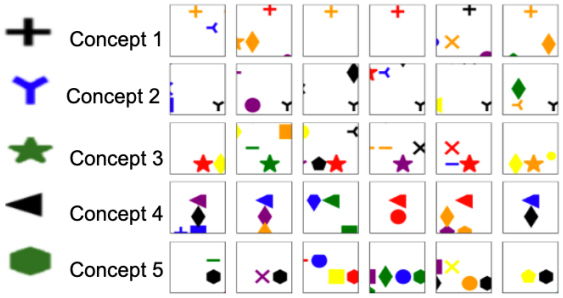
Definition 10. Given a prediction model $\mathsf{p}(\mathbf{x}) = h(\mathbf{f}(\mathbf{x}))$, a set of concept vectors $\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_m$ that lie in the feature subspace in $\mathbf{f}(\cdot)$, we define the completeness score $\eta_{\mathsf{p},j}(\mathbf{c}_1, ..., \mathbf{c}_m)$ for class j as:

$$\eta_{\mathsf{p},j}(\mathbf{c}_1, ..., \mathbf{c}_m) = \frac{\mathbb{P}_{\mathbf{x}, y \in V_j}[y = \arg\max_{y'} h_{y'}(\hat{g}(v_{\mathbf{c}}(\mathbf{x})))] - a_{r,j}}{\mathbb{P}_{\mathbf{x}, y \in V}[y = \arg\max_{y'} \mathsf{p}_{y'}(\mathbf{x}_{1:T})] - a_r}, \tag{6.4}$$

where V_j is the set of validation data with ground truth label j, and $a_{r,j}$ is the accuracy of random predictions for data in class j, and $\hat{g}$ is derived via the optimization of completeness. We then define the perclass ConceptSHAP for concept i with respect to class j as:



(a) Two random images and corresponding ground truth concepts (with their legend on the left) – each object corresponds to a ground truth concept solely via the shape information.



(b) Top nearest neighbors (each neighbor corresponds to a part of the full image) of each discovered concepts. The ground truth concepts, determined by their shape (with random colors), are on the left.

Figure 6.1: Examples (left) and nearest neighbors of our method (right) on Synthetic data.

Definition 11. Given a prediction model $\mathsf{p}(\mathbf{x})$, a set of concept vectors in the feature subspace in $\mathbf{f}(\cdot)$. We can define the perclass ConceptSHAP for concept i with respect to class j as: $\mathbf{s}_{i,j}(\eta) = \mathbf{s}_i(\eta_{\mathsf{p},j})$.

For each class j, we may select the concepts with the highest conceptSHAP score with respect to class j. We note that $\sum_j \frac{|V_j|}{|V|}\eta_{\mathsf{p},j} = \eta$ and thus with the additivity axiom, $\sum_j \frac{|V_j|}{|V|}\mathbf{s}_{i,j}(\eta_{\mathsf{p},j}) = \mathbf{s}_i(\eta)$.

6.3 Experiments

In this section, we demonstrate our method both on a synthetic dataset, where we have ground truth concept importance, as well as on real-world image and language datasets.

6.3.1 Synthetic data with ground truth concepts

Setting: We construct a synthetic image dataset with known and complete concepts, to evaluate how accurately the proposed concept discovery algorithm can extract them. In this dataset, each image contains at most 15 shapes (shown in Fig. 6.1a), and only 5 of them are relevant for the ground truth class, by construction. For each sample $\mathbf{x}^i$, $\mathbf{z}^i_j$ is a binary variable which represents whether $\mathbf{x}^i$ contains shape j. $\mathbf{z}^i_{1:15}$ is a 15-dimensional binary variable with elements independently sampled from Bernoulli distribution with $p = 0.5$. We construct a 15-dimensional multi-label target for each sample, where the target of sample i, y^i is a function that depends only on $\mathbf{z}^i_{1:5}$, which represents whether the first 5 shape exists in $\mathbf{x}^i$. For example, $y_1 = \sim(\mathbf{z_1} \cdot \mathbf{z_3}) + \mathbf{z_4}, y_2 = \mathbf{z_2} + \mathbf{z_3} + \mathbf{z_4}, y_3 = \mathbf{z_2} \cdot \mathbf{z_3} + \mathbf{z_4} \cdot \mathbf{z_5}$, where $\sim$ denotes logical Not (details are in Appendix). We construct 48k training samples and 12k evaluation samples and use a convolutional neural network with 5 layers, obtaining 0.999 accuracy. We take the last convolution layer as the feature layer $\mathbf{f}(\mathbf{x})$.

Evaluations: We conduct a user-study with 20 users to evaluate the nearest neighbor samples of a few concept discovery methods. At each question, a user sees 10 nearest neighbor images of

each discovered concept vector (as shown on the right of Fig. 6.1b), and is asked to choose the most common and coherent shape out of the 15 shapes based on the 10 nearest neighbors. We evaluate the results for our method, k-means clustering, PCA, ACE, and ACE-SP when $m = 5$ concepts are retrieved. Each user is tested on two randomly chosen methods in random order, and thus each method is tested on 8 users. We report the average number of correct concepts and the number of agreed concepts (where the mode of each question is chosen as the correct answer) for each method answered by users in Table 6.1. The average number of correct concepts measures how many of the correct concepts are retrieved by user via nearest neighbors. The average number of agreed concepts measures how consistent are the shapes retrieved by different users, which is related to the coherency and conciseness of the nearest neighbors for each method. We also provide an automated alignment score based on how the discovered concept direction classifies different concepts – see Appendix for details.

Results: We compare our methods to ACE, k-means clustering, and PCA. For k-means and PCA, we take the embedding of the patch as input to be consistent to our method. For ACE, we implement a version which replaces the superpixels with patches and another version that takes superpixels as input, which we refer as ACE and ACE-SP respectively. We report the correct concepts and agreed concepts from the user study, and an automated alignment score which does not require humans. We do not calculate the alignment score of ACE-SP since it does not operate on patches and thus is unfair to compare with others (which would lead to much lower scores.) Our method outperforms others on corrected concepts and alignment score, is superior in retrieving the accurate concepts beyond the limitations of others. The number of agreed concepts is also the highest for our method, showing how highly-interpretability it is to humans such that the same concepts are consistently retrieved based on nearest neighbors. As qualitative results, Fig. 6.1b shows the top-6 nearest neighbors for each concept $\mathbf{c}_k$ of our concept discovery method based on the dot product $\langle \mathbf{c}_k, \Phi(\mathbf{x}_a)^b \rangle$. All nearest neighbors contain a specific shape that corresponds to the ground-truth shapes 1 to 5. For example, all nearest neighbors of concept 1 contain the ground truth shape 1, which are cross as listed in Fig. 6.1a. A complete list of the top-10 nearest neighbors of all concept discovery methods is shown in Appendix.

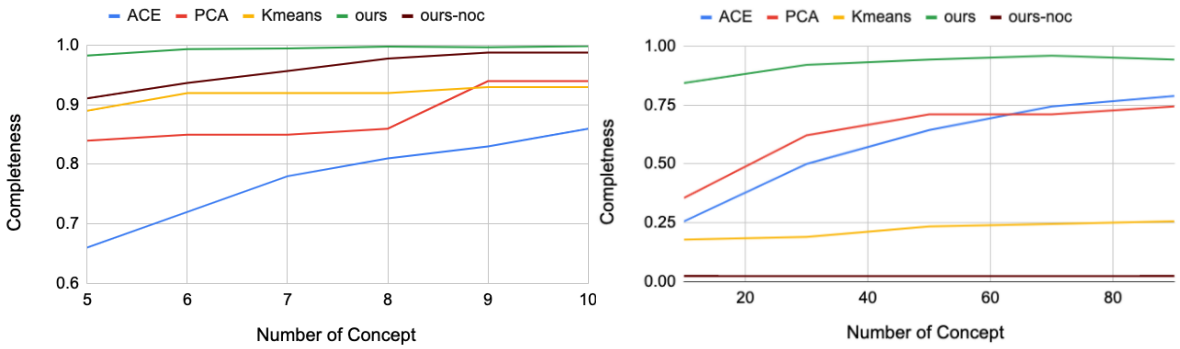


Figure 6.2: Completeness scores on synthetic dataset (left) and completeness scores on AwA (right) versus different number of discovered concepts m for all concept discovery methods in the synthetic dataset. Ours-noc refers to our method without the completeness score objective as an ablation study.

Table 6.1: The average number of correct and agreed concepts by users based on nearest neighbors.

	ACE	ACE-SP	PCA	k-means	**Ours**
correct concepts	3.0 ± 0	2.75 ± 0.46	3.875 ± 0.35	3.75 ± 0.46	$\mathbf{5.0 \pm 0}$
agreed concepts	4.625	4.75	4.375	4.75	**5.0**
automated alignment	0.741	–	0.876	0.864	**0.98**

6.3.2 Image classification

Setting and metrics: We perform experiments on Animals with Attribute (AwA) [96] that contains 50 animal classes. We use 26905 images for training and 2965 images for evaluation. We use the Inception-V3 model, pre-trained on Imagenet [159], which yields 0.9 test accuracy. We apply our concept discovery algorithm to obtain $m = 70$ concepts. We conduct ad-hoc duplicate concept removal, by removing one concept vector if there are two vectors where the dot product is over 0.95. This gives us 53 concepts in total. We then calculate the ConceptSHAP and per class saliency score for each concept and each class. For each class, the top concepts based on the conceptSHAP are the most important concepts to classify this class, as shown in Fig.6.3. While ConceptSHAP is useful in capturing the sufficiency of concepts for prediction, sometimes we may want to show examples. We propose to measure the quality of the nearest neighbors explanations by the average dot product between the nearest-neighbor patches that belongs to the class and the concept vector. In other words, the quality of the nearest neighbors explanations is simply the first term in $R(\mathbf{c})$, which we denote as $R_1(\mathbf{c}) = \sum_{k=1}^{m} \sum_{\mathbf{x}_a^b \subseteq T_{\mathbf{c}_k}} \langle \Phi(\mathbf{x}_a^b), \mathbf{c}_k \rangle$, where the top-K set is limited to image patches in the class of interest. When the nearest neighbor set contains patches of the same original image, we only show the patch with the highest similarity to the concept to increase the diversity.

Results: We show the top concepts (ranked by conceptSHAP) of 3 classes with $R_1(\mathbf{c})$>0.8 in Fig. 6.3 (full results are in Appendix). Note that since our method finds concepts for *all* classes as opposed to specific to one class (such as [23, 54]), we discover common concepts across many classes. For example, concept 7, whose nearest neighbors show grass texture, is important for the classes 'Squirrel', 'Rabbit', 'Bob Cat', since all these animals appear in prairie. Concept 8 shows a oval head and large black round eyes shared by the classes 'Rabbit', 'Squirrel', 'Weasel', while concept 46 shows head of the 'Bob cat', which is shared by the classes 'Lion', 'Leopard',

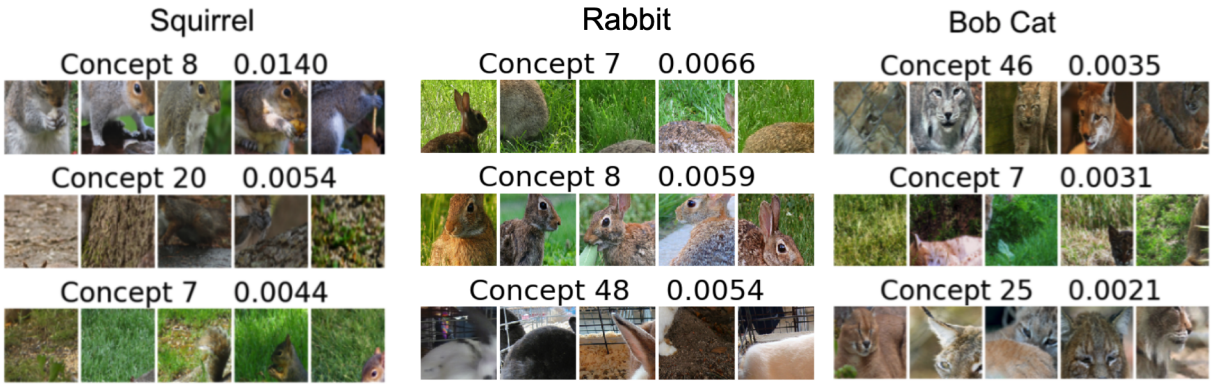


Figure 6.3: Concept examples with the samples that are the nearest to concept vectors in the activation space in AwA. The per-class ConceptSHAP score is listed above the images.

Table 6.2: The 4 discovered concepts and some nearest neighbors along with the most frequent words that appear in top-500 nearest neighbors.

Concept	Nearest Neighbors	Frequent words	ConceptSHAP
1	poorly constructed what comes across as interesting is the wasting my time with a comment but this movie awful in my opinion there were <UNK> and the	worst (168) ever (69) movie (61) seen (55) film (50) awful (42) time(40) waste (34) poorly (26) movies (24) films (18) long (17)	0.280
2	normally it would earn at least 2 or 3 <UNK> <UNK> is just too dumb to be called i feel like i was ripped off and hollywood	not (58) movie (39) make (25) too (23) film (22) even (19) like (18) 2 (16) never (14) minutes (13) 1 (12) doesn't (11)	0.306
3	remember awaiting return of the jedi with almost <UNK> better than most sequels for tv movies i hate male because marie has a crush on her attractive	movies (19) like (18) see (16) movie (15) love (15) good (12) character (11) life (11) little (10) ever (9) watch (9) first (9)	0.174
4	new <UNK> <UNK> via <UNK> <UNK> with absolutely hilarious homosexual and an italian clown <UNK> is an entertaining stephen <UNK> on the vampire <UNK> as a masterpiece	excellent (50) film (25) perfectly (19) wonderful (19) perfect (16) hilarious (15) best (13) fun (12) highly (11) movie (11) brilliant (9) old (9)	0.141

and 'Tiger', 'Antelope', and 'Gorilla', which all show animal heads that are more rectangular and significantly different to the animal heads of concept 8. We find that having concepts shared between classes is useful to interpret the model. Fig. 6.2 shows that our method achieves the highest completeness of all methods on both the synthetic dataset and AwA. As a sanity check, we include the baseline 'ours-noc', where the completeness objective is removed from (6.3). Our method has much higher completeness than 'ours-noc', demonstrating the necessity of the completeness term.

Human Study: We conduct a human study for the top neighbors for concepts discovered by our method, PCA, and Kmeans on the classes 'Squirrel', 'Rabbit', 'Bob Cat'. For each method, we randomly choose 1 top concept per class for the 3 classes, and thus we choose 3 concepts per method (9 concepts in total). For each concept, we show users 4 top images of that concept, and ask users to choose the image (out of 3 different options) that they believe should belong to the same concept (where one of the option will actually belong to the same concept, and the other two are random image patches of the same class that does not belong to that concept). We then calculate the average accuracy to measure the human interpretability of the concept discover method. We conduct a user study with 10 users, where each of them are asked with the same 9 questions (1 question per concept chosen). The average correct ratio for our method, PCA, and Kmeans are 0.733, 0.267, and 0.6 respectively, showing our method's superiority. Kmeans outperforms PCA as it also encourages closeness of top nearest neighbors (which is better for ostensive definition).

6.3.3 Text classification

Setting: We apply our method on IMDB, a text dataset with movie reviews classified as either positive or negative. We use 37500 reviews for training and 12500 for testing. We employ a 4-layer CNN model with 0.9 test accuracy. We apply our concept discover method to obtain 4 concepts, where the part of data $\mathbf{x}_j^i$ consists of 10 consecutive words of the sentence. The completeness of the 4 concepts is 0.97, thus the 4 concepts are highly representative of the classification model.

Result: For each concept, Table 6.2 shows (a) the top nearest neighbors based on the dot product of the concept and part of reviews (b) the most frequent words in the top-500 nearest neighbors

(excluding stop words) (c) the conceptSHAP score for each concept. We can see that concepts 1 and 2 contain mostly negative sentiments, evident from the nearest neighbors – concept 1 tends to criticize the movie/film directly, while concept 2 contains negativity in comments via words such as "not", "doesn't", "even". We note that the ratings in concept 2 are also negative since the scores 1 and 2 are considered to be very negative in movie review. On the other hand, concepts 3 and 4 contain mostly positive sentiments, as evident from the nearest neighbors – concept 3 seems to discuss the plot of the movie without directing acclaiming or criticizing the movie, while concept 4 often contains very positive adjectives such as "excellent", "wonderful" that are extremely positive. More nearest neighbors are provided in the Appendix.

Appending discovered concepts: We perform an additional experiment where we randomly append 5 nearest neighbors (out of 500-nearest neighbors) of each concept to the end of all testing instances for further validation of the usefulness of the discovered concepts. For example, we may add "wasting my time with a comment but this movie" along with 4 other nearest neighbors of concept 1 to the end of a testing sentence. The original average prediction score for the testing sentences is 0.516, and the average prediction score after randomly appending 5 nearest neighbors of each concept becomes 0.103, 0.364, 0.594, 0.678 for concept 1, 2, 3, 4. As a controlled experiment, we appended 5 random sentences to the testing sentences, and the average prediction score is 0.498. This suggests that the concept score is highly related to the how the model makes prediction and may be used to manipulate the prediction. We note that while concept 1 contains stronger and more direct negative words than concept 2, concept 2 has a higher conceptSHAP value than concept 1. We hypothesize this is due to the fact that concept 2 may better detect weak negative sentences that may be difficult to be explained by concept 1, and thus may contribute more to the completeness score.

Chapter 7

Faith-Shap: The Faithful Shapley Interaction Index

Explaining the prediction of a black-box machine learning model via attributions to its features is an increasingly important task. Most approaches have focused on attributions to *individual features*, which does not always suffice to provide insight into the model when there are heavy feature interactions. For instance, when explaining models with text input, we might also ask for attributions to phrases and sequences of words rather than just individual words. Similarly, in Question Answering (QA), it is of interest to measure attributions to query answer tuples, rather than just individual entities associated with answers. Such feature interactions are also salient with images as input, where instead of attributions to individual pixels, we might prefer attributions to groups of pixels.

A large class of recent approaches for individual feature attributions reduce the task to a cooperative game theory problem. Given a machine learning model, a test point, and the underlying data distribution, one can devise a "set value function" that takes as input a set of features, and outputs the value of that set of features. There are many choices for such a reduction to a set function [24, 46, 104, 155]. We can then relate this to a cooperative game theory problem where the features are players, the set function above is the value function of the coalition game that specifies the value of various player coalitions, and we wish to derive feature attributions given such a value function. This meta-approach has led to a slew of explanation approaches when the goal is to obtain individual feature attributions. The key question we focus on in this paper is to obtain attributions to *feature interactions* instead. In this setting, any feature interactions (up to a given order), along with each individual features, should get some attribution score. This question has attracted some attention in the cooperative game theory and the explainable AI literature, with the broad strategy of extending popular approaches for individual feature attributions, such as Shapley and Banzhaf values [71, 139], to the interaction context. But these existing proposals come with many caveats.

Part of the attraction of the cooperative game theory based explanations above is that for the case of individual feature attributions, if we stipulate some natural axioms such as linearity, symmetry, dummy, and efficiency (detailed in a later section), there exist unique attributions such as Shapley and Banzhaf (depending on the notion of efficiency). Thus we have both a strong axiomatic foundation to the explanations, as well as a very compelling uniqueness result that there

can exist no other explanations that satisfy these axioms. These have thus led to an explosion of Shapley value based explanations in the XAI literature that assign attributions to features, data, and even concepts [31, 50, 63, 79, 102, 104, 119, 120, 182]. However,when we move to the context of feature interactions, while the axioms above have natural extensions from the individual feature to the feature interaction context, they no longer result in a *unique feature attribution value*.

Approaches to address this have thus focused on adding additional less natural axioms to ensure uniqueness. One set of unique feature attributions — Shapley interaction and Banzhaf interaction indices [59] — derive unique attributions via a *recursive axiom*, which specifies how higher order feature attributions be derived from lower order feature interaction attributions (all the way to individual feature attributions). Thus, given the uniqueness at the level of individual feature attributions, we in turn get uniqueness at all levels of interaction attributions. One major caveat of these Shapley interaction and Banzhaf interaction indices is that they do not satisfy the efficiency axiom for interaction feature attributions, and hence can no longer be viewed as distributing the total contribution of the model prediction among all feature interactions. The other caveat is that the recursive axiom, while convenient to extend uniqueness from individual to interaction feature attributions, is much less "natural" when compared with the original Shapley axioms. To address these caveats, Sundararajan et al. [158] proposed the *interaction distribution axiom* that entails distributing higher order interactions to the topmost interaction indices at the expense of impoverished lower order interactions. This makes the interaction attributions unique for unanimity games [139], and since these act as a basis for set value functions, by linearity this ensures uniqueness of interaction attributions for general games. The caveat however is that the specified attribution distribution inordinately favors the topmost interactions, which in turn affects the usefuless of both the lower and highest order interactions as we show in our examples. And arguably, the interaction distribution axioms too is much less natural when compared to the original Shapley axioms. Thus, there remains an open problem to specify a "natural" restriction or axiom that allows for unique interaction attributions.

An additional desideratum is that the feature interaction attributions be cognizant of the *maximum interaction order* of the interaction attributions we require. For instance, with individual feature attributions, the maximum interaction order is one, while with pairwise feature attributions, the maximum interaction order is two. This would allow the explanations to be tailored to the set of possible interactions and satisfy the relevant axioms with respect to just these interactions, instead of all possible subsets of feature interactions.

In this work, rather than devising potentially less natural axioms to ensure uniqueness, we work from yet another viewpoint of Shapley values, that they be faithful to the set value function: for all subsets, the sum of individual feature attributions over a subset should approximate the set value function evaluated on that subset. When formalized as a weighted regression problem, this yields Shapley and Banzhaf values depending on the weights in the weighted regression [13, 132]. We then extend the above weighted regression to feature interactions up to a given maximum interaction order, which then yields what we call Faith-Interaction indices. We show that when restricting to the class of Faith-Interaction indices, together with the (interaction extensions of the individual) Shapley axioms, we obtain a unique interaction index, which we term the Faith-Shap (for Faithful Shapley Interaction) index, and which reduces to the individual feature Shapley values when the top interaction order is one. We thus posit Faith-Shap as the natural extension of Shapley

values from individual features to interaction indices. Similarly, when the efficiency axiom is replaced by the generalized 2-efficiency axiom, we obtain a unique interaction index, which we term Faith-Banzhaf (for Faithful Banzhaf Interaction) index. The latter has also appeared in other guises in prior work [61, 67]. Unlike the other restrictive axioms discussed earlier, here we only require that the explanations be faithful to the model, which has always been a big attraction of Shapley values in the explainable AI (XAI) context. We corroborate the usefulness of these Faith-Interaction indices by contrasting them with prior indices in two illustrative coalition games, as well as real-world XAI applications. We then discuss algebraic properties of Faithful Shapley Interaction index by relating them to cardinal indices, i.e. indices that can be expressed as a linear combinations of marginal contributions, as well as in terms of approximations to multilinear extensions of the coalition set value function. An additional benefit of the Faith Interaction indices is that the estimation becomes much more efficient via leveraging the weighted linear regression formulation, and which we validate in our experiments.

7.1 Preliminaries

7.1.1 Notations

Suppose we are given a black-box model $f : \mathcal{X} \mapsto \mathbb{R}^d$, with input domain $\mathcal{X} \subseteq \mathbb{R}^d$; and suppose we wish to explain its prediction at a given test point $\mathbf{x} \in \mathcal{X}$. Suppose also given the tuple $f, \mathbf{x}$ (and possibly with additional information about the underlying data distribution on which f is trained on, and from which $\mathbf{x}$ is drawn), there is a well-defined set function $\mathbf{f}_\mathbf{x} : 2^d \to \mathbb{R}^d$. We can interpret such a set function as specifying the value of a subset of the set of d features. Many popular explanations employ such a reduction of the model and its prediction context to set value functions; see Lundberg and Lee [104], Ribeiro et al. [129], Sundararajan et al. [158] for many examples. When clear from the context, and for notational simplicity, we will often omit $\mathbf{x}$ and simply use $\mathbf{f}$ to denote the set function. Such a reduction allows us to leverage results from cooperative game theory, by relating the set of features to a set of players, and the set function above as specifying the values of coalitions of players.

We are then interested in quantifying the importance of interactions between different features up to some order $\ell \in [d]$. Note that in this context, when we mean interactions between features, we mean non-self interactions between distinct features, since self-self interactions could simply be identified with the individual features. In other words, we require an importance function Φ which for each coalition $S \subseteq [d]$ where $0 \leq |S| \leq \ell$, outputs a scalar $\Phi_S(\mathbf{f}, \ell)$. Let $\mathcal{S}_\ell$ denote the set of all subsets of $[d]$ with size less than or equal to ℓ; the size of this set can be seen to be $d_\ell := \sum_{j=0}^{\ell} \binom{d}{j}$. We then use the shorthand $\Phi(\mathbf{f}, \ell) = (\Phi_S(\mathbf{f}, \ell))_{S \in \mathcal{S}_\ell} \in \mathbb{R}^{d_\ell}$. To simplify notation, we omit braces for small sets and write $T \cup i$ to represent $T \cup \{i\}$.

7.1.2 Definitions

We begin by recalling the concept of discrete derivatives.

Definition 1. *(Discrete Derivative) Given a set function $\mathbf{f} : 2^d \mapsto \mathbb{R}$ and two finite disjoint coalitions $S, T \subseteq [d]$ with $S \cap T = \emptyset$, the S-derivative of $\mathbf{f}$ at T, $\Delta_S(\mathbf{f}(T))$, is defined recursively*

as follows:

$$\Delta_i \mathbf{f}(T) = \mathbf{f}(T \cup i) - \mathbf{f}(T), \ \ \forall i \in [d], \text{ and} \tag{7.1}$$

$$\Delta_S(\mathbf{f}(T)) = \Delta_i[\Delta_{S\setminus i}(\mathbf{f}(T))] = \sum_{L \subseteq S} (-1)^{|S|-|L|} \mathbf{f}(T \cup L), \forall i \in S. \tag{7.2}$$

The second equality in Eqn. (7.2) can be shown via induction on S [47]. As an illustration of discrete derivatives, for a subset S of size 2, the discrete derivative can be written as

$$\Delta_{\{i,j\}} \mathbf{f}(T) = \mathbf{f}(T \cup \{i,j\}) - \mathbf{f}(T \cup j) - \mathbf{f}(T \cup i) + \mathbf{f}(T).$$

$\Delta_{\{i,j\}} \mathbf{f}(T)$ captures the joint effect of features i and j co-occurring compared to the individual effects of i and j. If $\Delta_{\{i,j\}} \mathbf{f}(T) > 0$ (resp. < 0), we say i and j have positive (resp. negative) interaction effect in the presence of T since the presence of i increases (resp. decreases) the marginal contribution of j to coalition T. Following the intuition from the two features example, the discrete derivative $\Delta_S(\mathbf{f}(T))$ can be viewed as a measurement of the *marginal interaction of S in the presence of T*. When a set of features have a positive (negative) interaction effect, the discrete derivative is positive (negative). Discrete derivatives play a fundamental role in measurement of interaction effects. As we will see in the following section, the Shapley and Banzhaf interaction indices can be viewed as a weighted average of S-derivatives over all subsets $T \subseteq [d] \setminus S$.

Next, let us recall the concept of the Möbius transform.

Definition 2. *(Möbius transform) Given set function $\mathbf{f} : 2^d \mapsto \mathbb{R}$, the Möbius transform of $\mathbf{f}(\cdot)$ is*

$$a(\mathbf{f}, S) = \sum_{T \subseteq S} (-1)^{|S|-|T|} \mathbf{f}(T) \ \ \textit{for all} \ \ S \subseteq [d]. \tag{7.3}$$

An important property [139] of the Möbius transform is that any set function $\mathbf{f}(\cdot)$ can be expressed as:

$$\mathbf{f} = \sum_{R \subseteq [d]} a(\mathbf{f}, R)\, \mathbf{f}_R, \tag{7.4}$$

where $\mathbf{f}_R$ for any $R \subseteq [d]$ has the form $\mathbf{f}_R(S) = 1$ if $S \supseteq R$ and 0 otherwise; and is also known as a *unanimity game* value function in game theory. Eqn. (7.4) states that any set function can be expressed as a linear combinations of these unanimity game value functions (so that $\{\mathbf{f}_R\}_{R \subseteq [d]}$ form a basis for real-valued set value functions), with the Möbius transforms $a(\mathbf{f}, R)$ as their coefficients. Note that if an interaction index satisfies the **interaction linearity axiom** (to be discussed in the sequel), the interaction index for general set value functions can be expressed as a linear combination of the interaction indices for unanimity games.

7.2 Background: Axioms for Interaction Indices

In this section, we present natural extensions of Shapley axioms for individual features to the feature interactions [59, 158]. We then discuss the key interaction indices proposed so far in the literature — the Shapley interaction index, Banzhaf interaction index and Shapley-Taylor

interaction index — with respect to these axioms. In all these axioms, we allow for dependence on the maximum interaction order $\ell \in [d]$.

Axiom 11. *(Interaction Linearity): For any maximum interaction order $\ell \in [d]$, and for any two set functions $\mathbf{f}_1$ and $\mathbf{f}_2$, and any two scalars $\alpha_1, \alpha_2 \in \mathbb{R}$, the interaction index satisfies:* $\Phi(\alpha_1 \mathbf{f}_1 + \alpha_2 \mathbf{f}_2, \ell) = \alpha_1 \Phi(\mathbf{f}_1, \ell) + \alpha_2 \Phi(\mathbf{f}_2, \ell)$.

The interaction linearity axiom states that the feature interaction index is a linear functional of the set function $\mathbf{f}(\cdot)$. It ensures that the corresponding indices scale with the value function $\mathbf{f}(\cdot)$.

Axiom 12. *(Interaction Symmetry): For any maximum interaction order $\ell \in [d]$, and for any set function $\mathbf{f} : 2^d \mapsto \mathbb{R}$ that is symmetric to elements $i, j \in [d]$, so that $\mathbf{f}(S \cup i) = \mathbf{f}(S \cup j)$ for any $S \subseteq [d] \backslash \{i, j\}$, the interaction index satisfies: $\Phi_{T \cup i}(\mathbf{f}, \ell) = \Phi_{T \cup j}(\mathbf{f}, \ell)$ for any $T \subseteq [d] \backslash \{i, j\}$ with $|T| < \ell$.*

The interaction symmetry axiom entails that if the value function treats two features the same, their corresponding feature interaction index values should be the same as well.

Axiom 13. *(Interaction Dummy): For any maximum interaction order $\ell \in [d]$, and for any set function $\mathbf{f} : 2^d \mapsto \mathbb{R}$ such that $\mathbf{f}(S \cup i) = \mathbf{f}(S)$ for some $i \in [d]$ and for all $S \subseteq [d] \backslash \{i\}$, the interaction index satisfies: $\Phi_T(\mathbf{f}, \ell) = 0$ for all $T \in \mathcal{S}_\ell$ with $i \in T$.*

The interaction dummy axiom entails that a dummy feature $i \in [d]$ that has no influence on the function $\mathbf{f}$ should have no interaction effect with the other features.

Axiom 14. *(Interaction Efficiency): For any maximum interaction order $\ell \in [d]$, and for any set function $\mathbf{f} : 2^d \to \mathbb{R}$, the interaction index satisfies: $\sum_{S \in \mathcal{S}_\ell \backslash \emptyset} \Phi_S(\mathbf{f}, \ell) = \mathbf{f}([d]) - \mathbf{f}(\emptyset)$ and $\Phi_\emptyset(\mathbf{f}, \ell) = \mathbf{f}(\emptyset)$.*

The interaction efficiency axiom ensures that the interaction index distributes the total value $\mathbf{f}([d])$ among the different subsets in $\mathcal{S}_\ell$. This form of interaction efficiency has also been considered by Sundararajan et al. [158]. As we will detail in the sequel, some of the recently proposed interaction indices do not satisfy such an efficiency axiom. For instance, the chaining interaction and Shapley interaction indices only requires the total sum of *individual feature importances* to sum to $\mathbf{f}([d]) - \mathbf{f}(\emptyset)$, without consideration of the higher order interaction importances.

Challenge: Lack of Uniqueness: These axioms are natural extensions to the interaction setting of classical axioms for individual feature attributions; see Fujimoto et al. [47], Grabisch and Roubens [59] for a counterpart of these interaction axioms without consideration of the maximum interaction order $\ell \in [d]$. As Sundararajan et al. [158] note, though the linearity, symmetry, dummy and efficiency axioms uniquely specify a feature attribution when the maximum interaction order $\ell = 1$ (i.e. for individual feature attributions), they no longer do when $\ell > 1$. In other words, there could exist many interaction indices that all satisfy the axioms specified above. A big attraction of the individual Shapley value was its uniqueness given the corresponding individual attribution axioms. Accordingly, a line of work has focused on specifying additional axioms that together specify a unique interaction index.

Axiom 15. *(Recursive Interaction): For any maximum interaction order $2 \leq \ell \leq d$, and for any set function $\mathbf{f} : 2^d \to \mathbb{R}$, let the reduced set functions $\mathbf{f}^{[d]\backslash j}, \mathbf{f}^{[d]\backslash j}_{\cup j} : 2^{d-1} \to \mathbb{R}$ be defined as:*

$$\mathbf{f}^{[d]\backslash j}(T) = \mathbf{f}(T), \quad \text{and} \quad \mathbf{f}^{[d]\backslash j}_{\cup j}(T) = \mathbf{f}(T \cup j) - \mathbf{f}(j) \ \ , \forall\, T \subseteq [d] \backslash j.$$

Then the interaction index satisfies: $\Phi_S(\mathbf{f}, \ell) = \Phi_{S\setminus j}(\mathbf{f}^{[d]\setminus j}_{\cup j}, \ell) - \Phi_{S\setminus j}(\mathbf{f}^{[d]\setminus j}, \ell),\ \forall S \in \mathcal{S}_\ell$ *with* $|S| \geq 2$, *and* $\forall j \in S$.

The recursive axiom above is a natural extension of the recursive axiom of Grabisch and Roubens [59] to account for arbitrary maximum interaction orders. The axiom can be informally interpreted as "how does the presence or absence of feature j influence the share of feature set S". But more importantly (and the reason is it is termed the recursive axiom) is that it specifies how higher-order interaction scores are *uniquely determined* given lower-order interaction indices. By recursion, the higher-order interaction indices are thus uniquely specified given just the singleton feature attributions. The reason this helps with uniqueness is that so long as the axioms entail unique singleton attributions, together with this recursive axiom, they would entail unique interaction attributions.

Shapley Interaction Index: Grabisch and Roubens [59] thus show that there is a unique interaction index that satisfies the interaction linearity, symmetry, dummy, and the recursive axioms (but not the efficiency axiom), and whose restrictions to singleton sets corresponds to Shapley values. They term this interaction index Shapley interaction index. This Shapley interaction index has the following closed form:

$$\Phi^{\text{Shap}}_S(\mathbf{f}, \ell) = \sum_{T \subseteq [d]/S} \frac{|T|!(d - |S| - |T|)!}{(d - |S| + 1)!} \Delta_S(\mathbf{f}(T)), \quad \forall S \in \mathcal{S}_\ell. \tag{7.5}$$

A critical caveat of the the resulting Shapley interaction value is that it no longer satisfies the interaction efficiency axiom when the maximum interaction order $\ell > 1$. Indeed, simply summing the contributions to singleton sets (i.e. the classical individual attribution Shapley values) is already equal to $\mathbf{f}([d]) - \mathbf{f}(\emptyset)$, so the only way for the interaction efficiency axiom to be satisfied if all the other interaction attributions sum to zero, which they do not.

Banzhaf Interaction Index: Grabisch and Roubens [59] further show that there is a unique interaction index that satisfies the interaction linearity, symmetry, dummy, and the recursive axioms (but not the interaction efficiency axiom), and whose restrictions to singleton sets corresponds to the Banzhaf values. They term this interaction index Banzhaf interaction index, which has the following closed form:

$$\Phi^{\text{Bzf}}_S(\mathbf{f}, \ell) = \sum_{T \subseteq [d]/S} \frac{1}{2^{d-|S|}} \Delta_S(\mathbf{f}(T)), \quad \forall S \in \mathcal{S}_\ell. \tag{7.6}$$

It can be again shown that the Banzhaf interaction index does not satisfy the interaction efficiency axiom even when $\ell = 1$; though they do satisfy the generalized 2-efficiency axiom, which can be stated as follows.

Axiom 16. *(Generalized Interaction 2-Efficiency): Define the reduced function* $\mathbf{f}_{[ij]} : 2^{d-1} \to \mathbb{R}$ *given any* $i, j \in [d]$ *as* $\mathbf{f}_{[ij]}(S) = \mathbf{f}(S)$ *for all sets* S *containing both* i *and* j, *and* $\mathbf{f}_{[ij]}(S \cup [ij]) = \mathbf{f}(S \cup \{i, j\})$ *for all* S *containing neither* i *nor* j. *That is, the reduced function considers features* i *and* j *together as a group* $[ij]$. *Then the interaction index satisfies:* $\Phi_{S\cup[ij]}(\mathbf{f}_{[ij]}, \ell) = \Phi_{S\cup i}(\mathbf{f}, \ell) + \Phi_{S\cup j}(\mathbf{f}, \ell)$ *for all* $S \subseteq [d]\setminus\{i, j\}$, *and* $\ell = |S| + 1$.

The generalized interaction 2-efficiency axiom above is an extension of the generalized 2-efficiency axiom of [59] to account for arbitrary maximum interaction orders. It states that when features i, j form a group in the set function $\mathbf{f}_{[ij]}$ with $d-1$ features, the importance of $S \cup [ij]$ equals to the sum of importances of $S \cup i$ and $S \cup j$ with respect to the original set value function. When $S = \emptyset$ and $\ell = 1$, it reduces to the classical 2-efficiency axiom [71] that indicates that the importance of $[ij]$ as a group should be equal to the sum of importance of individual features i and j.

Shapley Taylor Interaction Index: Sundararajan et al. [158] stipulate an additional *interaction distribution (ID) axiom*, which can be stated as follows.

Axiom 17. *(Interaction distribution [158]): Define $\mathbf{f}_T$ parameterized by a set $T \subseteq [d]$ as $\mathbf{f}_T(S) = 0$ if $T \not\subseteq S$ and $\mathbf{f}_T(S) = 1$ otherwise. Then for all $\ell \in [d]$, and for all S with $S \not\subseteq T$ and $|S| < \ell$, the interaction index satisfies: $\mathcal{E}_S(\mathbf{f}_T, \ell) = 0$.*

The key idea behind the ID axiom is to uniquely specify an interaction index for unanimity games $\{\mathbf{f}_T\}_{T \subseteq [d]}$, given the interaction linearity, symmetry, dummy and efficiency axioms. Since unanimity games form a basis for the set of all games, in the presence of interaction linearity axiom, we then get unique interaction indices. They thus show that there exists a unique interaction index that satisfies interaction linearity, symmetry, dummy, efficiency, and interaction distribution axioms and which they term Shapley Taylor index (for reasons which will become clearer in a later section when we discuss algebraic properties of various interaction indices). The Shapley Taylor interaction index has the following closed form:

$$\Phi_S^{\text{Taylor}}(\mathbf{f}, \ell) = \begin{cases} \Delta_S(\mathbf{f}(\emptyset)) & \text{, if } |S| < \ell. \\ \sum_{T \subseteq [d]/S} \frac{|T|!(d-|T|-1)!|S|}{d!} \Delta_S(\mathbf{f}(T)) & \text{, if } |S| = \ell. \end{cases} \tag{7.7}$$

A key advantage of this interaction index is that it depends on the maximum interaction order ℓ, in contrast to previously proposed interaction indices such as the Shapley interaction and Banzhaf interaction indices. Indeed, in order for an interaction index to satify the interaction efficiency axiom for maximum interaction order ℓ, it has to distribute the contributions among subsets in $\mathcal{S}_\ell$, and hence has to be cognizant of the maximum interaction order ℓ. However, a key caveat of the interaction distribution axiom is that the specified attribution distribution inordinately favors the topmost interaction. As can be seen from Eqn.(7.7), the importance of a set S with $|S| < \ell$ is only specified by the marginal contribution of S in the presence of the empty set, and not the presence of other subsets $T \subseteq [d] \backslash S$. This impoverishes lower-order interactions, which in turn hurts the meaningfulness of both lower and highest order interactions as we will show in Section 7.4.

Thus a key open question that this section has made salient is: how do we more naturally constrain interaction indices beyond interaction linearity, symmetry, dummy and efficiency axioms, so as to obtain a unique interaction index?

7.3 Faith-Interaction Indices

In this section, in contrast to additional axioms, we draw from another viewpoint of singleton Shapley feature attributions: that they are faithful to the underlying value function.

Faithfulness of Singleton Shapley Values: Given singleton feature attributions $\{\Phi_i\}_{i\in[d]}$, we can require that:

$$\mathbf{f}(S) \approx \sum_{i\in S} \Phi_i,\ \forall S \subseteq [d].$$

Note that we can only ask for approximate rather than exact equality for all sets S, since an exact equality would entail we solve 2^d linear equalities (corresponding to the subsets of $[d]$) with d variables (corresponding to the d singleton feature attributions $\{\Phi_i\}_{i\in[d]}$), which may not always have a feasible solution. One approach to formalize such an approximate equality is via weighted regression:

$$\min_{\Phi\in\mathbb{R}^{d+1}} \sum_{S\subseteq[d]} \mu(S) \left(v(S) - \Phi_\emptyset - \sum_{i\in S} \Phi_i \right)^2, \tag{7.8}$$

where $\mu : 2^{[d]} \mapsto \mathbb{R}^+ \cup \{\infty\}$ is some weighting over the subsets $S \subseteq [d]$ which can be interpreted as the importance of different coalitions. Note that the range of μ is the extended positive reals. When $\mu(S) = \infty$ for some sets S, we can interpret the above as solving the constrained problem:

$$\min_{\Phi\in\mathbb{R}^{d+1}} \sum_{S\subseteq[d]\,:\,\mu(S)<\infty} \mu(S) \left(v(S) - \sum_{i\in S} \Phi_i \right)^2 \text{ s.t. } \mathbf{f}(S) = \sum_{i\in S} \Phi_i,\ \forall S :\ \mu(S) = \infty.$$

It has been shown that we can recover the singleton Shapley values as the solution of the weighted regression problem above by setting $\mu(S) \propto \frac{d-1}{\binom{d}{|S|}\,|S|\,(d-|S|)}$ and $\mu(\emptyset) = \mu([d]) = \infty$ [22]. And we can recover singleton Banzhaf values by using the uniform distribution $\mu(S) = 1/2^d$ [67].

From Singleton Attributions to Interaction Indices: In this section, we consider the natural generalization of the above to *interaction indices*, so that we now require:

$$\mathbf{f}(S) \approx \sum_{T\subseteq S, |T|\leq \ell} \mathcal{E}_T(\mathbf{f}, \ell),\ \forall S \subseteq [d].$$

Again here we ask for approximate rather than exact equality since when the order of interactions is less than the number of features, so that $\ell < d$, the latter would entail we solve 2^d linear equalities with d_ℓ variables, which may not always have a feasible solution. Accordingly, we consider the following weighted regression problem as a formalization of the above:

$$\Phi(\mathbf{f}, \ell) = \arg\min_{\mathcal{E}\subseteq\mathbb{R}^{d_\ell}} \sum_{S\subseteq[d]} \mu(S) \left(\mathbf{f}(S) - \sum_{T\subseteq S, |T|\leq\ell} \mathcal{E}_T(\mathbf{f}, \ell) \right)^2, \tag{7.9}$$

where $\mu : 2^d \to \mathbb{R}^+ \cup \{\infty\}$ is a coalition weighting function. And as before of $\mu(S) = \infty$ for some sets S, we can interpret above as solving the constrained problem:

$$\begin{aligned}\Phi(\mathbf{f},\ell) \;&= \underset{\mathcal{E}\subseteq\mathbb{R}^{d_\ell}}{\arg\min} \sum_{S\subseteq[d]\,:\,\mu(S)<\infty} \mu(S)\left(\mathbf{f}(S) - \sum_{T\subseteq S,|T|\leq\ell} \mathcal{E}_T(\mathbf{f},\ell)\right)^2 \\ &\text{s.t. } \mathbf{f}(S) = \sum_{T\subseteq S,|T|\leq\ell} \mathcal{E}_T(\mathbf{f},\ell), \;\; \forall S:\, \mu(S)=\infty. \end{aligned} \tag{7.10}$$

We note that the range of the weighting function μ is not allowed to include zero since it is a necessary condition to ensure that there exists a unique minimizer (which is shown in the following proposition 8). This is not an issue in practice since we can always choose an arbitrary small positive value instead of zero to approximate the intended constraint that $\mu(S) = 0$ for some $S \subseteq [d]$.

Proposition 7. *If the coalition weighting function $\mu(\cdot)$ is finite such that $\mu(S) \in \mathbb{R}^+$ for all $S \subseteq [d]$, Eqn.(7.9) is strictly convex.*

Given that Eqn.(7.9) is strictly convex, we next show that the minimization problems have a unique minimizer.

Proposition 8. *The (constrained) regression problems defined in Eqn.(7.10) with a proper weighting function μ (Definition 3) have a unique minimizer.*

This proposition is a straight-forward application of the following fact: For a minimization problem with linear constraints, if the objective is strictly convex, then it has a unique minimizer.

Also, we note that having positive measure for all subsets of $[d]$ on the weighting function $\mu(\cdot)$ is necessary to ensure the uniqueness of the minimizer. Consider the case when the maximum interaction order equals to the number of feature, i.e. $\ell = d$, there are 2^d variables with 2^d equalities. That is, $\mathbf{f}(S) - \sum_{T\subseteq S} \Phi_S(\mathbf{f},d) = 0$ for all $S \subseteq [d]$. In this case, we can not have any $S \subseteq [d]$ such that $\mu(S) = 0$ due to the lack of equations.

In this special case of $\ell = d$, we have the following closed-form expression. We note that this results are independent of the weighting function as long as we have $\mu(S) > 0$ for all $S \subseteq [d]$.

Proposition 9. *When the maximum interaction order $\ell = d$, the minimizer of Eqn.(7.10) the Möbius transform of $\mathbf{f}$, i.e. $\Phi_S(\mathbf{f},d) = a(\mathbf{f},S) = \sum_{T\subseteq S}(-1)^{|S|-|T|}\mathbf{f}(T)$ for all subsets $S \subseteq [d]$.*

We can also see from Eqn. (7.9) that when the weighting function is infinite for many subsets, this entails corresponding equality constraints on the interaction index, which may not have a feasible solution. We thus consider the following set of what we term *proper* weighting functions.

Definition 3. *(Proper weighting function) We say that a weighting function $\mu : 2^d \mapsto \mathbb{R}^+ \cup \{\infty\}$ is proper if $\mu(S)$ is finite for all $S \subseteq [d]$ with $1 \leq S \leq d-1$.*

This then leads to our definition of Faith-interaction indices.

Definition 4. *(Faith-Interaction Indices): We say that $\mathcal{E}$ is a Faith-Interaction index, given any set value function $\mathbf{f} : 2^d \to \mathbb{R}$ and any maximum interaction order $\ell \in [d]$, if there exists a proper weighting function $\mu : 2^d \to \mathbb{R}^+ \cup \{\infty\}$ such that $\mathcal{E}(\mathbf{f},\ell)$ minimizes the corresponding weighted regression objective in Eqn.(7.10).*

When the coalition weighting function μ is fully finite so that $\mu(S)$ are finite for all sets

$S \subseteq [d]$, Faith-interaction indices have a simple closed-form expression as detailed in the following proposition.

Proposition 10. *Any Faith-Interaction index $\mathcal{E}(\mathbf{f}, \ell)$ with respect to a finite weighting function $\mu(\cdot)$ has the form:*

$$\mathcal{E}(\mathbf{f}, \ell) = \left(\sum_{S \subseteq [d]} \mu(S) p(S) p(S)^T \right)^{-1} \sum_{S \subseteq [d]} \mu(S) \mathbf{f}(S) p(S), \tag{7.11}$$

where $p : 2^{[d]} \to \{0,1\}^{d_\ell}$ is specified as: $p(S)[T] = \mathbb{1}[(T \subseteq S) \vee (T = \emptyset)]$ for any $T \in \mathcal{S}_\ell$.

When the coalition weighting function $\mu(\cdot)$ is not fully finite, we have a linearly constrained least squares problem which does not have a closed form, but whose solution can be characterized via its Lagrangian.

7.3.1 Axiomatic Characterization of Faith-Interaction Indices

In this section, we investigate the axiomatic properties of our class of Faith-Interaction indices. We first show that all faith-interaction indices satisfy the interaction linearity axiom.

Proposition 11. *Faith-Interaction indices $\mathcal{E}$ satisfy the interaction linearity axiom.*

For Faith-Interaction indices corresponding to finite coalition weighting functions $\mu(\cdot)$, this result easily follows from Proposition 10 that these are linear functionals of the set value function $\mathbf{f}(\cdot)$. For Faith-Interaction indices where the weighting function is no longer finite for some sets $S \in \{\emptyset, [d]\}$, they solve a linearly constrained least squares problem which does not have a closed-form solution. But by a more nuanced analysis of its Lagrangian, we can again show that the interaction indices are linear functionals of the set value function $\mathbf{f}(\cdot)$.

We next show that Faith-Interaction indices also satisfy the linearity symmetry axiom provided that the weighting functions are permutation invariant ("symmetric"), and hence only depend on the size of the set.

Proposition 12. *Faith-Interaction indices $\mathcal{E}$ satisfy the interaction symmetry axiom if and only if the weighting functions are permutation invariant, and hence only depend on the size of the set, so that $\mu(S)$ is only a function of $|S|$.*

We next consider the dummy axiom.

Proposition 13. *Faith-Interaction indices $\mathcal{E}$ satisfy the interaction dummy axiom if the features behave independently of each other when forming coalitions in the weighting function, so that the coalition weighting functions can be expressed as $\mu(S) \propto \prod_{i \in S} p_i \prod_{j \notin S} (1 - p_j)$ for all $S \subseteq [d]$, where $0 < p_i < 1$ is the probability of the feature i to be present.*

Proposition 13 implies that a dummy feature has no impact on other features when the weighting function treats features independently.

So far, we have analyzed when Faith-Interaction indices satisfy the interaction linearity, symmetry, and dummy axioms. When they satisfy all three simultaneously, and the coalition weighting function is finite, then we can show that the latter has a specific algebraic form.

Theorem 18. *Faith-Interaction indices $\mathcal{E}$ with a finite weighting function satisfy the interaction linearity, symmetry, and dummy axioms if and only if the weighting function μ has the following*

form:

$$\mu(S) \propto \sum_{i=|S|}^{d} \binom{d-|S|}{i-|S|}(-1)^{i-|S|} g(a,b,i), \text{ where } g(a,b,i) = \begin{cases} 1 & \text{if } i=0 \\ \prod_{j=0}^{j=i-1} \frac{a(a-b)+j(b-a^2)}{a-b+j(b-a^2)} & \text{if } 1 \le i \le d, \end{cases} \tag{7.12}$$

for some $a, b \in \mathbb{R}^+$ with $a > b$ such that $\mu(S) > 0$ for all $S \subseteq [d]$.

Theorem 18 shows the surprising fact that Faith-Interaction indices satisfying the interaction linearity, symmetry, and dummy axioms with finite weighting functions have only two degrees of freedom: $a, b \in \mathbb{R}$. Given these, we can fully specify the weighting function, and hence the corresponding Faith-Interaction indices.

Theorem 18 states that the finite weighting function must in the following form:

$$\mu(S) \propto \sum_{i=|S|}^{d} \binom{d-|S|}{i-|S|}(-1)^{i-|S|} g(a,b,i), \text{ where } g(a,b,i) = \begin{cases} 1 & \text{, if } i=0. \\ \prod_{j=0}^{j=i-1} \frac{a(a-b)+j(b-a^2)}{a-b+j(b-a^2)} & \text{, if } 1 \le i \le d. \end{cases}$$

for some $a, b \in \mathbb{R}^+$ with $a > b$ such that $\mu(S) > 0$ for all $S \subseteq [d]$.

To better understand this formula, a critical question needs to be answered: What kind of a, b makes $\mu(S) > 0$ for all $S \subseteq [d]$? To answer (1), we show that a simple condition $1 \ge a > b \ge a^2 > 0$ suffices to make $\mu(S) > 0$ for all $S \subseteq [d]$.

Proposition 14. *When $a, b \in \mathbb{R}^+$ such that $1 \ge a > b \ge a^2 > 0$, we have*

$$\sum_{i=|S|}^{d} \binom{d-|S|}{i-|S|}(-1)^{i-|S|} g(a,b,i) > 0 \quad \text{for all } S \subseteq [d],$$

where $g(a,b,i)$ is defined in Eqn.(7.12).

We note that it is only a sufficient condition for selecting a and b: For some small $d \in \mathbb{N}$, we may have some a, b such that $1 > a^2 > b > 0$ but makes $\mu(S) > 0$ for all $S \subseteq [d]$. However, if $a = \bar{\mu}_1$ and $b = \bar{\mu}_2$ need to make the weighting function positive for all $d \in \mathbb{N}$, we must have the condition $1 \ge a > b \ge a^2 > 0$.

Faith-Banzhaf Interaction Index: As a first application of this theorem, suppose in addition to the three axioms above, we additionally require the Faith-Interaction indices to satisfy generalized 2-efficiency. The following theorem shows that there is a unique Faith-Interaction index satisfying these four axioms, which we term the Faith-Banzhaf index.

Theorem 19. *(Faith-Banzhaf) For any $d \ge 3$, there is a unique Faith-Interaction index that satisfies the interaction linearity, symmetry, dummy and generalized 2-efficiency axioms, with its coalition weighting function given as $\mu(S) \propto \frac{1}{2^d}$ for all $S \subseteq [d]$. We term this unique interaction index as **Faithful Banzhaf Interaction index** (Faith-Banzhaf), and which has the form:*

$$\Phi_S^{F\text{-}Bzf}(\mathbf{f}, \ell) = a(\mathbf{f}, S) + (-1)^{\ell-|S|} \sum_{T \supseteq S, |T|>\ell} \left(\frac{1}{2}\right)^{|T|-|S|} \binom{|T|-|S|-1}{\ell-|S|} a(\mathbf{f}, T), \forall S \in \mathcal{S}_\ell, \tag{7.13}$$

where $a(\mathbf{f}, \cdot)$ is the Möbius transform of $\mathbf{f}(\cdot)$. Moreover, its highest order interaction terms coincides with corresponding interaction terms from the Banzhaf interaction index introduced earlier:

$$\mathcal{E}_S^{F\text{-}Bzf}(\mathbf{f}, \ell) = \sum_{T \subseteq [d] \setminus S} \frac{1}{2^{d-|S|}} \Delta_S(\mathbf{f}(T)) \quad \textit{for all } S \in \mathcal{S}_\ell \textit{ with } |S| = \ell. \tag{7.14}$$

Our derivation of Faith-Banzhaf indices follows the pseudo-Boolean function approximation results from Grabisch et al. [61].

Faith-Shapley Interaction Index: When moving from generalized 2-efficiency to the more natural interaction efficiency axiom, we have the following proposition.

Proposition 15. *Faith-Interaction indices satisfy the interaction efficiency axiom if and only if the weighting functions satisfy $\mu(\emptyset) = \mu([d]) = \infty$.*

That the condition in the proposition is sufficient is a straight-forward consequence of the fact that $\mu(\emptyset) = \mu([d]) = \infty$ entails that the corresponding linear constraint be exactly satisfied, so that: $\sum_{S \in \mathcal{S}_\ell} \Phi_S(\mathbf{f}, \ell) = \mathbf{f}([d])$ and $\Phi_\emptyset(\mathbf{f}, \ell) = \mathbf{f}(\emptyset)$, which is precisely the interaction efficiency axiom. We now have the machinery to present our main result on the unique Faith-Interaction index that satisfies the four (interaction counterparts of the) standard axioms that the singleton Shapley value satisfies.

Theorem 20. *(Faith-Shap) There is a unique Faith-Interaction index that satisfies the interaction linearity, symmetry, dummy and efficiency axioms, with its coalition weighting function given as:*

$$\mu(S) \propto \frac{d-1}{\binom{d}{|S|} |S| (d - |S|)} \textit{ for all } S \subseteq [d] \textit{ with } 1 \leq |S| \leq d-1, \textit{ and } \mu(\emptyset) = \mu([d]) = \infty. \tag{7.15}$$

*We term this unique interaction index as the **Faithful Shapley Interaction index** (Faith-Shap), and which has the form:*

$$\Phi_S^{F\text{-}Shap}(\mathbf{f}, \ell) = a(\mathbf{f}, S) + (-1)^{\ell - |S|} \frac{|S|}{\ell + |S|} \binom{\ell}{|S|} \sum_{T \supset S, |T| > \ell} \frac{\binom{|T|-1}{\ell}}{\binom{|T|+\ell-1}{\ell+|S|}} a(\mathbf{f}, T), \quad \forall S \in \mathcal{S}_\ell, \tag{7.16}$$

where $a(\mathbf{f}, \cdot)$ is the Möbius transform of $\mathbf{f}(\cdot)$. Moreover, its highest order interaction terms can be expressed as a weighted average of discrete derivatives:

$$\mathcal{E}_S^{F\text{-}Shap}(\mathbf{f}, \ell) = \frac{(2\ell-1)!}{((\ell-1)!)^2} \sum_{T \subseteq [d] \setminus S} \frac{(\ell + |T| - 1)!(d - |T| - 1)!}{(d + \ell - 1)!} \Delta_S(\mathbf{f}(T)) \quad \textit{for all } S \in \mathcal{S}_\ell \textit{ with } |S| = \ell. \tag{7.17}$$

When the maximum interaction order $\ell = 1$, so that we only require singleton feature contributions, the explanation coincides with the classical singleton Shapley values. Thus for larger orders with $\ell > 1$, Faith-Shap can be seen to be a "natural" generalization of the first-order Shapley value. Note that the set of axioms it satisfies are (interaction extensions of) the classical linearity, symmetry, dummy and efficiency axioms. As noted before in an interaction context these axioms alone do not uniquely specify an interaction index. In contrast to the less intuitive

axioms such as recursive and interaction distribution axioms, we merely require an interaction extension of the faithfulness property of singleton Shapley values: that the interaction Shapley values approximate the given set value function for all possible subsets.

7.4 Contrasting Faith-Interaction with other Interaction Indices

In this section, we compare our Faith-Interaction indices, specifically Faith-Shap, with the other interaction indices introduced earlier.

Comparison with Shapley Interaction and Banzhaf Interaction Indices: As noted earlier, the Shapley interaction and Banzhaf interaction indices do not satisfy the interaction efficiency axiom, that states that the sum of interaction weights should equal the difference between the value function evaluated over the complete and empty sets. A critical advantage of the interaction efficiency axiom is that it forces the interaction index to distribute a fixed contribution (difference between the value function evaluated over the complete and empty sets) among the different interactions; without such a distributive requirement, the resulting weights can become quite non-intuitive. For instances of such non-intuitive behaviors, we refer to Sundararajan et al. [158], who provided many simple examples where the sum of Shapley interaction values over all subsets diverges as the number of features increases, even when the value function is bounded and $\mathbf{f}([d]) = 1$. Another caveat with these two interaction indices is that they are not cognizant of the maximum interaction order, and hence we cannot compute Shapley values that differ with varying maximum interaction orders.

Comparison with Shapley Taylor index: The Shapley Taylor index does satisfy the four axioms of interaction linearity, symmetry, dummy and efficiency. However, as noted earlier these four axioms do not uniquely determine interaction indices. The fifth axiom Shapley Taylor index then imposes for uniqueness is the interaction distribution axiom, which has caveats of imbalanced distributions of values to coalitions of different orders, namely, inordinately favoring the maximum interaction order. In particular, the interaction distribution axiom states that higher than max order interaction values (order $> \ell$) be distributed to the max-order interactions (order $= \ell$), but these max-order terms end up unable to solely explain all higher order interactions. On the other hand, it entails lower than max order interactions (order $< \ell$) do not take into account sub-coalitions other than the empty set, which can be contrasted for instance with singleton Shapley value that explicitly takes into account even higher order coalitions that contain the single feature. Thus the interaction distribution has the consequence of making both lower and max order interactions less faithful to the model.

In contrast, in our Faith-Interaction indices, even lower-order interaction weights take into account all possible coalitions, and where the weights are balanced so that the overall set of interaction indices optimally approximate the behavior of the underlying value function.

7.4.1 Examples

Example 1: We illustrate the difference between these interaction indices using a function with diminishing marginal utility. Consider the following value function with 11 features:

$$\mathbf{f}(S) = \begin{cases} 0 & \text{, if } |S| \leq 1. \\ |S| - p \times \binom{|S|}{2} & \text{, otherwise.} \end{cases} \tag{7.18}$$

This function represents the payoff when any subset of 11 people work on a task. Each person contributes 1 unit to the overall payoff, and the task requires at least 2 people. However, the marginal utility is diminishing in nature, since any two people also have a probability of p of being non-cooperative. Given this payoff function, it is worth reflecting that what the attributions to individuals should be. While it might seem that zero is a good value since at least two people are needed for the task, this attribution would only correspond to the *marginal contribution* of an individual player i.e. how much a player would contribute when they are by themselves. Whereas we would like our attributions to also take into account larger coalitions, and marginal contributions to such larger coalitions: this is one of the motivations for considering coalitional game-theoretic indices. Once we do so, then it can be seen that an individual effect of one is much more reasonable. Similarly, we would expect that the pairwise interaction effects be close to $-p$.

In Table 7.1, we list the values for different interaction indices for $p = 0.1, 0.2$. When the maximum interaction order $\ell = 1$, all indices are similar since their restrictions to singleton are the Shapley/Banzhaf values. When the maximum interaction order $\ell = 2$, our Faith-Shap accurately captures individual contribution and pairwise interaction effects by assigning $0.95/0.95$ and $-0.091/-0.191$ for order 1 and 2 and for $p = 0.1/0.2$ respectively, which are very close to the intuitions we outlined earlier. However, the Shapley Taylor index assigns the individual effect of i by using the marginal $\mathbf{f}(\{i\}) - \mathbf{f}(\emptyset)$, which can be highly inaccurate since such a marginal contribution does not take into account marginal contributions to larger coalitions.

For $p = 0.1$, Shapley Taylor along with Interaction Shapley assign a positive/zero value to interaction effect, which suggests that forming groups have complimentary/no effects. On the contrary, Banzhaf interaction and Faith-Banzhaf give negative values for interaction between players, which correctly reflects the decrease in marginal utility of this game.

For $p = 0.2$, the Shapley Taylor index is uniformly zero for any order. This highlights the other drawback of the Shapley Taylor index: the impoverished lower-order interaction indices make the max-order indices less faithful to the model. Specifically, for $p = 0.2$, and with $d = 11$ players, we can see that $\mathbf{f}([d]) = \mathbf{f}(\emptyset) = 0$. We have already seen that $\Phi_{\{i\}}^{\text{Taylor}}(\mathbf{f}, \ell) = 0$, for $i \in [d]$. For $\ell = 2$, we then have that the summation of the max-order (i.e. order two) indices equals $\mathbf{f}([d]) - \mathbf{f}(\emptyset) - \sum_{i=1}^{d} \Phi_{\{i\}}^{\text{Taylor}}(\mathbf{f}, \ell) = 0$ by the efficiency axiom. Since all max-order indices have the same value by the symmetry axiom, the max-order indices are uniformly zero. In this case, the Shapley Taylor indices do not take into account the function values $\mathbf{f}(S)$ with $\ell \leq |S| < d$, and can be arbitrarily unfaithful to these orders. Here, the Banzhaf interaction and Faith-Banzhaf again correctly reflect the negative interaction between players. However, the Banzhaf interaction value gives a value close to 0 for the first-order indices. Taken together with its negative interaction effects, it might seem that coalitions can only be hurtful to the payoff, which is misleading since the total utility is positive when 2 to 10 players are present. On the other

hand, our Faith-Banzhaf gives a positive value close to 1 for individual effects of order 1. Taken together with its negative interaction effects, the value given by the Faith-Banzhaf seems more intuitive: each single player contributes to the utility, while each pair of players hurts the utility.

Another instructive viewpoint for interaction values is by inspecting their utility for approximating the overall payoff function. In Figure 7.1 and 7.2, we approximate the function $\mathbf{f}(S)$ using $\sum_{T\subseteq S,|T|\leq 2}\mathcal{E}_T(\mathbf{f},\ell)$ for different interaction indices. We can see that our Faith-Shap/Faith-Banzhaf are (almost) faithful to all orders except for $|S|=1$. However, the Shapley Taylor index is only fully faithful to the model when the order is $0,1,11$, and curves for other interaction indices are unfaithful.

Indices	$p=0.1$			$p=0.2$		
	$\ell=1$	$\ell=2$		$\ell=1$	$\ell=2$	
	Order 1	Order 1	Order 2	Order 1	Order 1	Order 2
Faith-Shap	0.5	0.95	-0.091	0	0.95	-0.191
Shapley Taylor	0.5	0	0.1	0	0	0
Interaction Shapley	0.5	0.5	0	0	0	-0.1
Banzhaf Interaction	0.51	0.51	-0.113	0.009	0.009	-0.213
Faith-Banzhaf	0.51	1.08	-0.113	0.009	1.08	-0.213

Table 7.1: Values for different interaction indices of different orders for $p=0.1,0.2$ with different maximum interaction orders. Note that $\Phi_{\emptyset}^{\text{F-Shap}}(\mathbf{f},\ell)=0$ and $\Phi_{\emptyset}^{\text{F-Bzf}}(\mathbf{f},\ell)=-0.24$ for both $p=0.1$ and $p=0.2$.

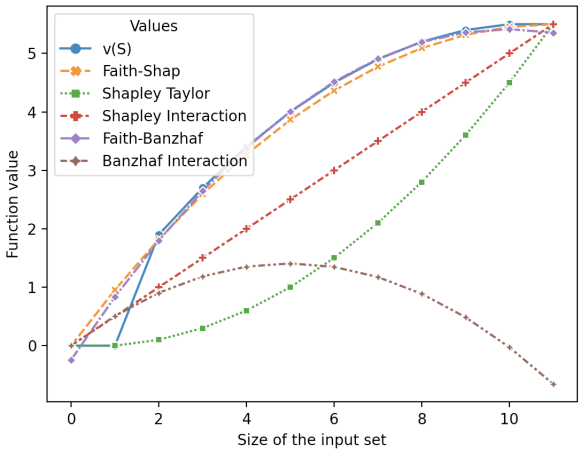


Figure 7.1: Function approximation of Eqn.(7.18) using different interaction indices for $p=0.1$ with the maximum interaction order $\ell=2$.

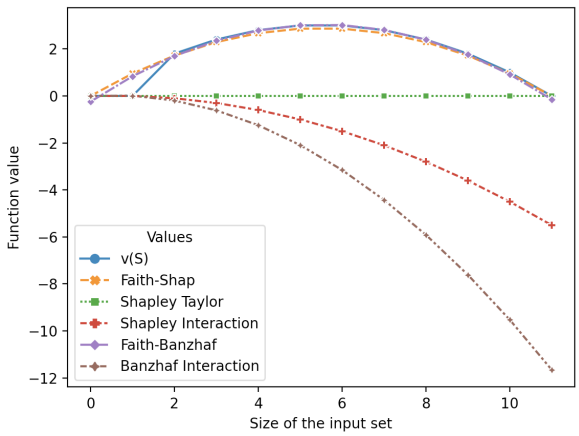


Figure 7.2: Function approximation of Eqn.(7.18) using different interaction indices for $p=0.2$ with the maximum interaction order $\ell=2$.

Example 2: We provide another example, this time with increasing marginal utility. Consider a family who are in the wind energy business, with $d=11$ family members. Currently, the family

owns 1 wind turbine, and they can get 3 units of revenue per wind turbine they own. Now, each family member is considering whether to manage a wind turbine. To build x wind turbines, the cost is described by the function $\text{cost}(x) = x + 2\log(x+1)$, as they may get a discount from the constructor to build more wind turbines at the same time. If exactly one member chooses to manage a wind turbine, the building cost will be 0 since the family already owns one wind turbine. The total revenue for the family when S is the set of members that participate in building new wind turbines can be described by the following function:

$$\mathbf{f}(S) = \begin{cases} 0 & \text{, if } |S| = 0. \\ 3 & \text{, if } |S| \leq 1. \\ 3|S| - (|S| - 2\log(|S|+1)) & \text{, if } 2 \leq |S| \leq 11. \end{cases} \tag{7.19}$$

This function has an increasing marginal utility since the marginal cost is decreasing. Therefore, we would expect the interaction effect to be positive. However, from Table 7.2, only Faith-Shap, Faith-Banzhaf and Banzhaf interaction indices capture this effect.

Moreover, the Faith-Shap and Faith-Banzhaf indices have the following intuitive interpretation: Having one more member joining the family business increases the total revenue by 1.20/1.19 unit, with 0.07/0.09 additional unit of revenue when two members join together since they are cooperative. In contrast, we can not interpret the Banzhaf interaction index for orders 1 and 2 jointly since it is not cognizant of the maximum interaction order ℓ.

Indices	$\ell = 1$	$\ell = 2$	
	Order 1	Order 1	Order 2
Faith-Shap	1.55	1.20	0.07
Shapley Taylor	1.55	3	-0.29
Shapley Interaction	1.55	1.55	-0.12
Faith-Banzhaf	1.65	1.19	0.09
Banzhaf Interaction	1.65	1.65	0.09

Table 7.2: Values for different interaction indices of different orders with the maximum interaction order $\ell = 2$. Note that $\Phi_{\emptyset}^{\text{F-Shap}}(\mathbf{f}, \ell) = 0$ and $\Phi_{\emptyset}^{\text{F-Bzf}}(\mathbf{f}, \ell) = 0.48$ for the indices corresponding to empty sets.

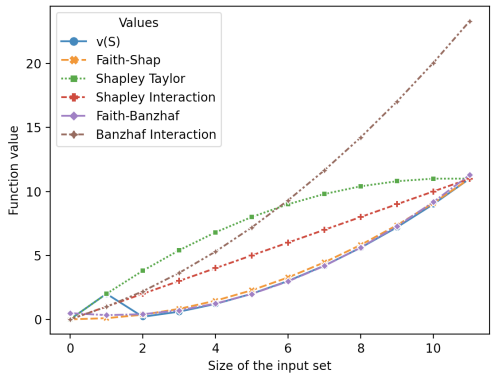


Figure 7.3: Function approximation of Eqn.(7.19) using different interaction indices with the maximum interaction order $\ell = 2$.

7.5 Algebraic Properties of Faith-Interaction Indices

In the following two sub-sections, we discuss how Faith-Shap can be represented as a cardinal index, as well as through the lens of a multilinear approximation.

7.5.1 Cardinal Indices

Grabisch and Roubens [59] show that any interaction index (they only consider the classical case with maximum interaction order $\ell = d$) that satisfies the linearity, dummy, and symmetry axioms necessarily has the following form:

$$\Phi_S(\mathbf{f}, d) = \sum_{T \subseteq [d] \setminus S} p_{|T|}^{|S|} \Delta_S \mathbf{f}(T), \quad \forall S \subseteq [d], \tag{7.20}$$

and for some family of constants $\{p_t^s\}_{s \in [0:d], t \in [0:d-s]}$. They term this class of interaction indices as *cardinal* interaction indices. Of course this is a large class, and it is not apriori clear how to further constrain the indices so as to get specific values for the constants $\{p_t^s\}$. We remark in passing that Shapley and Banzhaf interaction indices impose additional structure on the constants $\{p_t^s\}$.

We can also consider the class of probabilistic interaction indices:

$$\Phi_S(\mathbf{f}, d) = \sum_{T \subseteq [d] \setminus S} p_T^S \Delta_S v(T),$$

where for any $S \subseteq [d]$, the constants $\{p_T^S\}_{T \subseteq [d] \setminus S}$ form a probability distribution on $[d] \setminus S$. We can then define cardinal-probabilistic indices as those indices that are both cardinal and probabilistic interaction indices, so that $p_T^S = p_{|T|}^{|S|}$, for some family of constants $\{p_t^s\}_{s \in [1:d], t \in [0:d-s]}$ that satisfy:

$$\sum_{t=0}^{d-s} \binom{d-s}{t} p_t^s = 1.$$

Fujimoto et al. [47] shows that indices that satisfy certain additivity, monotonicity, symmetry, and dummy partnership axioms are necessarily cardinal probabilistic indices. As Fujimoto et al. [47] shows, Shapley and Banzhaf interaction indices do fall into this class.

One could of course naturally extend these notions of cardinal, probabilistic and cardinal-probabilistic indices to be cognizant of the maximum interaction order $\ell \in [d]$. It is an interesting open question to investigate extensions of results of Fujimoto et al. [47] to such a sub-class of cardinal-probabilistic indices cognizant of the max-interaction order. In this section, we provide a modest initial result along these lines, focusing on the top interaction level of the interaction index.

Proposition 16. *For any maximum interaction order $1 \leq \ell \leq d$, and for any set value function $\mathbf{f} : 2^d \mapsto \mathbb{R}$, the top level of the Faithful Shapley Interaction index can be expressed as a cardinal-probabilistic index:*

$$\Phi_S^{F\text{-}Shap}(\mathbf{f}, \ell) = \sum_{T \subseteq [d] \setminus S} p_{|T|}^{\ell} \Delta_S(\mathbf{f}(T)), \quad \forall S \subseteq [d] \text{ with } |S| = \ell, \tag{7.21}$$

where $p_t^\ell = \frac{(2\ell-1)!(\ell+t-1)!(d-t-1)!}{((\ell-1)!)^2(d+\ell-1)!}$. Moreover, it satisfies $\sum_{t=0}^{d-\ell} \binom{d-\ell}{t} p_t^\ell = 1$.

Therefore, the top level of the Faithful Shapley Interaction index captures the interactions of features in S *in the presence of all subsets* $T \subseteq [d] \setminus S$.

7.5.2 Multilinear Formulation

Any set value function $\mathbf{f} : 2^{[d]} \mapsto \mathbb{R}$ has a unique multi-linear extension $g : [0,1]^d \mapsto \mathbb{R}$, also referred to the *Owen multilinear extension* [121], given as:

$$g(x) := \sum_{T \subseteq [d]} \mathbf{f}(T) \prod_{i \in T} x_i \prod_{i \notin T} (1 - x_i), \quad \forall x \in [0,1]^d.$$

For any set $S \subseteq [d]$, with $S = \{i_1, \ldots, i_s\}$, denote its S-derivative as $\Delta_S g(x) := \frac{\partial^s g(x)}{\partial x_{i_1} \ldots \partial x_{i_s}}$.

7.5.2.1 Path Integrals

Grabisch et al. [61] show that Shapley interaction index can be written as:

$$\Phi_S^{\text{Shap}}(\mathbf{f}, d) = \int_{x=0}^{1} \Delta_S g(x, \ldots, x) dx, \ \forall S \subseteq [d].$$

That is, we can obtain the Shapley interaction index by integrating the S-derivative along the diagonal of the unit hypercube.

On the other hand, the Banzhaf interaction index can be written as:

$$\Phi_S^{\text{Bzf}}(\mathbf{f}, d) = \int_{x \in [0,1]^d} \Delta_S g(x) dx, \ \forall S \in \mathcal{S}_d.$$

That is, we can obtain the Banzhaf interaction index by integrating the S-derivative over the entire unit hypercube. In this case, it also has the closed form: $\Delta_S g(1/2, \ldots, 1/2)$.

Fujimoto et al. [47] show that any cardinal probabilistic index Φ has the form:

$$\Phi_S(\mathbf{f}, d) = \int_{x=0}^{1} \Delta_S g(x, \ldots, x) dF_{|S|}(x), \ \forall S \in \mathcal{S}_d,$$

for some family of CDFs $\{F_s\}_{s \in [d]}$. That is, we can obtain any cardinal probabilistic index by integrating the S-derivative along the diagonal of the unit hypercube with respect to some distribution over $[0,1]$.

It is an interesting open question whether we could extend these results from Grabisch et al. [61] and Fujimoto et al. [47] to interaction indices that are cognizant of the maximum interaction order $\ell \in [d]$. In this section, we provide a modest initial result along these lines, focusing on the top interaction level of the interaction index.

Proposition 17. *For any maximum interaction order $1 \leq \ell \leq d$, and for any set function $\mathbf{f} : 2^d \mapsto \mathbb{R}$, the top level of the Faithful Shapley Interaction index value can be expressed as:*

$$\Phi_S^{\text{F-Shap}}(\mathbf{f}, \ell) = \int_{x=0}^{1} \Delta_S g(x, \cdots, x) dI_x(\ell, \ell), \ \forall S \in \mathcal{S}_\ell \text{ with } |S| = \ell, \tag{7.22}$$

where $I_x(\ell, \ell)$ is cumulative distribution function of the beta distribution $B(x; \ell, \ell)$.

7.5.2.2 Taylor Expansion

In contrast to path integrals, Sundararajan et al. [158] use the Taylor expansion of $g(\mathbf{1}) = \mathbf{f}([d])$ around $g(\mathbf{0}) = \mathbf{f}(\emptyset)$ Taylor derivations to derive their interaction index. Specifically, they show that Shapley Taylor index $\Phi_S^{\text{Taylor}}(\mathbf{f}, \ell)$ is equal to the $|S|^{\text{th}}$ term of the $(\ell-1)^{\text{th}}$ order Taylor expansion of $g(\cdot)$ with Lagrange remainder:

$$\begin{aligned} g(\mathbf{1}) &= \sum_{j=0}^{\ell-1} \frac{g^{(j)}(\mathbf{0})}{j!} g(\mathbf{0}) + \int_{x=0}^{1} \frac{(1-x)^{\ell-1}}{(\ell-1)!} g^{(\ell)}(x, \cdots, x) dx \\ &= \sum_{j=0}^{\ell-1} \sum_{|S|=j} \Delta_S g(\mathbf{0}) + \sum_{|S|=\ell} \int_{x=0}^{1} \ell(1-x)^{\ell-1} \Delta_S g(x, \cdots, x) dx \\ &\quad \text{[158, Theorem 3]} \\ &= \sum_{j=0}^{\ell-1} \sum_{|S|=j} \Phi_S^{\text{Taylor}}(\mathbf{f}, \ell) + \sum_{|S|=\ell} \Phi_S^{\text{Taylor}}(\mathbf{f}, \ell), \end{aligned}$$

where $g^{(j)}(x)$ is the j^{th} derivative of the function $g(x, \cdots, x)$, $\Phi_S^{\text{Taylor}}(\mathbf{f}, \ell) = \Delta_S g(\mathbf{0})$ for $|S| < \ell$ and $\Phi_S^{\text{Taylor}}(\mathbf{f}, \ell) = \int_{x=0}^{1} \ell(1-x)^{\ell-1} \Delta_S g(x, \cdots, x) dx$ with $|S| = \ell$. This can be seen to result in impoverished lower order subset interactions, which now no longer take into account higher order coalitions that include that subset.

7.5.2.3 Pseudo-Boolean Function Approximation

While we have so far discussed the continuous multi-linear extension of a set value function $\mathbf{f} : 2^{[d]} \mapsto \mathbb{R}$, we can also simply consider its equivalent pseudo-Boolean counterpart $g \in \mathcal{F}$ with $\mathcal{F} = \{g : \{0,1\}^d \mapsto \mathbb{R}\}$:

$$g(x) := \sum_{T \subseteq [d]} \mathbf{f}(T) \prod_{i \in T} x_i \prod_{i \notin T} (1 - x_i), \quad \forall x \in \{0,1\}^d.$$

One can also derive the pseudo-Boolean function g_Φ corresponding to interaction indices Φ, and ask for interaction indices with pseudo-Boolean counterparts g_Φ that best approximate the pseudo-Boolean counterpart g of the set value function. Specifically, given a maximum interaction order $\ell \in [d]$ and an interaction index $\Phi \in \mathbb{R}^{d_\ell}$, its pseudo-Boolean counterpart $g_\Phi \in \mathcal{F}$ is defined as:

$$g_\Phi(x) := \sum_{T \subseteq [d], |T| \leq \ell} \Phi_T(\mathbf{f}, \ell) \prod_{i \in T} x_i, \quad \forall x \in \{0,1\}^d.$$

Hammer and Holzman [67] and Grabisch et al. [61] consider solving for the best ℓ_2-norm approximation by the function $g_\Phi(\cdot)$ with degree up to ℓ. That is, $\|g - g_\Phi\|_2 = \sqrt{\sum_{x \in \{0,1\}^d} (g(x) - g_\Phi(x))^2}$. Using this perspective, we can see that Faith-Banzhaf interaction indices can in turn be related to such a function approximation:

$$\Phi^{\text{F-Bzf}}(\mathbf{f}, \ell) = \min_{\Phi \in \mathbb{R}^{d_\ell}} \|g(x) - g_\Phi(x)\|_2 = \min_{\Phi \in \mathbb{R}^{d_\ell}} \sum_{S \subseteq [d]} \left(\mathbf{f}(S) - \sum_{T \subseteq S, |T| \leq \ell} \Phi_T \right)^2,$$

and where the solution has the closed-form expression we detail in Theorem 18.

For the singleton attribution case, with max order $\ell = 1$, Ding et al. [36] and Ruiz et al. [133] consider μ-norm function approximations $\|g(x) - g_\Phi(x)\|_\mu = \sqrt{\sum_{x\in\{0,1\}^d} \mu(x)(g(x) - g_\Phi(x))^2}$, but where μ only depends on $\|x\|_1$, and where $\mu(\mathbf{0})$ and $\mu(\mathbf{1})$ can both be infinity. Ding et al. [36] provide a closed-form expression for $g_\Phi(x)$, while Ruiz et al. [133] analyze its axiomatic properties.

For the specific case where the probability of coalition S can be expressed as $\mu(x) = \prod_{i:x_i=1} p_i \prod_{j:x_j=0}(1-p_j)$ for some $0 < p_i < 1$ indicating the probability of the feature i being present, Ding et al. [37] and Marichal and Mathonet [109] considers solving the best ℓ^{th} order polynomial approximation under $\|\cdot\|_\mu$ norm.

In contrast to the above work, our developments could be cast as pseudo-Boolean approximations for the general weighted norm case $\|\cdot\|_\mu$, for general weighting functions $\mu(\cdot)$ without stringent structural assumptions, and while allow for arbitrary maximum interaction orders $\ell \in [d]$.

7.6 Experiments

We first provide some experiments validating the relative computational efficiency of computing our Faith-Interaction indices, followed by quantitative and qualitative demonstrations of their use as explanations of ML models over a language dataset.

The language dataset we use throughout the experiment is the simplified IMDB [107] dataset, where the model only uses the first two sentences of movie reviews as input, and predicts the probability of the reviews being positive. The model being explained is a BERT language model [34] with 0.82 accuracy on the test set.

7.6.1 Computational Efficiency

Exact computation of interaction indices that aggregate over all possible feature subsets exactly typically requires 2^d model evaluations (with d features) which is impractical in most machine learning applications. A key advantage of our Faith-Interaction indices, as compared to other recently proposed interaction indices such as the Shapley Taylor index and Shapley Interaction index, is that they can be computed by solving a weighted least squares problem. As we empirically show in this section, this enables us to provide more accurate estimates with fewer model evaluations, compared to the other recent approaches that employ permutation-based sampling methods.

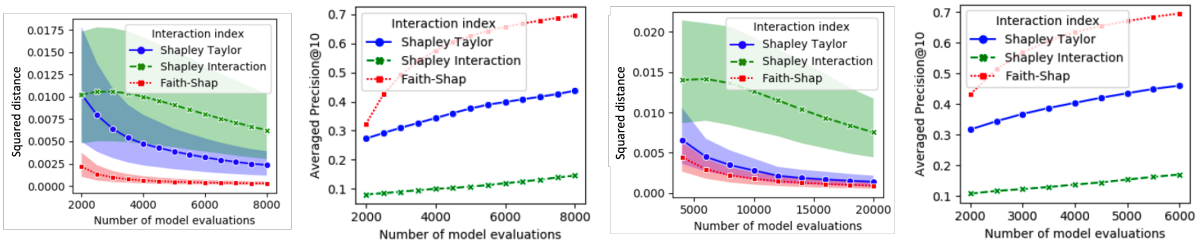
Setup: To demonstrate the computational efficiency of Faith-Interaction indices, we compare our proposed Faith-Shap with Shapley interaction and Shapley Taylor interaction indices using different estimation methods. For the Faith-Shap interaction index, we sample each coalition $S \in [d]$ with probability $\propto \frac{d-1}{\binom{d}{|S|}\,|S|\,(d-|S|)}$, and set $\mu(\emptyset)$ and $\mu([d])$ to a large number (in lieu of infinity). We then solve the corresponding linear regression problem with ℓ_1 regularization, and regularization parameter $\alpha = 10^{-6}$ and $\alpha = 10^{-4}$ for maximum interaction orders $\ell = 2$ and

$\ell = 3$ respectively. For the Shapley Taylor interaction and Shapley Interaction indices, we use the permutation-based sampling methods.

We compare these indices in two settings: (1) language data: we randomly choose 50 data with length $d = 15$ from simplified IMDB dataset and set $\ell = 3$. (2) sparse synthetic function: We parameterize the synthetic sparse function $\mathbf{f} : 2^d \to \mathbb{R}$ with $\sum_{i=1}^{N} a_i \prod_{j \in S_i} x_i$, where $S_1, S_2, \cdots, S_N$ are subsets of $[d]$ and $a_1, \cdots, a_N$ are coefficients. We set $d = 70$, $N = 30$, $\ell = 2$, sample each a_i uniformly over $[\frac{i}{10}, \frac{i}{10}]$, and each S_i uniformly over subsets of $|S|$ with sizes ≤ 5.

To measure how close is an interaction index to its ground-truth value, we use two evaluation metrics: (1) averaged squared distance, $\|\Phi - \Phi^{\text{est}}\|_2^2 / \binom{d}{\ell}$, and (2) precision at 10, which we measure the proportion of top-10 feature interactions (with respect to absolute value) in the top-10 ground-truth interactions as top interactions are more critical when these indices are used in XAI. We also note that we drop the lower-order indices and only compare top-order indices (order=ℓ) since computing lower-order Shapley Taylor indices are trivial. Each evaluation metric is reported by averaging 50 different inputs with 100/10 different random seeds for language data and sparse synthetic functions, respectively.

Results: From Figure 7.4, we see that Faith-Shap can be estimated more accurately and uses fewer model evaluations: in both language data and sparse settings, as well as in terms of all evaluation metrics.



(a) Averaged squared distance (language). (b) Precision@10 (language).] (c) Average squared distance (sparse) (d) Precision@10 (sparse).

Figure 7.4: Comparison of Faith-Shap, Shapley Taylor and Shapley interaction indices in terms of computational efficiency in language data and synthetic sparse functions. The shaded areas indicate the 5th and 95th percentiles.

7.6.2 Explanations on a Language Dataset

In this section, we use our Faith-Shap interaction index to explain a deep-learning model on the simplified IMDB dataset. We set the maximum interaction order $\ell = 2$. Table 7.3 shows some of the interesting interactions we found.

In the first two examples, we see non-complementary interaction effects. In the first example, while the importance values of the individual words "Never" and "forgot" are negative (as shown in Tables 7.5, 7.6), their joint effect as shown in the table here is extremely positive. Similarly, for the second, the words "only" and "good" are individually positive, while their joint effect

Index	Sentences (bold words are the interactions with the highest (absolute) importance values)	Model Prediction	Interaction score
1	I have **Never forgot** this movie. All these years and it has remained in my life.	Positive	0.818
2	TWINS EFFECT is a poor film in so many respects. The **only good** element is that it doesn't take itself seriously..	Negative	-0.375
3	I rented this movie to get an easy, entertained view of the history of Texas. I got a **headache instead**.	Negative	0.396
4	Truly **appalling waste** of space. Me and my friend tried to watch this film to its conclusion but had to switch it off about 30 minutes from the end.	Negative	0.357
5	I still remember watching Satya for the first time. I was completely **blown away**.	Positive	0.283

Table 7.3: Top interactions of different examples on IMDB.

is strongly negative. The fourth and fifth examples show more subtle non-complementarity effects. In the fourth example, while the individual words "headache" and "instead" have negative importance scores, their joint effect is positive, since the total effect of the phrase is less than the sum of the individual importance of these two words. The last example shows the effect of complementarity: words in a phrase are only meaningful when all words are present, and hence have a positive interaction effect. In Tables 7.4, 7.5, 7.6, we further show the top-15 important interactions and compare them to those from the Shapley Taylor index. We find that the first-order Shapley Taylor index yields meaningless interactions since they are the difference of predicted probabilities of a sentence containing only one word and an empty sentence (a baseline). They are all nearly zero since both the empty sentence and the sentence containing only one word are meaningless. This is another consequence of impoverished lower-order values of Shapley Taylor indices.

7.7 Related work

In cooperative game theory, a set function $\mathbf{f}(\cdot)$ with $\mathbf{f}(\emptyset) = 0$ corresponds to a transferable utility game (TU-game), and a set function with order $\leq \ell$ is called an ℓ-additive TU-game [62]. Therefore, our approach can be viewed as a least squares approximation of a TU-game by an ℓ-additive TU-game; see for instance Eqn. (7.10). Variants and special cases of this least squares approximation problem have been studied in the cooperative game theory field. For $\ell = 1$, Charnes et al. [22] first give general solutions when the weighting function is symmetric and positive, and show that the Shapley value results from a particular choice of the weighting function. Ruiz et al. [132, 133] consider the same setting, and study the axiomatic properties of the solutions of the least squares problems. Ding et al. [36] further generalize the previous results by considering the cases where some weights are allowed to be zero. For the case where maximum interaction order $\ell > 1$, Hammer and Holzman [67] and Grabisch et al. [61] solve the least squares problem when the weighting function is a constant, and show that the top-level coefficients coincide with those of the Banzhaf interaction indices of order ℓ. Ding et al. [37] and

Marichal and Mathonet [109] consider a certain weighted version of the problem, and propose weighted Banzhaf interaction indices. Grabisch and Rusinowska [60] consider the approximation problem under the constraints that both TU-games yield the same Shapley value. Marichal and Roubens [110] extend the Shapley value and propose the chaining interaction index whose definition is based on maximal chains of ordered sets. For more details on this line of work, see the recent book [62]. From the lens of TU-game approximation, our work could be viewed as allowing for general weighting functions $\mu(\cdot)$ without stringent structural assumptions, as well as arbitrary maximum interaction orders $\ell \in [d]$.

Feature interactions have also been investigated in the machine learning community. Lundberg et al. [106] quantify feature interactions in tree-based models using the Shapley interaction index. Tsang et al. [165] quantify the interaction within a feature group S via the marginal importance $\mathbf{f}(S) - \mathbf{f}(\emptyset)$. Cui et al. [29], Janizek et al. [78] explain pairwise interactions in neural networks, and Bayesian neural networks respectively via second-order derivatives. Singh et al. [145] build hierarchical explanations within a deep neural network (DNN) using hierarchical clustering of features. Tsang et al. [163] detect feature interactions by examining weight matrices of DNNs. Tsang et al. [164] disentangle complex feature interactions within DNNs by forcing the weights matrices to be block-diagonal. Molnar et al. [114] measures the strength of feature interaction via accumulated local effects (ALE).

7.8 Conclusion

Deriving unique interaction indices that satisfy the interaction extensions of the individual Shapley axioms has been a long-standing open problem. Existing approaches introduce additional less natural axioms, with some even sacrificing natural ones such as efficiency, in order to specify unique interaction indices. In this work, we take the alternate route of considering the family of what we term faithful interaction indices, which similar to individual Shapley values, aim to approximate the given set value function for all feature subsets. We show that when restricting to the class of faithful interaction indices, we obtain a unique interaction index that satisfies the interaction extensions of the individual Shapley axioms, and which we term the Faithful Shapley Interaction Index (Faith-Shap). We show the benefits of the faithful Shapley interaction index via specific games of interest where there is diminishing return and increasing return, and connect the Faith-Shap to cardinal probabilistic indices and multilinear approximations. Finally, we show that Faith-Shap is efficient to estimate thanks to its connection to weighted linear regression in sparse settings, and provide some qualitative results for their use as explanations of machine learning models on a real language dataset.

Acknowledgements

The authors would like to thank Michel Grabisch and Hung-Hsun Yu for their generous feedback and assistance.

Index	Sentences			Predicted Prob.
1	I have Never forgot this movie. All these years and it has remained in my life.			0.992
	Faithful Shapley indices		*Shapley Taylor indices*	
	Feature (interactions)	Scores	Feature (interactions)	Scores
	Never, forgot	0.818	Never, forgot	1.077
	life	0.383	Never, life	-0.211
	forgot	-0.254	remained, movie	-0.177
	and	0.168	Never, this	-0.160
	it	0.168	forgot, life	-0.149
	Never	-0.163	and, forgot	-0.149
	years	0.156	in, life	-0.143
	All	0.132	Never, it	-0.122
	my	0.126	Never, movie	-0.114
	has	0.120	have, Never	-0.110
	have	0.112	I, have	0.106
	Never, life	-0.106	forgot, in	-0.105
	forgot, it	-0.096	Never, All	-0.104
	my, life	-0.086	years, life	-0.101
	this	0.081	it, forgot	-0.101

Index	Sentences			Predicted Prob.
2	TWINS EFFECT is a poor film in so many respects. The only good element is that it doesn't take itself seriously.			0.012
	Faithful Shapley indices		*Shapley Taylor indices*	
	Feature (interactions)	Scores	Feature (interactions)	Scores
	poor	-0.341	only, good	-0.450
	respects	0.297	EFFECT, good	-0.182
	only, good	-0.243	good, is	-0.171
	poor, only	0.206	poor, film	-0.169
	good	0.176	only, element	-0.168
	poor, respects	-0.173	doesn't, poor	0.151
	doesn't	-0.169	only, poor	0.150
	poor, good	0.122	respects, poor	-0.149
	only, doesn't	0.115	itself, poor	-0.142
	poor, doesn't	0.111	respects, good	-0.137
	many	0.095	it, doesn't	-0.108
	it	0.084	it, only	-0.098
	itself	0.083	take, seriously	0.095
	element	0.076	doesn't, good	-0.094
	poor, many	-0.070	doesn't, only	0.093

Table 7.4: Top-15 important feature (interactions) for Faithful Shapley indices and Taylor Shapley indices for language dataset. The predicted probability is the out probability of the sentence having positive sentiment.

Index	Sentences			Predicted Prob.
3	I rented this movie to get an easy, entertained view of the history of Texas. I got a headache instead.			0.026
	Faithful Shapley indices		*Shapley Taylor indices*	
	Feature (interactions)	Scores	Feature (interactions)	Scores
	instead	-0.321	headache, instead	0.268
	headache, instead	0.252	view, instead	-0.178
	headache	-0.205	headache, Texas	-0.139
	easy,	0.158	rented, instead	0.137
	view	0.130	instead, easy,	-0.125
	history	0.123	got, headache	-0.118
	rented	-0.122	entertained, instead	-0.115
	Texas	0.101	rented, headache	0.109
	entertained	0.095	got, easy,	-0.108
	rented, instead	0.085	got, history	-0.105
	Texas, headache	-0.069	a, I	-0.100
	history, instead	-0.064	view, history	0.100
	the	0.059	got, rented	-0.100
	entertained, instead	-0.057	got, a	-0.099
	this	0.052	history, an	0.094

Index	Sentences			Predicted Prob.
4	Truly appalling waste of space. Me and my friend tried to watch this film to its conclusion but had to switch it off about 30 minutes from the end.			0.002
	Faithful Shapley indices		*Shapley Taylor indices*	
	Feature (interactions)	Scores	Feature (interactions)	Scores
	waste	-0.345	appalling, waste	0.298
	appalling, waste	0.257	Truly, waste	-0.296
	appalling	-0.251	switch, it	-0.248
	Truly	0.169	tried, waste	0.230
	waste, tried	0.167	but, watch	-0.210
	friend	0.162	friend, waste	-0.184
	space	0.149	friend, tried	-0.172
	tried	-0.134	friend, but	-0.169
	Truly, waste	-0.118	Truly, but	-0.145
	watch	0.087	but, waste	0.145
	off	-0.086	waste, watch	-0.140
	and	0.078	waste, off	0.138
	waste, friend	-0.074	had, space	-0.128
	waste, space	-0.058	Truly, film	0.126
	of	0.055	30, waste	0.124

Table 7.5: Top-15 important feature (interactions) for Faithful Shapley indices and Taylor Shapley indices for language dataset. The predicted probability is the out probability of the sentence having positive sentiment.

Index	Sentences			Predicted Prob.
5	I still remember watching Satya for the first time. I was completely blown away.			0.994
	Faithful Shapley indices		*Shapley Taylor indices*	
	Feature (interactions)	Scores	Feature (interactions)	Scores
	remember	0.337	blown, away	0.345
	blown, away	0.293	the, first	0.191
	time	0.281	time, first	0.182
	Satya	0.208	watching, for	-0.169
	remember, blown	-0.158	time, away	-0.167
	watching	0.153	time, Satya	-0.151
	blown	0.146	time, still	-0.145
	time, away	-0.127	still, watching	-0.144
	completely, away	-0.101	I, watching	-0.131
	Satya, time	-0.091	watching, first	-0.128
	remember, time	-0.073	remember, away	0.118
	I, watching	-0.071	Satya, away	-0.118
	completely, blown	0.063	was, watching	-0.115
	first, blown	-0.053	remember, blown	-0.110
	first	0.049	completely, away	-0.107

Table 7.6: Top-15 important feature (interactions) for Faithful Shapley indices and Taylor Shapley indices for language dataset. The predicted probability is the out probability of the sentence having positive sentiment.

Chapter 8

First is better than last for Language data Influence

Training data influence methods study the influence of training examples on a model's weights (learned during the training process), and in turn on the predictions of other test examples. They enable us to debug predictions by attributing them to the training examples that most influence them, debug training data by identifying mislabeled examples, and fixing mispredictions via training data curation. While the idea of training data influence originally stems from the study of linear regression [28], it has recently been developed for complex machine learning models like deep networks.

Prominent methods for quantifying training data influence for deep networks include influence functions [89], representer point selection [179], and TracIn [128]. While the details differ, all methods involves computing the gradients (w.r.t. the loss) of the model parameters at the training and test examples. Thus, they all face a common computational challenge of dealing with the large number of parameters in modern deep networks. In practice, this challenge is circumvented by restricting the study of influence to only the parameters in the last layer of the network. While this choice may not be explicitly stated, it is often implicit in the implementations of larger neural networks. In this work, we revisit the choice of restricting influence computation to the last layer in the context of large-scale Natural Language Processing (NLP) models.

We first introduce the phenomenon of "cancellation effect" of training data influence, which happens when the sum of the influence magnitude among different training examples is much larger than the actual influence sum. This effect causes most training examples to have a higher magnitude of influence score, and reduces the discriminative power of data influence. We also observe that different weight parameters may have different level of cancellation effects, and the weight parameters of bias parameters and latter layers may have larger cancellation effects. To mitigate the "cancellation effect" and find a scalable algorithm, we propose to operate data influence on weight parameters with the least cancellation effect – the first layer of weight parameter, which is also known as the word embedding layer.

While word embedding representations might have the issue of not capturing any high-level input semantics, we surprisingly find that the gradients of the embedding weights do not suffer from this. Since the gradient chain through the higher layers, it thus takes the high-level information captured in those layers into account. As a result, the gradients of the embedding

weights of a word depend on both the context and importance of the word in the input. We develop the idea of word embedding based influence in the context of TracIn due to its computational and resource efficiency over other methods. Our proposed method, TracIn-WE, can be expressed as the sum of word embedding gradient similarity over overlapping words between the training and test examples. Requiring overlapping words between the training and test sentences helps capture low-level similarity, while the word gradient similarity helps capture the high-level semantic similarity between the sentences. A key benefit of TracIn-WE is that it affords a natural word-level decomposition, which is not readily offered by existing methods. This helps us understand which words in the training example drive its influence on the test example.

We evaluate TracIn-WE on several NLP classification tasks, including toxicity, AGnews, and MNLI language inference with transformer models fine-tuned on the task. We show that TracIn-WE outperforms existing influence methods on the case deletion evaluation metric by $4-10\times$. A potential criticism of TracIn-WE is its reliance on word overlap between the training and test examples, which would prevent it from estimating influence between examples that relate semantically but not syntactically. To address this, we show that the presence of common tokens in the input, such as a "start" and "end" token (which are commonly found in modern NLP models), allows TracIn-WE to capture influence between semantically related examples without any overlapping words, and outperform last layer based influence methods on a restricted set of training examples that barely overlaps with the test example.

8.1 Preliminaries

Consider the standard supervised learning setting, with inputs $x \in \mathcal{X}$, outputs $y \in \mathcal{Y}$, and training data $\mathsf{D} = \{(x_1, y_1), (x_2, y_2), ...(x_n, y_n)\}$. Suppose we train a predictor $\mathbf{f}$ with parameter Θ by minimizing some given loss function ℓ over the training data, so that $\Theta = \arg\min_\Theta \sum_{i=1}^n \ell(\mathbf{f}(x_i), y_i)$. In the context of the trained model $\mathbf{f}$, and the training data D, we are interested in the data importance of a training point x to the testing point x', which we generally denote as $\delta(x, x')$.

8.1.1 Existing Methods

We first briefly introduce the commonly used training data influence methods: Influence functions [89], Representer Point selection [179], and TracIn [128]. We demonstrate that each method can be decomposed into a similarity term $S(x, x')$, which measures the similarity between a training point x and the test point x', and loss saliency terms $L(x)$ and $L(x')$, that measures the saliency of the model outputs to the model loss. The decomposition largely derives from an application of chain rule to the parameter gradients.

$$\delta(x, x') = L(x)S(x, x')L(x')$$

The decomposition yields the following interpretation. A training data x has a larger influence on a test point x' if (a) the training point model outputs have high loss saliency, (b) the training point x and the test point x' are similar as construed by the model. In Section 8.2.3,we show that restricting the influence method to operate on the weights in the last layer of the model critically affects the similarity term, and in turn the quality of influence. We now introduce the form of each method, and the corresponding similarity and loss saliency terms.

Influence Functions:

$$\text{Inf}(x, x') = -\nabla_\Theta \ell(x, \Theta)^T H_\Theta^{-1} \nabla_\Theta \ell(x', \Theta),$$

where H_Θ is the hessian $\sum_{i=1}^n \nabla_\Theta^2 \ell(x, \Theta)$ computed over the training examples. By an application of the chain rule, we can see that $\text{Inf}(x, x') = L(x)S(x, x')L(x')$, with the similarity term $S(x, x') = \frac{\partial \mathbf{f}(x,\Theta)}{\partial \Theta}^T H_\Theta^{-1} \frac{\partial \mathbf{f}(x',\Theta)}{\partial \Theta}$, and the loss saliency terms $L(x) = \frac{\partial \ell(x,\Theta)}{\partial \mathbf{f}(x,\Theta)}$. The work by Sui et al. [153] is very similar to extending the influence function to the last layer to satisfy the representer theorem.

Representer Points:

$$\text{Rep}(x, x') = -\frac{1}{2\lambda n} \frac{\partial \ell(x, \Theta)}{\partial \mathbf{f}_j(x, \Theta)} \sigma(x, \Theta)^T \sigma(x', \Theta), \tag{8.1}$$

where $\sigma(x, \Theta)$ is the final activation layer for the data point x, λ is the strength of the ℓ_2 regularizer used to optimize Θ, and j is the targeted class to explain. The similarity term is $S(x, x') = \sigma(x, \Theta)^T \sigma(x', \Theta)$, and the loss saliency terms are $L(x) = \frac{1}{2\lambda n} \frac{\partial \ell(x,\Theta)}{\partial \mathbf{f}_j(x,\Theta)}$, $L(x') = 1$.

TracIn:

$$\text{TracIn}(x, x') = -\sum_{c=1}^{d} \eta_c \nabla_{\Theta_c} \ell(x, \Theta_c)^T \nabla_{\Theta_c} \ell(x', \Theta_c), \tag{8.2}$$

where Θ_c is the weight at checkpoint c, and η_c is the learning rate at checkpoint c. In the remainder of the work, in our notation, we suppress the sum over checkpoints of TracIn for notational simplicity. (This is not to undermine the importance of summing over past checkpoints, which is a crucial component in the working on TracIn.) For TracIn, the similarity term is $S(x, x') = \nabla_\Theta \mathbf{f}(x, \Theta)^T \nabla_\Theta \mathbf{f}(x', \Theta)$, while the loss terms are $L(x) = \frac{\partial \ell(x,\Theta)}{\partial \mathbf{f}(x,\Theta)}$, $L(x') = \frac{\partial \ell(x',\Theta)}{\partial \mathbf{f}(x',\Theta)}$.

8.1.2 Evaluation: Case Deletion

We now discuss our primary evaluation metric, called *case deletion diagnostics* [28], which involves retraining the model after removing influential training examples and measuring the impact on the model. This evaluation metric helps validate the efficacy of any data influence method in detecting training examples to remove or modify for targeted fixing of misclassifications, which is the primary application we consider in this work. This evaluation metric was also noted as a key motivation for influence functions [89]. Given a test example x', when we remove training examples with positive influence on x' (*proponents*), we expect the prediction value for the ground-truth class of x' to decrease. On the other hand, when we remove training examples with negative influence on x' (*opponents*), we expect the prediction value for the ground-truth class of x' to increase.

An alternative evaluation metric is based on detecting mislabeled examples via self-influence (i.e. influence of a training sample on that same sample as a test point). We prefer the case deletion evaluation metric, as it more directly corresponds to the concept of data influence. Similar evaluations that measure the change of predictions of the model after a group of points is removed is seen in previous works. Han et al. [70] measures the test point prediction change after 10% training data with the most and least influence are removed, and Koh et al. [91] measures the

correlation of the model loss change after a group of trained data is removed and the sum of influences of samples in the group, where the group can be seen as manually defined clusters of data.

Deletion curve. Given a test example x' and influence measure δ, we define the metrics $\text{DEL}_+(x', k, \delta)$ and $\text{DEL}_-(x', k, \delta)$ as the impact on the prediction of x' (for its groundtruth class) upon removing top-k proponents and opponents of x' respectively:

$$\text{DEL}_+(x', k, \delta) = \mathsf{E}[f_c(x', \Theta_{+;k}) - f_c(x', \Theta)],$$

$$\text{DEL}_-(x', k, \delta) = \mathsf{E}[f_c(x', \Theta_{-;k}) - f_c(x', \Theta)],$$

where, $\Theta_{+;k}$ ($\Theta_{-;k}$) are the model weights learned when top-k proponents (opponents) according to influence measure I are removed from the training set, and c is the groundtruth class of x'. The expectation is over the number of retraining runs. We expect DEL_+ to have large negative, and DEL_- to have large positive values. To evaluate the deletion metric at different values of k, we may plot $\text{DEL}_+(x', k, \delta)$ and $\text{DEL}_-(x', k, \delta)$ for different values of k, and report the area under the curve (AUC): $\text{AUC-DEL}_+ = \sum_{k=k_1}^{k_m} \frac{1}{m} \text{DEL}_+(x', k, \delta)$, and $\text{AUC-DEL}_- = \sum_{k=k_1}^{k_m} \frac{1}{m} \text{DEL}_-(x', k, \delta)$.

We note that the *case deletion diagnostics* is different to the leave-one-out evaluation of Koh and Liang [89] by two points. First, leave-one-out evaluation focuses on removing one point, which is more meaningful in the convex regime where the optimization is initialization-invariant. We consider the leave-k-out evaluation which is closer to actual applications, as one may need to alter more than one training data to fix a prediction. Second, we consider the expected value of leave-k-out, to hedge the variance caused by specific model states, which was pointed out by Søgaard et al. [149] to be a major issue for leave-one-out evaluation (especially when the objective is no longer convex).

8.2 Cancellation Effect of Data Influence

The goal of a data influence method is to distribute the test data loss (prediction) across training examples, which can be seen as an attribution problem where each training example is an agent. We observe *cancellation* across the data influence attributions to training examples, i.e., the sign of attributions across training examples disagree and cancels each other out. This leads to most training examples having a large attribution magnitude, which reduces the discriminatory power of attribution-based explanations.

Our next observation is that the cancellation effect varies across different weight parameters. In particular, when a weight parameter is used by most of the training examples, the cancellation effect is especially severe. One such parameter is the bias, whose cancellation effect is illustrated by the following example:

Example 1. *Consider an example where the input $x \in \mathbb{R}^d$ is sparse, and x_i has feature i with value 1 and all other features with value 0. The prediction function has the form $\mathbf{f}(x) = x \cdot w + b$. It follows that a set of optimal parameters are $w_i = y_i, b = 0$. For a test point $x_t = x_0$, it is clear that x_0 should be the most influential training example contributing to reducing the loss of x_t, and w_0 should be the most influential weight parameter that contributes to it. However, all training examples will influence x_t through the bias parameter b, while only x_0 can influence x_t*

through the parameter w_0. We also note that the gradients $\frac{\partial L(\mathbf{f}(x_0),y_0)}{\partial w_0}$ and $\frac{\partial L(\mathbf{f}(x_0),y_0)}{\partial b}$ will always be equal mathematically, and thus TracIn(x_0, x_t) will be contributed by parameters w_0 and b equally. This will result in the bias parameter b, which is not crucial for the model $\mathbf{f}(\cdot)$, to have high influence magnitude but differing signs across all examples.

The above example illustrates that while the bias parameter is not important for the prediction model (removing the bias can still lead to the same optimal solution), the total gradient that flows through the bias is much higher than the gradient that flows through each w_i. Hence, the total influence that flows through the bias will be larger than that flowing through the weight, since each training example's gradient will affect the bias but the total contribution will be cancelled out, so the bias will remain 0.

8.2.1 Measuring the Cancellation Effect

In the above example, we defined strong cancellation effect when some weight parameters does not change a lot during training (or has saturated in the training process), but the total strength of the gradient of the weight parameters summed over training data is large. For weights W, we first define two terms ΔW_c and $G(W)_c$,

$$\Delta W_c = \|W_{c+1} - W_c\|,$$

$$G(W)_c = \sum_{x_i,y_i \sim D} \eta_c \| \frac{\partial l(x_i, y_i)}{\partial W_c} \|,$$

where ΔW^c measures the norm of weight parameter change between checkpoint c and $c+1$, and $G(W)_c$ measures the sum of weight gradient norm times learning rate summed over all training data. When ΔW_c is small, this means that the weight W may have saturated at checkpoint c, and the weight may not actually change the model output much. When $G(W)_c$ is large, this means that the sum of gradient norm with respect to W_c is still large, and the influence norm caused by $\frac{\partial l(x_i,y_i)}{\partial W_c}$ will also be large.

To measure the cancellation effect, we define the cancellation ratio of a weight parameter W as:

$$C(W) = \frac{\sum_c G(W)_c}{\sum_c \Delta W_c}.$$

When $G(W)_c$ is large and ΔW_c is small, this means that a non-important weight W_c caused a large change in the total influence norm, which implies high cancellation. In the contrary, when $G(W)_c$ is closer to ΔW_c, that means that the total actual weight change has a similar scale to the total influence caused by the weight change, which implies low cancellation.

8.2.2 Removing Bias In TracIn Calculation to Reduce Cancellation Effect

To investigate whether removing weights with high cancellation effect really helps improve influence quality, we conducted an experiment on a CNN text classification on Agnews dataset with 87% test accuracy. The model is defined as follows: first a token embedding with dimension 128, followed by two convolution layers with kernel size 5 and filter size 10, one convolution layers with kernel size 1 and filter size 10, a global max pooling layer, and a fully connected layer;

all weights are randomly initialized. The first layer is the token embedding, the second layer is the convolution layer, and the last layer is a fully connected layer.

The model has 21222 parameters in total (excluding the token embedding), in which 102 parameters are bias variables. We find $C(\text{bias})$ to be 16789, and $C(\text{weight})$ to be 2555, which validates that the bias variables have a much stronger cancellation effect than the weight variables. A closer analysis shows that $G(\text{bias})$ is similar to $G(\text{weight})$ (627206 and 559142), but $\Delta(\text{bias})$ is much smaller than $\Delta(\text{weight})$ (0.74 and 4.37.) Even though the bias parameters has a much smaller total change compared to the weight parameters, their impact on the gradient norm (and thus influence norm) is even higher than the weight parameters. This verifies the intuition in Example 1 that the bias parameter has a stronger cancellation effect since the gradient to bias is almost activated for all examples despite the actual bias change being small.

To further verify that the TracIn score contributed by the bias may lower the overall discriminatory power, we compute AUC-DEL$_+$ and AUC-DEL$_-$ for TracIn and TracIn-weight on AGnews with our CNN model. The AUC-DEL$_+$ for TracIn and TracIn-weight is -0.036 and -0.065 respectively, and the AUC-DEL$_+$ for TracIn and TracIn-weight is 0.011 and 0.046. The result shows that by removing the TracIn score contributed by the bias (with only 102 parameters), the overall influence quality improves significantly. Thus, in all future experiments, we automatically remove the bias in calculation of data influence if not stated otherwise.

8.2.3 Influence of Latter Layers May Suffer from Cancellation

For scalability reasons, most influence methods choose to operate only on the parameters of the last fully-connected layer Θ_{last}. We argue that this is not a great choice, as the influence scores that stems from the last fully-connected weight layer may suffer from cancellation effect, as different examples "share logics" in the activation representation of this layer. Early layers, where examples have unique logic, may suffer less from the cancellation effect. We report the cancellation ratio for each of the TracIn layer varaint in Table 8.1, where TracIn-first, TracIn-second, TracIn-third, TracIn-last, TracIn-All refer to TracIn scores based on weights of the first layer, second layer, third layer, last layer, and all layers (the bias is always omitted). As we suspected, early layers suffers less from cancellation, and latter layers suffers more from cancellation.

To assess the impact on influence quality, we evaluate the AUC-DEL$_+$ and AUC-DEL$_-$ score for TracIn calculated with different layers on the AGnews CNN model in Tab. 8.1. We observe that removing examples based on influence scores calculated using parameters of later layers (with more "share logic"") leads to worse deletion score compared to removing examples based on influence scores calculated using parameters of earlier layers (with more "unique logic"). Interestingly, the performance of TracIn-first even outperforms TracIn-all where all parameters are used. We hypothesize that since the TracIn score based on later layers contain too much cancellation, it is actually harmful to include these weight parameters in the TracIn calculation. In the following, we develop data influence methods by only using the first layer of the model, which suffers the least from cancellation effect.

8.3 Word Embedding Based Influence

In the previous section, we argue that using the latter layers to calculate influence may lead to the cancellation effect, which over-estimates influence. Another option is to calculate influence on all weight parameters, but may be computational infeasible when larger models with several

Table 8.1: Cancellation Ratio and AUC-DEL table for various layers in CNN model in AGnews.

Dataset	Metric	TR-first	TR-second	TR-third	TR-last	TR-all
AGnews	Cancellation ↓	**1863**	2019	3126	2966	2368
	AUC-DEL+ ↓	**−0.077**	**−0.075**	0.012	−0.016	−0.065
	AUC-DEL− ↑	**0.045**	0.022	0.006	−0.032	**0.046**

Table 8.2: Examples for word similarity for different examples containing word "not".

Example	Premise	Hypothesis	Label
S1	I think he is very annoying.	I do **not** like him.	Entailment
S2	I think reading is very boring.	I do **not** like to read.	Entailment
S3	I think reading is very boring.	I do **not** hate burying myself in books.	Contradiction
S4	She **not** only started playing the piano before she could speak, but her dad taught her to compose music at the same time.	She started to playing music and making music from very long ago.	Entailment
S5	I think he is very annoying.	I don't like him.	Entailment
S6	She thinks reading is pretty boring	She doesn't love to read	Entailment
S7	She not only started playing the piano before she could speak, but her dad taught her to compose music at the same time	She started to playing music and making music from quite long ago	Entailment

millions of parameters are used. To remedy this, we propose operating on the first layer of the model, which contains the less cancellation effect since early layers encodes "unique logit". The first layer for language classification models is usually the word embedding layer in the case of NLP models. However, there are two questions in using the first layer to calculate data influence: 1. the word (token) embedding contains most of the weight parameters, and may be computational expensive 2. the word embedding layer may not capture influential examples through high-level information. In the rest of this section, we develop the idea of word embedding layer based training-data influence in the context of TracIn. We focus on TracIn due to challenges in applying the other methods to the word embedding layer: influence functions on the word embedding layer are computationally infeasible due to the large size (vocab size $\times$ embedding_dimension) of the embedding layer, and representer is designed to only use the final layer. We show that our proposed influence score is scalable thanks to the sparse nature of word embedding gradients, and contains both low-level and high-level information since the gradient to the word embedding layer can capture both high-level and low-level information about the input sentence.

8.3.1 TracIn on Word Embedding Layer

We now apply TracIn on the word embedding weights, obtaining the following expression:

$$\text{TracIn-WE}(x, x') = -\frac{\partial \ell(x, \Theta)}{\partial \Theta_{\text{WE}}}^T \frac{\partial \ell(x', \Theta)}{\partial \Theta_{\text{WE}}}, \tag{8.3}$$

Implementing the above form of TracIn-WE would be computationally infeasible as word embedding layers are typically very large (vocab size $\times$ embedding dimension). For instance, a BERT-base model has 23M parameters in the word embedding layer. To circumvent this, we leverage the sparsity of word embedding gradients $\frac{\partial \ell(x,\Theta)}{\partial \Theta_{\text{WE}}}$, which is a sparse vector where only embedding weights associated with words that occur in x have non-zero value. Thus, the dot product between two word embedding gradients has non-zero values only for words w that occur in both x, x'. With this observation, we can rewrite TracIn-WE as:

$$\text{TracIn-WE}(x, x') = -\sum_{w \in x \cap x'} \frac{\partial \ell(x)}{\partial \Theta_w}^T \cdot \frac{\partial \ell(x')}{\partial \Theta_w}, \tag{8.4}$$

where Θ_w are the weights of the word embedding for word w. We call the term $\frac{\partial \ell(x)}{\partial \Theta_w}^T \cdot \frac{\partial \ell(x')}{\partial \Theta_w}$ the *word gradient similarity* between sentences x, x' over word w.

Table 8.3: Word Decomposition Examples for TracIn-WE

	Sentence content	Label
Test Sentence 1 - T1	I can always end my conversations so you would not get any answers because you are too lazy to remember anything	Toxic
Test Sentence 2 - T2	For me, the lazy days of summer is not over yet, and I advise you to please kindly consider to end one's life, thank you	Toxic
Train Sentence - S1	Oh yeah, if you're too lazy to fix tags yourself, you're supporting AI universal takeover in 2020. end it. kill it now.	Non-Toxic
	Word Importance	Total
TracIn-WE(S1, T1)	[S]: -0.28, [E]: -0.07, to: -0.15, **lazy**: -7.6, you: -0.3, end:-0.3, too:-0.3	-9.2
TracIn-WE(S1, T2)	[S]: -0.17, [E]: -0.23, to: 0.54, lazy: -0.25, you: 0.25, **end**: -3.12	-3.45

8.3.2 Interpreting Word Gradient Similarity

Equation 8.4 gives the impression that TracIn-WE merely considers a bag-of-words style similarity between the two sentences, and does not take the semantics of the sentences into account. This is surprisingly not true! Notice that for overlapping words, TracIn-WE considers the similarity between gradients of word embeddings. Since gradients are back-propagated through all the intermediate layers in the model, they take into account the semantics encoded in the various layers. This is aligned with the use of word gradient norm $\| \frac{\partial \mathbf{f}(x)}{\partial \Theta_w} \|$ as a measure of importance of the word w to the prediction $\mathbf{f}(x)$ [144, 169]. Thus, word gradient similarity would be larger for words that are deemed important to the predictions of the training and test points.

Word gradient similarity is not solely driven by the importance of the word. Surprisingly, we find that word gradient similarity is also larger for overlapping words that appear in similar contexts in the training and test sentences. We illustrate this via an example. Table 8.2 shows 4 synthetic premise-hypothesis pairs for the Multi-Genre Natural Language Inference (MNLI) task [173]. An existing pretrained model [73] predicts these examples correctly with softmax probability between 0.65 and 0.93. Notice that all examples contain the word 'not' once. The

word gradient importance $\|\frac{\partial \mathbf{f}(x)}{\partial \Theta_w}\|$ for "not" is comparable in all 4 sentences. The value of word gradient similarity for 'not' is 0.34 for the pair S1-S2, and -0.12 for S1-S3, while it is -0.05 for S1-S4. This large difference stems from the context in which 'not' appears. The absolute similarity value is larger for S1-S2 and S1-S3, since 'not' appears in a negation context in these examples. (The word gradient similarity of S1-S3 is negative since they have different labels.) However, in S4, 'not' appears in the phrase "not only ... but", which is not a negation (or can be considered as double negation). Consequently, word gradient similarity for 'not' is small between S1 and S4. In summary, we expect the absolute value of TracIn-WE score to be large for training and test sentences that have overlapping important words in similar (or strongly opposite) contexts. On the other hand, overlap of unimportant words like stop words would not affect the TracIn-WE score.

8.3.3 Word-Level Decomposition for TracIn-WE

An attractive property of TracIn-WE is that it decomposes into word-level contributions for both the testing point x' and the training point x. As shown in (8.4), word w in x contributes to TracIn-WE(x, x') by the amount $\frac{\partial \ell(x)}{\partial \Theta_w}^T \cdot \frac{\partial \ell(x')}{\partial \Theta_w} \mathbb{1}[w \in x']$; a similar word-level decomposition can be obtained for x'. Such a decomposition helps us identify which words in the training point (x) drive its influence towards the test point (x'). For instance, consider the example in Table. 8.3, which contains two test sentences (T1, T2) and a training sentence S1. We decompose the score TracIn-WE(S1, T1) and TracIn-WE(S1,T2) into words contributions, and we see that the word "lazy" dominates TracIn-WE(S1, T1), and the word "end" dominates TracIn-WE(S1, T2). This example shows that different key words in a training sentence may drive influence towards different test points. The word decomposition of TracIn-WE is also helpful in applications where we seek to fix mis-classifications by dropping or substituting words in the training examples that most influence it.

8.3.4 An approximation for TracIn-WE

As we note in Sec. 8.3.1, the space complexity of saving training and test point gradients scales with the number of words in the sentence. This may be intractable for tasks with very long sentences. We alleviate this by leveraging the fact that the word embedding gradient for a word w is the sum of input word gradients from each position where w is present. Given this decomposition, we can approximate the word embedding gradients by saving only the top-k largest input word gradients for each sentence. (An alternative is to save the input word gradients that are above a certain threshold.) Formally, we define the approximation

$$\frac{\partial \ell(x, \Theta)}{\partial \Theta_w}|_{\text{top-k}} = \sum_{i \in x^{\text{top-k}} \wedge x^i = w} \frac{\partial \ell(x, \Theta)}{\partial x^i} \tag{8.5}$$

where x^i is the word at position i, and x^{top-k} is the set of top-k input positions by gradient norm. We then propose

$$\text{TracIn-WE-Topk}(x, x) = -\sum_{w \in x \cap x'} \frac{\partial \ell(x, \Theta_w)}{\partial \Theta_w}|_{\text{top-k}}^T \cdot \frac{\partial \ell(x', \Theta_w)}{\partial \Theta_w}|_{\text{top-k}}. \tag{8.6}$$

Computational complexity Let L be the max length of each sentence, d be the word embedding dimension, and o be the average overlap between two sentences. If the training and test point

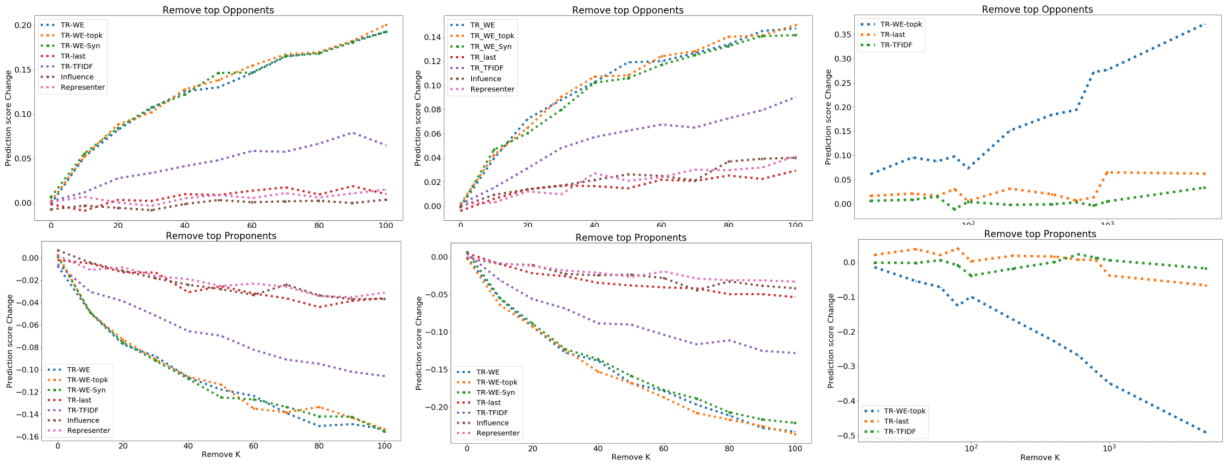


Figure 8.1: Deletion Curve for removing opponents (top figure, larger better) and proponents (bottom figure, smaller better) on Toxicity (left), AGnews (mid), and MNLI (right).

gradients are precomputed and saved then the average computation complexity for calculating TracIn-WE for m training points and n testing points is $O(mnod)$. This can be contrasted with the average computation complexity for influence functions on the word embedding layer, which takes $O(mnd^2v^2 + d^3v^3)$, where v is the vocabulary size which is typically larger than 10^4, and o is typically less than 5. The approximation for TracIn-We-Topk drops the computational complexity from $O(mnod)$ to $O(mno_kd)$ where o_k is the average overlap between the sets of top-k words from the two sentences. It has the additional benefit of preventing unimportant words (ones with small gradient) from dominating the word similarity by multiple occurrences, as such words may get pruned. In all our experiments, we set k to 10 for consistency, and do not tune this hyper-parameter.

8.3.5 Influence without Word-Overlap

One potential criticism of TracIn-WE is that it may not capture any influence when there are no overlapping words between x and x'. To address this, we note that modern NLP models often include a "start" and "end" token in all inputs. We posit that gradients of the embedding weights of these tokens take into account the semantics of the input (as represented in the higher layers), and enable TracIn-WE to capture influence between examples that are semantically related but do not have any overlapping words. We illustrate this in S5-S7 in Tab. 8.2 via examples for the MNLI task. Sentence S5 has no overlapping words with S6 and S7. However, the word gradient similarity of "start" and "end" tokens for the pair S5-S6 is 1.15, while that for the pair S5-S7 is much lower at -0.05. Indeed, sentence S5 is more similar to S6 than S7 due to the presence of similar word pairs (e.g., think and thinks, annoying and boring), and the same negation usage. We further validate that TracIn-WE can capture influence from examples without word overlap via a controlled experiment in Sec. 8.4.

8.4 Experiments

We evaluate the proposed influence methods on 3 different NLP classification datasets with BERT models. We choose a transformer-based model as it has shown great success on a series of down-stream tasks in NLP, and we choose BERT model as it is one of the most commonly used

Table 8.4: AUC-DEL table for various methods in different datasets.

Dataset	Metric	Inf-Last	Rep	TR-last	TR-WE	TR-WE-topk	TR-TFIDF
Toxic	AUC-DEL+ ↓	−0.022	−0.021	−0.025	**−0.105**	−0.104	−0.067
	AUC-DEL− ↑	−0.001	0.006	0.007	0.122	**0.125**	0.044
AGnews	AUC-DEL+ ↓	−0.025	−0.021	−0.032	−0.148	**−0.152**	−0.083
	AUC-DEL− ↑	0.023	0.021	0.017	**0.100**	**0.100**	0.054
MNLI	AUC-DEL+ ↓			0.006		**−0.198**	−0.004
	AUC-DEL− ↑			0.026		**0.169**	0.005
Dataset	Metric	Inf-Last	Rep	TR-last	TR-WE	TR-WE-topk	TR-WE-NoC
Toxic	AUC-DEL+ ↓	−0.011	−0.015	−0.007	**−0.033**	−0.016	0.005
Nooverlap	AUC-DEL− ↑	0.013	0.012	0.013	**0.043**	0.042	−0.002

transformer model. For the smaller Toxicity and AGnews dataset, we operate on the Bert-Small model, as it already achieves good performance. For the larger MNLI dataset, we choose the Bert-Base model with $110M$ model parameters, which is a decently large model which we believe could represent the effectiveness of our proposed method on large-scale language models. As discussed in Section 8.1.2, we use the *case deletion* evaluation and report the metrics on the deletion curve.

Baselines One question to ask is whether the good performance of TracIn-WE is a result that it captures the low-level word information well. To answer this question, we design a synthetic data influence score as the TF-IDF similarity [135] multiplied by the loss gradient dot product for x and x'. TR-TFIDF can be understood by replacing the embedding similarity of TracIn-Last by the TF-IDF similarity, which captures low level similarity.

$$\text{TR-TFIDF}(x, x') = -\text{Tf-Idf}(x, x') \frac{\partial \ell(x, \Theta)}{\partial \mathbf{f}(x, \Theta)}^T \frac{\partial \ell(x', \Theta)}{\partial \mathbf{f}(x, \Theta)}. \tag{8.7}$$

Toxicity. We first experiment on the toxicity comment classification dataset [81], which contains sentences that are labeled toxic or non-toxic. We randomly choose $50,000$ training samples and $20,000$ validation samples. We then fine-tune a BERT-small model on our training set, which leads to 96% accuracy. Out of the $20,000$ validation samples, we randomly choose 20 toxic and 20 non-toxic samples, for a total of 40 samples as our targeted test set. For each example x' in the test set, we remove top-k proponents and top-k opponents in the training set respectively,

and retrain the model to obtain $\text{DEL}_+(x', k, \delta)$ and $\text{DEL}_-(x', k, \delta)$ for each influence method δ. We vary k over $\{10, 20, \ldots, 100\}$. For each k, we retrain the model 10 times and take the average result, and then average over the 40 test points. We implement the methods Influence-last, Representer Points, TracIn-last, TracIn-WE, TracIn-WE-Topk, TracIn-WE-Syn, TracIn-TFIDF, and abbreviate TracIn with TR in the experiments. The AUC-DEL$_+$ and AUC-DEL$_-$ scores are reported in Table 8.4. We see that our proposed TracIn-WE method, along with its variants TracIn-WE-Topk outperform other methods by a significant margin. As mentioned in Sec. 8.2.3, TF-IDF based method beats the existing data influence methods using last layer weights by a decisive margin as well, but is still much worse compared to TracIn-WE. Therefore, TracIn-WE did not succeed by solely using low-level information.

AGnews. We next experiment on the AG-news-subset [64, 190], which contains a corpus of news with 4 different classes. We follow our setting in toxicity and choose $50,000$ training samples, $20,000$ validation samples, and fine-tune with the same BERT-small model that achieves 90% accuracy on this dataset. We randomly choose 100 samples with 25 from each class as our targeted test set. The AUC-DEL$_+$ and AUC-DEL$_-$ scores for $k \in \{10, 20, \ldots, 100\}$ are reported in Table 8.4. Again, we see that the variants of TracIn-WE significantly outperform other existing methods applied on the last layer. In both AGnews and Toxicity, removing 10 top-proponents or top-opponents for TracIn-WE has more impact on the test point compared to removing 100 top-proponents or top-opponents for TracIn-last.

MNLI. Finally, we test on a larger scale dataset, Multi-Genre Natural Language Inference (MultiNLI) [173], which consists of $433k$ sentence pairs with textual entailment information, including entailment, neutral, and contradiction. In this experiment, we use the full training and validation set, and BERT-base which achieves 84% accuracy on matched-MNLI validation set. We choose 30 random samples with 10 from each class as our targeted test set. We only evaluate TracIn-WE-Topk, TracIn-last and TracIn-TFIDF as those were the most efficient methods to run at large scale. We vary $k \in \{20, \ldots, 5000\}$, and the AUC-DEL$_+$ and AUC-DEL$_-$ scores for our test set are reported in Table 8.4. Unlike previous datasets, here TracIn-TFIDF does not perform better than TracIn-Last, which may be because input similarity for MNLI cannot be merely captured by overlapping words. For instance, a single negation would completely change the label of the sentence. However, we again see TracIn-WE-Topk significantly outperforms TracIn-Last and TracIn-TFIDF, demonstrating its efficacy over natural language understanding tasks as well. This again provides evidence that TracIn-WE can capture both low-level information and high-level information. The deletion curve of Toxicity, AGnews, MNLI is in shown in Fig. 8.1.

No Word Overlap. To assess whether TracIn-WE can do well in settings where the training and test examples do not have overlapping words, we construct a controlled experiment on the Toxicity dataset. Given a test sentence x', we only consider the top-5000 training sentences (out of $50,000$) with the least word overlap for computing influence. We use TF-IDF similarity to rank the number of word overlaps so that stop word overlap will not be over-weighted. We also find that when the word-embedding layer is fixed during training (result when word-embedding is not fixed is in the appendix, where removing examples based on any influence method does not change the prediction), sentence with no word overlaps carry more influence. The AUC-DEL$_+$ and AUC-DEL$_-$ scores are reported in the lower section of Table 8.4. We find that TracIn-WE

variants can outperform last-layer based influence methods even in this controlled setting, showing that TracIn-WE can retrieve influential examples even without non-trivial word overlaps. In Section 8.3.5, we claimed that this gain stems from the presence of common tokens ("start", "end"). To validate this, we compared with a controlled variant, TracIn-WE-NoCommon (TR-WE-NoC) where the common tokens are removed from TracIn-WE. As expected, this variant performed much worse on the AUC-DEL$_+$ and AUC-DEL$_-$ scores, thus confirming our claim.

8.5 Related Work

In the field of explainable machine learning, our works belongs to training data importance [79, 84, 89, 128, 153, 179]. Other forms of explanations include feature importance feature-based explanations, gradient-based explanations [5, 10, 11, 104, 124, 129, 142, 144, 156, 181, 187] and perturbation-based explanations [104, 124, 129, 181], self-explaining models [23, 100, 170], counterfactuals to change the outcome of the model [35, 58, 74, 167, 168], concepts of the model [88, 192]. For applications on applying data importance methods on NLP tasks, there have been works identifying data artifacts [70, 125] and improving models [68, 69] based on existing data importance method using the influence function or TracIn. In this work, we discussed weight parameter selection to reduce cancellation effect for training data attribution. There has been works that discuss how to cope with cancellation in the context of feature attribution: Liu et al. [103] discusses how regularization during training reduces cancellation of feature attribution, Kapishnikov et al. [82] discusses how to optimize IG paths to minimize cancellation of IG attribution, and Sundararajan et al. [157] discusses improved visualizations to adjust for cancellation.

8.6 Conclusion

In this work, we revisit the common practice of computing training data influence using only last layer parameters. We show that last layer representations in language classification models can suffer from the cancellation effect, which in turn leads to inferior results on influence. We instead recommend computing influence on the word embedding parameters, and apply this idea to propose a variant of TracIn called TracIn-WE. We show that TracIn-WE significantly outperforms last versions of existing influence methods on three different language classification tasks, with trained-from-scratch to fine-tuning training strategy, and also affords a word-level decomposition of influence that aids interpretability.

Chapter 9

Conclusion and Discussions

In this thesis, we have discussed extensive forms of objective criteria for different types of explanations, where our main goal is to develop new and useful explanations based on the objective criteria. Faithfulness-motivated objective criteria are crucial to capture model properties by explanations, and theoretical-motivated objective criteria are useful to motivate the design of explanations theoretically. Application-motivated objective criteria are closely connected to real-world applications but may be less straightforward to guide our design of explanations. In Chapter 8, we have demonstrated the deletion evaluation (a application-motivated objective criterion) is useful for layer selection of explanations of TracIn [128], which itself is motivated by theoretical-motivated objective criteria. This is an example where multiple objective criteria can be combined to design a new explanation.

While our focus in this thesis is on how to use objective criteria to guide the design of explanations, we emphasize that objective criteria should not be used as the golden standard evaluation for explanations but rather as a motivation to design explanations. It is possible that explanations satisfy some objective criteria, but is not interpretable or faithful to the model at all. Therefore, users of explanations should always try to first understand the objective criteria that motivated an explanation, and then decide whether these objective criteria suit the user's application. For instance, while we learn a set of complete concepts in [182], the concepts are not guaranteed to be interpretable based on the completeness criteria. To address this, we introduced an interpretable regularizer to enforce the concepts to be understandable by humans, which is verified by a user study.

Another limitation is that the objective criteria might not be generally applicable to all use cases. For example, the infidelity criteria require a user-defined perturbation, which makes a lot of sense if the users know which features they would like to perturb. However, there may not be a very clear perturbation distribution based on the application, and it may not be straightforward how to use the infidelity metric. This limitation similarly applies to theoretically-motivated explanations, as it may not always be obvious whether a specific axiomatic property is helpful for a user or not. Therefore, the faithfulness-motivated and theoretically-motivated objective criteria should be seen as a design motivation for explanations in a specific context but not a golden standard evaluation for any explanations. Practitioners should only use the objective criteria to select or design explanations when the objective criteria fit the user's need (which potentially makes the choice of "objective" criteria not so objective). On the other hand, the application-driven

objective criteria may be more generally applicable to different explanations, since the evaluation is designed with a specific application in mind, which can be applied whenever the application is relevant to the end users. The pitfall of application-driven objective criteria is that it is usually not as straightforward to design explanations that optimize the objective criteria.

Bibliography

[1] Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *arXiv preprint arXiv:1903.10464*, 2019. 3

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9525–9536, 2018. 1, 1.1, 2.4

[3] Philip Adler, Casey Falk, Sorelle A Friedler, Tionney Nix, Gabriel Rybeck, Carlos Scheidegger, Brandon Smith, and Suresh Venkatasubramanian. Auditing black-box models for indirect influence. *Knowledge and Information Systems*, 54(1):95–122, 2018. 3

[4] David Alvarez-Melis and Tommi S. Jaakkola. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*, 2018. 2, 2.2

[5] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. A unified view of gradient-based attribution methods for deep neural networks. *International Conference on Learning Representations*, 2018. 1.1, 2, 2.1.1, 2.1.1, 2.1.5, 3.6.2, 4, 4.3.1, 4.4, 8.5

[6] Rushil Anirudh, Jayaraman J Thiagarajan, Rahul Sridhar, and Timo Bremer. Influential sample selection: A graph signal processing approach. *arXiv preprint arXiv:1711.05407*, 2017. 1.1, 5.1

[7] André Araujo, Wade Norris, and Jack Sim. Computing receptive fields of convolutional neural networks. *Distill*, 4(11):e21, 2019. 6.1

[8] Sharon Lee Armstrong, Lila R. Gleitman, and Henry Gleitman. What some concepts might not be. *Cognition*, 13(3):263–308, 1983. ISSN 0010-0277. 6

[9] Leila Arras, Ahmed Osman, Klaus-Robert Müller, and Wojciech Samek. Evaluating recurrent neural network explanations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 113–126, 2019. 3.6.2

[10] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015. 1.1, 2.1.1, 2, 4, 4.4.3, 5.1, 8.5

[11] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010. 1.1, 4, 8.5

[12] Mohammad Taha Bahadori and David Heckerman. Debiasing concept-based explanations with causal analysis. In *International Conference on Learning Representations*, 2020. 1.1

[13] John F Banzhaf III. Weighted voting doesn't work: A mathematical analysis. *Rutgers L. Rev.*, 19:317, 1964. 4.2.2, 7

[14] Jacob Bien and Robert Tibshirani. Prototype selection for interpretable classification. *The Annals of Applied Statistics*, pages 2403–2424, 2011. 1.1, 5.1

[15] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *JMLR*, 3 (Jan):993–1022, 2003. 6

[16] Bastian Bohn, Michael Griebel, and Christian Rieger. A representer theorem for deep kernel learning. *arXiv preprint arXiv:1709.10441*, 2017. 5.1

[17] Diane Bouchacourt and Ludovic Denoyer. Educe: Explaining model decisions through unsupervised concepts extraction. *arXiv preprint arXiv:1905.11852*, 2019. 6

[18] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017. 1.1, 4, 4.1.2

[19] Javier Castro, Daniel Gómez, and Juan Tejada. Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730, 2009. 3.6.2

[20] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1MXz20cYQ. 1.1, 4, 4.4.2

[21] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan Boyd-Graber, and David Blei. Reading tea leaves: How humans interpret topic models. *Advances in neural information processing systems*, 22, 2009. 1.1

[22] A Charnes, B Golany, M Keane, and J Rousseau. Extremal principle solutions of games in characteristic function form: core, chebychev and shapley value generalizations. In *Econometrics of planning and efficiency*, pages 123–133. Springer, 1988. 7.3, 7.7

[23] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems*, pages 8928–8939, 2019. 6.3.2, 8.5

[24] Hugh Chen, Joseph D Janizek, Scott Lundberg, and Su-In Lee. True to the model or true to the data? *arXiv preprint arXiv:2006.16234*, 2020. 3, 7

[25] Valerie Chen, Jeffrey Li, Joon Sik Kim, Gregory Plumb, and Ameet Talwalkar. Interpretable machine learning: Moving from mythos to diagnostics. *Queue*, 19(6):28–56, 2022. 1

[26] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020. 1.1

[27] Hao-Fei Cheng, Logan Stapleton, Ruiqi Wang, Paige Bullock, Alexandra Chouldechova, Zhiwei Steven Steven Wu, and Haiyi Zhu. Soliciting stakeholders' fairness notions in child maltreatment predictive systems. In *Proceedings of the 2021 CHI Conference on Human*

Factors in Computing Systems, pages 1–17, 2021. 1

[28] R Dennis Cook and Sanford Weisberg. *Residuals and influence in regression*. New York: Chapman and Hall, 1982. 8, 8.1.2

[29] Tianyu Cui, Pekka Marttinen, and Samuel Kaski. Learning global pairwise interactions with bayesian neural networks. *arXiv preprint arXiv:1901.08361*, 2019. 7.7

[30] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*, pages 6967–6976. NeurIPS, 2017. 2, 4, 4.4.2

[31] Anupam Datta, Shayak Sen, and Yair Zick . Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 598–617. IEEE, 2016. 1.2, 2.1.1, 3, 7

[32] Glenn De'ath and Katharina E Fabricius. Classification and regression trees: a powerful yet simple technique for ecological data analysis. *Ecology*, 81(11):3178–3192, 2000. 1.1

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*. CVPR, 2009. 4.4

[34] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 7.6

[35] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems*, pages 592–603. NeurIPS, 2018. 1.1, 4, 4.4.3, 8.5

[36] Guoli Ding, Robert F Lax, Jianhua Chen, and Peter P Chen. Formulas for approximating pseudo-boolean random variables. *Discrete Applied Mathematics*, 156(10):1581–1597, 2008. 7.5.2.3, 7.7

[37] Guoli Ding, Robert F Lax, Jianhua Chen, Peter P Chen, and Brian D Marx. Transforms of pseudo-boolean random variables. *Discrete Applied Mathematics*, 158(1):13–24, 2010. 7.5.2.3, 7.7

[38] Finale Doshi-Velez and Been Kim. A roadmap for a rigorous science of interpretability. *CoRR*, abs/1702.08608, 2017. URL http://arxiv.org/abs/1702.08608. 2.4

[39] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017. 1, 4

[40] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml. 3.6.1

[41] Pradeep Dubey and Lloyd S Shapley. Mathematical properties of the banzhaf power index. *Mathematics of Operations Research*, 4(2):99–131, 1979. 4.2.2, 4.2.2

[42] Jacob Eisenstein, Amr Ahmed, and Eric P Xing. Sparse additive generative models of text. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1041–1048, 2011. 1.1

[43] Edith Elkind, Piotr Faliszewski, Martin Lackner, Dominik Peters, and Nimrod Talmon. Committee scoring rules, banzhaf values, and approximation algorithms. In *4th workshop on exploring beyond the worst case in computational social choice (EXPLORE'17)*, 2017. 2

[44] Gabriel Erion, Joseph D Janizek, Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Learning explainable models using attribution priors. *arXiv preprint arXiv:1906.10670*, 2019. 4.3, 4.3.1, 4.4

[45] Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3449–3457, 2017. 1.1, 4, 4.4

[46] Christopher Frye, Damien de Mijolla, Laurence Cowton, Megan Stanley, and Ilya Feige. Shapley-based explainability on the data manifold. *arXiv preprint arXiv:2006.01272*, 2020. 3, 3.2.1, 3.2.2.2, 3.2.2.2, 3.2.2.3, 3.2.3, 7

[47] Katsushige Fujimoto, Ivan Kojadinovic, and Jean-Luc Marichal. Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices. *Games and Economic Behavior*, 55(1):72–99, 2006. 7.1.2, 7.2, 7.5.1, 7.5.2.1

[48] 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. 4.1.2

[49] Asma Ghandeharioun, Been Kim, Chun-Liang Li, Brendan Jou, Brian Eoff, and Rosalind W Picard. Dissect: Disentangled simultaneous explanations via concept traversals. *arXiv preprint arXiv:2105.15164*, 2021. 1.1

[50] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR, 2019. 3, 7

[51] Amirata Ghorbani and James Zou. Neuron shapley: Discovering the responsible neurons. *arXiv preprint arXiv:2002.09815*, 2020. 3

[52] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. *AAAI*, 2019. 2

[53] Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations. In *Advances in Neural Information Processing Systems*, 2019. 1.1

[54] Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations. *NeurIPS*, 2019. 6, 6.1, 6.3.2

[55] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018. 6

[56] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1.1

[57] Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. Explaining classifiers with causal concept effect (cace). *arXiv preprint arXiv:1907.07165*, 2019. 1.1

[58] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. ICML, 2019. 1.1, 8.5

[59] Michel Grabisch and Marc Roubens. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of Game Theory*, 28(4):547–565, 1999. 7, 7.2, 7.2, 7.2, 7.2, 7.2, 7.2, 7.5.1

[60] Michel Grabisch and Agnieszka Rusinowska. k-additive upper approximation of tu-games. *Operations Research Letters*, 48(4):487–492, 2020. 7.7

[61] Michel Grabisch, Jean-Luc Marichal, and Marc Roubens. Equivalent representations of set functions. *Mathematics of Operations Research*, 25(2):157–178, 2000. 7, 7.3.1, 7.5.2.1, 7.5.2.3, 7.7

[62] Michel Grabisch et al. *Set functions, games and capacities in decision making*, volume 46. Springer, 2016. 7.7

[63] Ulrike Grömping. Estimators of relative importance in linear regression based on variance decomposition. *The American Statistician*, 61(2):139–147, 2007. 3, 7

[64] Antonio Gulli. Ag corpus of news articles. *http://groups.di.unipi.it/ gulli/AG_corpus_of_news_articles.html*, 2015. 8.4

[65] Han Guo, Nazneen Fatema Rajani, Peter Hase, Mohit Bansal, and Caiming Xiong. Fastif: Scalable influence functions for efficient model interpretation and debugging. *arXiv preprint arXiv:2012.15781*, 2020. 1.1

[66] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010. 3.5

[67] Peter L Hammer and Ron Holzman. Approximations of pseudo-boolean functions; applications to game theory. *Zeitschrift für Operations Research*, 36(1):3–21, 1992. 4.2.2, 4.2.2, 7, 7.3, 7.5.2.3, 7.7

[68] Xiaochuang Han and Yulia Tsvetkov. Fortifying toxic speech detectors against disguised toxicity. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7732–7739, 2020. 8.5

[69] Xiaochuang Han and Yulia Tsvetkov. Influence tuning: Demoting spurious correlations via instance attribution and instance-driven updates. *arXiv preprint arXiv:2110.03212*, 2021. 8.5

[70] Xiaochuang Han, Byron C Wallace, and Yulia Tsvetkov. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5553–5563, 2020. 8.1.2, 8.5

[71] John C Harsanyi. A simplified bargaining model for the n-person cooperative game.

International Economic Review, 4(2):194–220, 1963. 7, 7.2

[72] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5.3.2

[73] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020. 8.3.2

[74] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *ECCV*. ECCV, 2018. 1.1, 8.5

[75] Gaurush Hiranandani, Harikrishna Narasimhan, and Sanmi Koyejo. Fair performance metric elicitation. *Advances in Neural Information Processing Systems*, 33:11083–11095, 2020. 1

[76] Giles Hooker and Lucas Mentch. Please stop permuting features: An explanation and alternatives. *arXiv preprint arXiv:1905.03151*, 2019. 3

[77] Cheng-Yu Hsieh, Chih-Kuan Yeh, Xuanqing Liu, Pradeep Kumar Ravikumar, Seungyeon Kim, Sanjiv Kumar, and Cho-Jui Hsieh. Evaluations and methods for explanation through robustness analysis. In *International Conference on Learning Representations*. ICLR, 2021. URL https://openreview.net/forum?id=4dXmpCDGNp7. 1.2

[78] Joseph D Janizek, Pascal Sturmfels, and Su-In Lee. Explaining explanations: Axiomatic feature interactions for deep networks. *Journal of Machine Learning Research*, 22(104): 1–54, 2021. 7.7

[79] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1167–1176. PMLR, 2019. 3, 7, 8.5

[80] S. Joshi, O. Koyejo, Warut D. Vijitbenjaronk, Been Kim, and Joydeep Ghosh. Towards realistic individual recourse and actionable explanations in black-box decision making systems. *ArXiv*, abs/1907.09615, 2019. 1.1

[81] Kaggle.com. Toxic comment classification challenge: Identify and classify toxic online comments. *https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge*, 2018. 8.4

[82] Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5050–5058, 2021. 8.5

[83] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017. 4, 4.1.2

[84] Rajiv Khanna, Been Kim, Joydeep Ghosh, and Oluwasanmi Koyejo. Interpreting black box predictions using fisher kernels. *arXiv preprint arXiv:1810.10118*, pages 3382–3390, 2018. 8.5

[85] Rajiv Khanna, Been Kim, Joydeep Ghosh, and Sanmi Koyejo. Interpreting black box predictions using fisher kernels. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3382–3390. PMLR, 2019. 1.1

[86] Been Kim, Cynthia Rudin, and Julie A Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Advances in Neural Information Processing Systems*, pages 1952–1960, 2014. 1.1, 1.1, 5.1

[87] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems*, pages 2280–2288. NeurIPS, 2016. 1.1, 5.1

[88] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, pages 2673–2682. ICML, 2018. 1.1, 1.2, 2.4, 6, 6.1, 8.5

[89] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. ICML, 2017. 1.1, 4, 5.1, 5.3.2, 5.3.5, 8, 8.1.1, 8.1.2, 8.1.2, 8.5

[90] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. *arXiv preprint arXiv:2007.04612*, 2020. 1.1

[91] Pang Wei W Koh, Kai-Siang Ang, Hubert Teo, and Percy S Liang. On the accuracy of influence functions for measuring group effects. *Advances in neural information processing systems*, 32, 2019. 8.1.2

[92] Satyapriya Krishna, Tessa Han, Alex Gu, Javin Pombra, Shahin Jabbari, Steven Wu, and Himabindu Lakkaraju. The disagreement problem in explainable machine learning: A practitioner's perspective. *arXiv e-prints*, pages arXiv–2202, 2022. 1

[93] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5.2.5

[94] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pages 126–137. ACM, 2015. 2

[95] I Elizabeth Kumar, Suresh Venkatasubramanian, Carlos Scheidegger, and Sorelle Friedler. Problems with shapley-value-based explanations as feature importance measures. *arXiv preprint arXiv:2002.11097*, 2020. 3

[96] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, pages 951–958. IEEE, 2009. 6.3.2

[97] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5.3.1

[98] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs*

[Online]. Available: http://yann. lecun. com/exdb/mnist, 2, 2010. 4.4

[99] Chung-Wei Lee, Wei Fang, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. Multi-label zero-shot learning with structured knowledge graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1576–1585, 2018. 1.2.1

[100] Guang-He Lee, Wengong Jin, David Alvarez-Melis, and Tommi Jaakkola. Functional transparency for structured data: a game-theoretic approach. In *International Conference on Machine Learning*, pages 3723–3733. PMLR, 2019. 8.5

[101] Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220, 2016. URL http://arxiv.org/abs/1612.08220. 4.4

[102] Richard Harold Lindeman. Introduction to bivariate and multivariate analysis. Technical report, 1980. 3, 7

[103] Frederick Liu, Amir Najmi, and Mukund Sundararajan. The penalty imposed by ablated data augmentation. *arXiv preprint arXiv:2006.04769*, 2020. 8.5

[104] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774. NeurIPS, 2017. 1.1, 1.2, 2, 2.1.1, 5, 2.1.4, 2.4, 3, 3.1.2.1, 3.1.2.2, 3.4, 4, 4.1.2, 4.4, 6, 6.2.3, 7, 7.1.1, 8.5

[105] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018. 3

[106] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1): 56–67, 2020. 7.7

[107] 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. Association for Computational Linguistics, 2011. 7.6

[108] 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. 4, 4.1.2

[109] Jean-Luc Marichal and Pierre Mathonet. Weighted banzhaf power and interaction indexes through weighted approximations of games. *European journal of operational research*, 211(2):352–358, 2011. 7.5.2.3, 7.7

[110] Jean-Luc Marichal and Marc Roubens. The chaining interaction index among players in cooperative games. In *Advances in Decision Analysis*, pages 69–85. Springer, 1999. 7.7

[111] Jon D Mcauliffe and David M Blei. Supervised topic models. In *NIPS*, 2008. 6.2.2

[112] Luke Merrick and Ankur Taly. The explanation game: Explaining machine learning models using shapley values. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 17–38. Springer, 2020. 3

[113] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *arXiv preprint arXiv:1706.07269*, 2017. 2

[114] Christoph Molnar, Giuseppe Casalicchio, and Bernd Bischl. Quantifying model complexity via functional decomposition for better post-hoc interpretability. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 193–204. Springer, 2019. 7.7

[115] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2017. 2

[116] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019. 1, 1

[117] J William Murdock, David W Aha, and Leonard A Breslow. Assessing elaborated hypotheses: An interpretive case-based reasoning approach. In *International Conference on Case-Based Reasoning*, pages 332–346. Springer, 2003. 1.1

[118] Jose Oramas, Kaili Wang, and Tinne Tuytelaars. Visual explanation by interpretation: Improving visual feedback capabilities of deep neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1ziPjC5Fm. 4.4.3

[119] Art B Owen. Sobol'indices and shapley value. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):245–251, 2014. 3, 7

[120] Art B Owen and Clémentine Prieur. On shapley value for measuring importance of dependent inputs. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):986–1002, 2017. 3, 7

[121] Guillermo Owen. Multilinear extensions of games. *Management Science*, 18(5-part-2): 64–79, 1972. 7.5.2

[122] Biswajit Paria, Chih-Kuan Yeh, Ian EH Yen, Ning Xu, Pradeep Ravikumar, and Barnabás Póczos. Minimizing flops to learn efficient sparse representations. In *International Conference on Learning Representations*, 2019. 1.2.1

[123] The European Parliament and The European Council. General data protection regulation. *Off. J. Eur. Union*, 2014. 1

[124] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018. 1.1, 2.1.4, 4, 4.1.2, 4.1.2, 4.4.2, 8.5

[125] Pouya Pezeshkpour, Sarthak Jain, Sameer Singh, and Byron C Wallace. Combining feature and instance attribution to detect artifacts. *arXiv preprint arXiv:2107.00323*, 2021. 8.5

[126] Gregory Plumb, Denali Molitor, and Ameet S Talwalkar. Model agnostic supervised local explanations. In *Advances in Neural Information Processing Systems*, pages 2515–2524, 2018. 1.1, 2.1.1, 2.1.5, 4

[127] Rafael Poyiadzi, Kacper Sokol, Raúl Santos-Rodríguez, T. D. Bie, and Peter A. Flach. Face: Feasible and actionable counterfactual explanations. *Proceedings of the AAAI/ACM*

Conference on AI, Ethics, and Society, 2020. 1.1

[128] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33, 2020. 1.1, 8, 8.1.1, 8.5, 9

[129] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016. 1.1, 4, 5.1, 6, 7.1.1, 8.5

[130] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 1.1, 4.4, 5.1

[131] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. 1.1

[132] Luis M Ruiz, Federico Valenciano, and Jose M Zarzuelo. The least square prenucleolus and the least square nucleolus. two values for tu games based on the excess vector. *International Journal of Game Theory*, 25(1):113–134, 1996. 7, 7.7

[133] Luis M Ruiz, Federico Valenciano, and Jose M Zarzuelo. The family of least square values for transferable utility games. *Games and Economic Behavior*, 24(1-2):109–130, 1998. 7.5.2.3, 7.7

[134] Hadi Salman, Greg Yang, Huan Zhang, Cho-Jui Hsieh, and Pengchuan Zhang. A convex relaxation barrier to tight robust verification of neural networks. *arXiv preprint arXiv:1902.08722*, 2019. 4.1.2

[135] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988. 8.4

[136] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016. 3.6.2, 4, 4.1.2, 4.1.2, 4.4.2

[137] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001. 5.1

[138] Jessica Schrouff, Sebastien Baur, Shaobo Hou, Diana Mincu, Eric Loreaux, Ralph Blanes, James Wexler, Alan Karthikesalingam, and Been Kim. Best of both worlds: local and global explanations with human-understandable concepts. *arXiv preprint arXiv:2106.08641*, 2021. 1.1

[139] Lloyd Shapley. A value fo n-person games. *Ann. Math. Study28, Contributions to the Theory of Games, ed. by HW Kuhn, and AW Tucker*, 2(28):307–317, 1953. 1, 1.2, 7, 7.1.2

[140] Lloyd S. Shapley. *A value for n-person games*, volume 2, page 31–40. Cambridge University Press, 1988. doi: 10.1017/CBO9780511528446.003. 3, 3.1.1, 3.4, 6, 6.2.3,

6.2.3

[141] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016. 3

[142] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *International Conference on Machine Learning*, 2017. 1.1, 2.1.1, 2, 2.4, 4, 4.4, 5.1, 8.5

[143] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5.2.5

[144] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 1.1, 4, 5.1, 8.3.2, 8.5

[145] Chandan Singh, W James Murdoch, and Bin Yu. Hierarchical interpretations for neural network predictions. *arXiv preprint arXiv:1806.05337*, 2018. 7.7

[146] Gagandeep Singh, Timon Gehr, Matthew Mirman, Markus Püschel, and Martin Vechev. Fast and effective robustness certification. In *Advances in Neural Information Processing Systems*, pages 10802–10813, 2018. 4.1.2

[147] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020. 3, 3.2.1, 3.2.3, 3.6.1

[148] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 2, 2.3, 2.4, 2.4, 5.3.4

[149] Anders Søgaard et al. Revisiting methods for finding influential examples. *arXiv preprint arXiv:2111.04683*, 2021. 8.1.2

[150] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 1.1, 2.4

[151] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014. 2.1.1, 3

[152] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Visualizing the impact of feature attribution baselines. *Distill*, 5(1):e22, 2020. 4.1.2, 4.4, 4.4.2

[153] Yi Sui, Ga Wu, and Scott Sanner. Representer point selection via local jacobian expansion for post-hoc classifier explanation of deep neural networks and ensemble models. *Advances in Neural Information Processing Systems*, 34, 2021. 8.1.1, 8.5

[154] Yi Sun and Mukund Sundararajan. Axiomatic attribution for multilinear functions. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 177–178, 2011. 3.1.2.2

[155] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. *arXiv preprint arXiv:1908.08474*, 2019. 3, 3.1.2.1, 3.2.2.1, 3.4, 3.4, 3.4, 8, 7

[156] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017. 1.1, 2, 2.1.1, 2.1.2, 2, 2.4, 4, 4.1.2, 4.4, 5.1, 8.5

[157] Mukund Sundararajan, Jinhua Xu, Ankur Taly, Rory Sayres, and Amir Najmi. Exploring principled visualizations for deep network attributions. In *IUI Workshops*, volume 4, 2019. 8.5

[158] Mukund Sundararajan, Kedar Dhamdhere, and Ashish Agarwal. The shapley taylor interaction index. In *International Conference on Machine Learning*, pages 9259–9268. PMLR, 2020. 7, 7.1.1, 7.2, 7.2, 7.2, 7.2, 17, 7.4, 7.5.2.2

[159] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 6.3.2

[160] Joshua Brett Tenenbaum. *A Bayesian framework for concept learning*. PhD thesis, Massachusetts Institute of Technology, 1999. 6

[161] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. 1.1

[162] Che-Ping Tsai, Chih-Kuan Yeh, and Pradeep Ravikumar. Faith-shap: The faithful shapley shapley interaction index. *arXiv preprint arXiv:2203.00870*, 2022. 1.2

[163] Michael Tsang, Dehua Cheng, and Yan Liu. Detecting statistical interactions from neural network weights. *arXiv preprint arXiv:1705.04977*, 2017. 7.7

[164] Michael Tsang, Hanpeng Liu, Sanjay Purushotham, Pavankumar Murali, and Yan Liu. Neural interaction transparency (nit): Disentangling learned interactions for improved interpretability. *Advances in Neural Information Processing Systems*, 31:5804–5813, 2018. 7.7

[165] Michael Tsang, Sirisha Rambhatla, and Yan Liu. How does this interaction affect me? interpretable attribution for feature interactions. *arXiv preprint arXiv:2006.10965*, 2020. 7.7

[166] Michael Unser. A representer theorem for deep neural networks. *arXiv preprint arXiv:1802.09210*, 2018. 5.1

[167] Jasper van der Waa, Marcel Robeer, Jurriaan van Diggelen, Matthieu Brinkhuis, and Mark Neerincx. Contrastive Explanations with Local Foil Trees. In *2018 Workshop on Human Interpretability in Machine Learning (WHI)*. WHI, 2018. 1.1, 8.5

[168] S. Wachter, Brent D. Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *European Economics: Microeconomics & Industrial Organization eJournal*, 2017. 1.1, 4.4, 8.5

[169] Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. Allennlp interpret: A framework for explaining predictions of nlp models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*

and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, pages 7–12, 2019. 8.3.2

[170] Fulton Wang and Cynthia Rudin. Falling rule lists. In *Artificial Intelligence and Statistics*, pages 1013–1022. PMLR, 2015. 1.1, 8.5

[171] Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Efficient formal safety analysis of neural networks. In *Advances in Neural Information Processing Systems*, pages 6367–6377, 2018. 4.1.2

[172] Tsui-Wei 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 5273–5282, 2018. 4.1.2

[173] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101. 8.3.2, 8.4

[174] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5283–5292, 2018. 4.1.2

[175] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *arXiv preprint arXiv:1707.00600*, 2017. 5.3.2

[176] Kaidi Xu, Sijia Liu, Pu Zhao, Pin-Yu Chen, Huan Zhang, Quanfu Fan, Deniz Erdogmus, Yanzhi Wang, and Xue Lin. Structured adversarial attack: Towards general implementation and better interpretability. *arXiv preprint arXiv:1808.01664*, 2018. 1.1

[177] Fan Yang, Mengnan Du, and Xia Hu. Evaluating 'explanation without ground truth in interpretable machine learning. *arXiv preprint arXiv:1907.06831*, 2019. 6

[178] Chih-Kuan Yeh, Jianshu Chen, Chengzhu Yu, and Dong Yu. Unsupervised speech recognition via segmental empirical output distribution matching. In *International Conference on Learning Representations*, 2018. 1.2.1

[179] 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*, pages 9291–9301. NeurIPS, 2018. 1.2, 4, 8, 8.1.1, 8.5

[180] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David Inouye, and Pradeep Ravikumar. On the (in)fidelity and sensitivity of explanations. In *Neural Information Processing Systems (NeurIPS)*, pages 10965–10976. NeurIPS, dec 2019. 1.2

[181] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I. Inouye, and Pradeep Ravikumar. On the (in)fidelity and sensitivity of explanations. In *NeurIPS*, volume abs/1901.09392, pages 10965–10976, 2019. 1.1, 4, 8.5

[182] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33, 2020. 1.1, 1.2, 3, 7, 9

[183] Chih-Kuan Yeh, Been Kim, and Pradeep Ravikumar. Human-centered concept explanations for neural networks. In *Neuro-Symbolic Artificial Intelligence: The State of the Art*, pages 337–352. IOS Press, 2021. 1.2.1

[184] Chih-Kuan Yeh, Kuan-Yun Lee, Frederick Liu, and Pradeep Ravikumar. Threading the needle of on and off-manifold value functions for shapley explanations. In *International Conference on Artificial Intelligence and Statistics*, pages 1485–1502. PMLR, 2022. 1.2

[185] Chih-Kuan Yeh, Ankur Taly, Mukund Sundararajan, Frederick Liu, and Pradeep Ravikumar. First is better than last for training data influence. *Advances in neural information processing systems*, 2022. 1.2

[186] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnn explainer: A tool for post-hoc explanation of graph neural networks. *arXiv preprint arXiv:1903.03894*, 2019. 1.1

[187] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 1.1, 4, 4, 4.4, 8.5

[188] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in neural information processing systems*, pages 4939–4948, 2018. 4.1.2

[189] Huan Zhang, Pengchuan Zhang, and Cho-Jui Hsieh. Recurjac: An efficient recursive algorithm for bounding jacobian matrix of neural networks and its applications. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5757–5764, 2019. 4.1.2

[190] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015. 4.4, 8.4

[191] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2989, 2017. 4

[192] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134. ECCV, 2018. 1.1, 6, 8.5

[193] Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. ICLR, 2017. URL https://openreview.net/forum?id=BJ5UeU9xx. 1.1, 2.1.4, 4