Striving for Conceptual Soundness in Deep Learning

Zifan Wang

Department of Electrical and Computer Engineering

Carnegie Mellon University

Pittsburgh, PA, USA

April 15, 2022

Thesis Proposal

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

Committee Members:

Professor Anupam Datta	*Department of Electrical and Computer Engineering*
Professor Matt Fredrikson	*Department of Computer Science*
Professor Lujo Bauer	*Department of Electrical and Computer Engineering*
Dr. Sugato Basu	*Google*
Dr. David Alvarez-Melis	*Microsoft Research*

Contents

Abstract

Despite the great success of Deep Neural Networks (DNNs) in outperforming human-level performance in many recent tasks, they are often notorious to capture spurious correlations between input and labels, which is often not *conceptually sound*. That is, the model's decision may not build on those correct and domain-related concepts. Explainability techniques are therefore expected to provide insights in diagnosing and improving the model's quality by peering into its internal and output behaviors. In this thesis, we first discuss the connections between *explainability* and *adversarial robustness* and *generalization*, two broad axes of conceptual soundness. Second, we narrow down the discussion on domain-specific concepts that deep models are expected to master, which include 1) graph-based tasks and 2) combinatorial constraint solving. Taken together, the thesis demonstrates that faithfully explaining the model's decision is not an end but another milestone towards conceptually sound DNNs.

1 Introduction

Although Deep Neural Networks (DNNs) have greatly improved the performance of classification, recognition, and many other Machine Learning tasks in the past decades, their internal reasoning logic still remains opaque to us. Unlike linear models, the lack of transparency in DNNs makes it challenging to predict when the network will fail and how it will fail, leaving the models' decisions questionable to humans. Indeed, deep models are capable of learning spurious correlations from the data, e.g., distinguishing a wolf from a husky using the existence of snow pixels in the background [1], mistakes that human beings are likely to avoid. Ideally, we would expect a model to be accurate and *conceptually sound* (Def. 1). The aforementioned phenomena have motivated the research on *explainability*, which aims to open the black box of a DNN towards understanding, analyzing and finally improving the *conceptual soundness* of deep models.

Definition 1 (Conceptual Soundness) *The ability to reason with correct and domain-related concepts.*

In this thesis, we first explore how explanation methods can be used to diagnose conceptual soundness. In particular, we focus on *robustness* and *generalization*, two universal properties that human experts are expected to master, respectively. The goal is to discover and analyze the connections between the quality of the explanations with *robustness* and *generalization*. Later, we dives into specific concepts deep models are expected to capture for the following two domains: 1) graph-based reasoning; and 2) combinatorial constraints solving. The rest of the introduction part is organized as follows: first, Section 1.1 covers the basis of explainability techniques to discuss in this thesis; second, we briefly introduce the completed work connecting *explainability* and *robustness* in Section 1.1.1 with more details in Section 2; thirdly, we introduce the proposed work to connect *explainability* and *generalization* in Section 1.1.2 with more details in Section 3; lastly, we discuss the proposed work and ongoing work to explain and improve domain-specific tasks in Section 1.2 with further

details in Section 4.The full list of completed and proposed work is included in Section 1.2.2 together with the thesis timeline.

1.1 Deep Learning Explainability

The *explainability* of deep models can be considered as an ability to answer a set of questions regarding the behavior of the model [2], e.g., *why does this particular input lead to this particular output?* Two orthogonal directions for building explainable deep learning systems are *self-explainable models* and *post-hoc explanations*. Self-explainable models focus on the design of specific architectures such that the behavior of the network is under certain constraints, narrowing down the space of functions the network could possibly learn from the data [3, 4, 5, 6, 7, 8]. As a result, explainability is enforced before or during training. On the other hand, post-hoc explanations usually assume a pre-trained model to begin with and are therefore invariant to the training strategy and more general to different architectures.

Among post-hoc explanations, we are mainly interested in *feature attribution* in this thesis because they are usually implementation-invariant, widely adopted in many applications [9, 10, 11, 12] and publicly available for modern deep learning frameworks [13, 14], i.e. Pytorch [15] and Tensorflow [16]. We will also cover another explanation method, *counterfactual example*, in support of our findings with feature attributions separately in Section 2.3.

Feature Attribution. A feature attribution [17, 18, 19, 20, 21, 22, 23, 24, 25] takes an input and a quantity of interest, e.g. the top logit of the model's output, to calculate an importance score for each feature in the input towards that pre-selected quantity of interest. In particular, gradient-based feature attributions can be unified with the framework of *distributional influence* [20], which captures the importance of all neurons for any given layer l_j towards another successive layer l_i when the model takes input points from a neighborhood of interest, e.g., a set of points following a Gaussian distribution centered on the given input. In particular, when l_j is the input layer and l_i is the top layer, distributional

influence captures the importance of the input feature, which is the main case considered in this thesis. We will return to the discussion of distributional influence, viewing it from a geometric perspective together with its formal definition, in Section 1.1.1 and 1.1.2.

1.1.1 Connection I: Explainability and Robustness

Despite the great success of deep models to surpass the human-level performance on some tasks, they are well-known to be more vulnerable against well-crafted and tiny perturbations to the input [26], which are usually meaningless to humans. Deep models without guarantees, empirically or provably, to be robust against these perturbations are not conceptually sound because it suggests that the model is not able to leverage the correct concepts to draw the underlying decision. Instead, the model may just overfit to the spurious correlations between the input and the label. For example, recent work has found that models trained with background noise separated from training images can also be generalized to benign test images [27, 28]. In this part, we show that feature attributions are significantly influenced by the aforementioned *robustness* of the underlying model, which, therefore, is an insightful tool to examine the conceptual soundness.

Completed Work: Comparing Attributions in Standard and Robust Models In our recent work, *Robust Models Are More Inpterpretable Because Attributions Look Normal* [29], we study the cause of the observation that gradient-based feature attributions are usually more visually aligned with objects in the input image generated from robust models than those generated from standard models. We approach the problem from a geometric perspective of gradient attributions. We show that smooth decision boundaries play an important role in this enhanced interpretability, as the model's input gradients around data points will more closely align with boundaries' normal vectors when they are smooth. Therefore, because robust models have smoother boundaries, the results of gradient-based attribution methods will capture more accurate information about the nearby decision boundaries. This

work will be described in detail in Section 2.1.

Interpreting gradient-based attributions from a geometric perspective further serves as the foundation of the discussion of the quality of explanations in the rest of the thesis.

Completed Work: Attribution Robustness as A Result of Surface Smoothness Recent work also observes that not only the prediction of deep models but also the feature attributions are vulnerable to small perturbations [30, 31]. In our work *Smoothed Geometry for Robust Attribution* [32], we demonstrated that the lack of robustness for gradient-based attribution methods can be characterized by Lipschitz continuity of the gradient of the model's output w.r.t. the input, i.e. the surface smoothness in the input space. The finding is empirically demonstrated and theoretically analyzed. Towards robust gradient attributions, we proposed Smooth Surface Regularization (SSR). Our finding suggests the following take-away: the lack of robustness in attributions is not a problem of the underlying attribution method, as argued by the prior work. Instead, it is a problem of the underlying model. Towards improving the smoothness of the model's learned function can eventually lead to more robust attributions. This work will be described in detail in Section 2.2.

Completed Work: Counterfactual Consistency as A Result of Surface Continuity In our work *Consistent Counterfactuals for Deep Models* [33], we study the connection between the quality of another explanation technique, counterfactual example, and the robustness of the deep model. During the deployment of deep network, its parameters may be continuously updated due to retraining or fine-tuning for performance requirements. Therefore, counterfactuals are expected to be continuously valid to keep the instructions for *recourse* generated by the model valid for the same user. This paper studies the consistency of model prediction on counterfactual examples in deep networks under small changes to initial training conditions, such as weight initialization and leave-one-out variations in data, as often occurs during model deployment. Our analysis shows that a model's Lipschitz continuity around the counterfactual, along with confidence of its prediction, is key to its consistency

across related models. We will discuss this work in detail in Section 2.3.

1.1.2 Connection II: Explainability and Generalization

In this section, we explore *generalization*, another axis of conceptual soundness, and its connection with feature attributions. Characterization of the model performance on unseen test data focuses on deriving the generalization gap between training and test loss. The PAC-Bayesian bound [34, 35, 36], among learning theories, is shown to be a promising solution to provide non-vacuous generalization bound for deep models that align better with empirical evaluations [37]. We now introduce our proposed work motivated by the generalization gap bounded by the PAC-Bayesian.

Proposed Work: Exploring Parameter Space Smoothness with Attributions One important insight from minimizing the generalization gap from a PAC-Bayesian theory is the local smoothness of the network's training loss in the parameter space [38]. That is, small perturbations to the network's parameters should not cause significant increase of the training loss in a local neighborhood of the current parameters. With that being said, one question to the above observations is as follows: *can feature attributions capture the smoothness of the model's loss in the parameter space for analyzing the generalization gap?* The answer to this question can further lead to a training strategy that enforces both generalization and adversarial robustness through the lens of gradient regularization. This work will be discussed in detail in Section 3.2

1.2 Explanations and Domain-Specific Concepts

In the previous section, we mainly explore the connections between explainability and two types of conceptual soundness, robustness, and generalization, two major metrics that are often used across domains and applications. In this section, we explore how explanation methods can be used to analyze how well the model has learned to incorporate domain-

specific concepts from the training data. In particular, we dive into the following domains: 1) graph-based reasoning with graph neural networks; and 2) learning with theories.

1.2.1 Explainability and Graph-based Task

Graph Neural Networks (GNNs) have shown great success in reasoning with structural data that can be converted into a graph of nodes and edges, e.g. chemical compounds [39], social networks [40] and Internet of Things [41]. Whereas domain-related features in vision tasks usually refer to relevant objects in the input, the connectivity among different input nodes, i.e. features on the edges, can also encode domain-related knowledge for graph tasks. Therefore, graph models are not only expected to capture relevant node features, they are also expected to leverage relevant adjacency information for the purpose of being conceptually sound. In this section, we first discuss our work on providing faithful explanations for graph models with a focus on edge features. Secondly, we introduce the proposed work to leverage insights from our explanations to improve graph models with a focus on using the model for anomaly detection.

Completed Work: Faithfully Capturing Edge Importance for Graph Networks In our recent work *Faithful Explanations for Deep Graph Models* [42], we provide a new and general method to formally characterize the faithfulness of explanations for GNNs. Our analytical and empirical results demonstrate that feature attribution methods cannot capture the nonlinear effect of edge features, while existing subgraph explanation methods are not faithful. Based on our analysis, we introduce *k-hop Explanation with a Convolutional Core* (KEC), a new explanation method that provably maximizes faithfulness to the original GNN by leveraging information about the graph structure in its adjacency matrix and its *k-th* power. Lastly, our empirical results over both synthetic and real-world datasets for classification and anomaly detection tasks with GNN demonstrate the effectiveness of our approach. This work will be discussed in detail in Section 4.1.

Proposed Work: Anomaly Detection using Graph Models with Influential Edges Graph networks have shown advantages over other machine learning models in detecting anomalous behaviors: Because they can encode the connectivity information explicitly as adjacency information, the node-level anomalous behavior can be located and only neighbors in the corresponding computation graph may be responsible. In our work with the KEC explanation, we have shown a case study on finding the causes of anomalous nodes. By locating the most influential edges of the predicted anomalous nodes and following them to find those influential neighbors, we observe that, for most false negative cases, the most influential edges can actually lead back to the nodes under the attack, though these nodes do not appear in the model's prediction. Building on findings above, we seek to incorporate the insights from graph explanations back to GNNs to improve the performance on anomaly detection. This work will be discussed in detail in Section 4.2.

1.2.2 Enforcing Explainability by Learning with Theories

In addition to improving conceptual soundness with better training objectives, as discussed in previous sections, recent work also aims to bake domain-related knowledge directly into the model architecture [3, 4, 5, 6, 7, 8]. Thus, these models are also self-explainable because their internal behaviors are under constraints. One possible way to enforce constraints during training is to combine deep networks together with combinatorial constraint solvers, e.g., Z3 [43]. We now introduce the following on-going work.

On-going Work: Explainability and Constraints Solving In this work, our objective is to combine a combinatorial constraint solver as a *differentiable* layer. A solver is encoded a logically sound theory, e.g. additions between numbers and Sudoku's rules, while the deep network is used to extract relevant features for the solver, e.g. digits from images, to meet the satisfaction condition that the solver requires. When the extracted features are unable to make the solver produce outputs that satisfy the theory, the solver will provide constructive

feedback to the network to improve the feature extraction pipeline. Towards that end, we aim to supervise the model's learning procedure by a theory that correctly incorporates the domain-related concepts. This work will be discussed in detail in Section 4.3.

Summary of Completed and Proposed Work

In summary, we present the following completed work and propose additional tasks in support of this thesis:

- **Completed**: Comparing Attributions in Standard and Robust Models [29] (Section 2.1)
- **Completed**: Attribution Robustness as A Result of Surface Smoothness [32] (Section 2.2)
- **Completed**: Counterfactual Consistency as A Result of Surface Continuity [33] (Section 2.3)
- **Completed**: Faithfully Capturing Edge Importance for Graph Networks [42] (Section 4.1)
- **On-going**: Training Deep Models with A Combinatorial Constraint Solver (Section 4.3)
  Expected Completion: February 2023
- **Proposed**: Exploring Parameter Space Smoothness with Attributions (Section 3.2)
  Expected Completion: August 2022
- **Proposed**: Anomaly Detection using Graph Models with Influential Edges (Section 4.2)
  Expected Completion: February 2023
- **Expected Defense Date**: April 2023

2 Explainability and Robustness

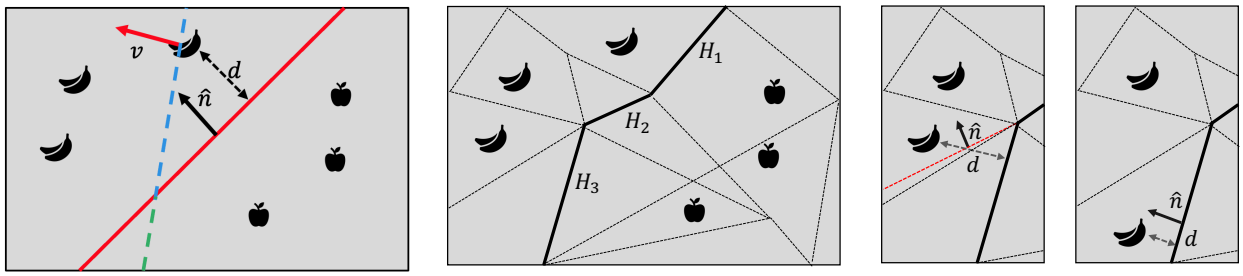
In this section, we provide a set of formal and empirical analysis between explanation methods, i.e. feature attributions and counterfactual examples, and the *adversarial robustness* of deep models. In Section 2.1, we demonstrate that gradient-based attributions better capture the normal vectors of the nearby decision boundaries. In Section 2.2 and 2.3, we show how the Lipschitz conditions of the model's output w.r.t. its input influence the quality of feature attribution and counterfactual examples.

2.1 Comparing Attributions in Standard and Robust Models

Feature attribution methods assign an importance score for each input feature towards the model's output confidence, e.g. the logit of the predicted class. The target question that a feature attribution aims to answer is *why does the model predict the input into a particular class?*. In this work, *Robust Models Are More Inpterpretable Because Attributions Look Normal* [29], we aim to use feature attributions to capture the model's decision boundary. Feature attribution methods considered in this work can be formalized by *distributional influence* [20] (Def. 2), an aggregation of the gradients w.r.t the given input and a set of its neighbors. Distributional influence generalizes many well-known methods, e.g. Saliency Map [22], Integrated Gradient [19] and Smooth Gradient [21].

Definition 2 (Distributional Influence [20]) *Suppose $f(x)$ is the logit output of the class of interest and a distribution of interest $\mathcal{D}_x$ that describes a reference neighborhood around the input x, distributional influence χ is defined as follows: $\chi(x, f, \mathcal{D}_x) \stackrel{\text{def}}{=} \mathbb{E}_{z\sim\mathcal{D}_x} \nabla_z f(z)$.*

Recent work [44, 45] has observed that the distributional influence in adversarially robust image models, when visualized, tends to be more interpretable: attributions are more clearly aligned with the discriminative portions of the input. In this work, we build on a geometric understanding of robustness and interpretability. Calculating the gradient w.r.t. the input,



(a) (b) (c)

Figure 1: Different classifiers that partition the space into regions associated with `apple` or `banana`. (a) A linear classifier where $\hat{\mathbf{n}}$ is the only faithful explanations and $\mathbf{v}$ is not. (b) A deep network with ReLU activations. Solid lines correspond to decision boundaries while dashed lines correspond to facets of activation regions. (c) Gradient of the target instance may be normal to the closest decision boundary (right) or normal to the prolongation of other local boundaries (left).

that is, $\nabla_x f(x)$, is the core step in Distributional Influence. Thus, we begin with a discussion of how input gradients capture a classifier's decision boundary.

Gradient and Decision Boundary. Consider a linear model $C(x) = \text{sign}(w^\top x + b)$, a decision boundary is a hyperplane that separates the input space into two half-spaces. Accordingly, the normal vector of the decision boundary is the only vector that faithfully explains the model's classification (Fig. 1a). In deep networks with piecewise linear activation functions, i.e., ReLU, the decision boundary is a set of piecewise linear hyperplanes (Fig. 1b). While gradients always point to the closest point on the decision boundary in a linear model, this may not always be the case in the deep network, as observed in the literature [46, 47] (Fig. 1c).

Boundary-Based Attributions. To capture the normal vectors from the nearest or nearby boundaries, we intend to modify the distribution of interest D_x in Def 2. Towards that end, we introduce Boundary-based Integrated Gradeint (BIG) in this work, which aggregates the input gradients of points uniformly distributed over a linear path between the closest point on the decision boundary and the given input. Although solving the exact clos-

CIFAR10	standard	$\ell_2\|0.5$		
IG-BIG	31.22	2.73		
ImageNet	standard	$\ell_2\|3.0$	$\ell_\infty\|\frac{4}{255}$	$\ell_\infty\|\frac{8}{255}$
IG-BIG	17.07	0.69	1.74	1.45

Table 1: ℓ_2 differences between IG and BIG. The heading of each column reports the respective training epsilon and the corresponding ℓ_p norm constraint.

ImageNet	Metrics	BIG	SG	IG
standard	Loc.	**0.38**	0.34	0.34
	EG	0.54	**0.55**	0.5
	PP	**0.87**	0.50	0.51
	Con.	**4.35**	4.06	3.97
$\ell_2\|3.0$	Loc.	**0.39**	0.34	0.33
	EG	**0.74**	0.62	0.65
	PP	**0.92**	0.51	0.65
	Con.	**5.03**	4.23	4.37

Table 2: Metrics of localizing objects over 1500 images of IamgeNet using a standard and robust ResNet50 (training ϵ is reported in the first column). BIG: Boundary-based Integrated Gradient. SG: Smoothed Gradient [21]. IG: Integrated Gradient [19].

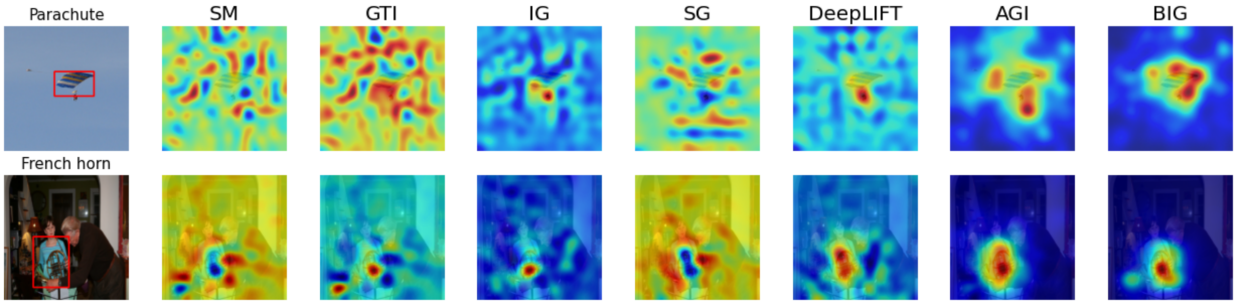


Figure 2: Visualizations of attributions for two examples classified by a standard ResNet50. BIG: Boundary-based Integrated Gradient; AGI: Adversarial Gradient Integration [49]; SM: Saliency Map [22]; GTI: `grad×input`; SG: Smoothed Gradient [21]; IG: Integrated Gradient [19]; DeepLIFT [50].

est point on the decision boundary is NP-hard and not feasible for modern architectures, we approximate the boundary search by adversarial attacks, e.g. Projected Gradient Descend (PGD) [48].

Results. In Fig. 2, we include two instances from a standard ResNet50 model to visually inspect the attributions. We show the bounding box of the target object in the input. In Table 1, we calculate the ℓ_2 distances between the BIG and its non-boundary counterpart, i.e. Integrated Gradient (IG). We find that in robust models (obtained from PGD training [48]),

BIG and IG are more similar and the similarity increases as the model is trained with a larger robust radius. In Table 2, we measure how well BIG localizes the relevant objects in the images using four localization metrics (Loc., EG, PP and Con.) [29], which shows that BIG outperforms other methods in both standard and robust models. Taken together, we show that standard attributions in robust models are visually more interpretable because they better capture the nearby decision boundaries. Therefore, the final take-away is that if more resources are devoted to training robust models, effectively identical explanations can be obtained using much less costly standard gradient-based methods, i.e. IG.

Related Work

Our analysis suggests that we need to capture decision boundaries in order to better explain classifiers, while a similar line of work, AGI [49], also involves computations of adversarial examples is motivated to find a non-linear path that is linear in the representation space instead of the input space compared to IG. The idea of using boundaries in the explanation has also been explored by T-CAV [51], where a linear decision boundary is learned for internal activations and associated with their proposed notion of *concept*. When viewing our work as using nearby boundaries as a way of exploring the local geometry of the model's output surface, a related line of work is NeighborhoodSHAP [52], a local version of SHAP [53]. When viewing our method as a different use of adversarial examples, some other work focuses on counterfactual examples (semantically meaningful adversarial examples) on the data manifold [54, 55, 56].

2.2 Robust Attributions and Surface Smoothness

When using feature attributions to explain the model's decision, it is naturally necessary for the explanations to be consistent for similar input when the model outputs the same decisions. In our work *Smoothed Geometry for Robust Attribution* [32], we demonstrated that the lack of robustness for gradient-based attribution methods can be characterized by

the Lipschitz continuity of the gradient of the model's output w.r.t. the input (Def. 3).

Definition 3 (Attribution Robust) *Suppose $f(x)$ is the logit output of the class of interest of a network and a distributional influence $\chi(x, f, \mathcal{D}_x)$ as defined in Def. 2, $\chi(x, f, \mathcal{D}_x)$ is (ϵ, λ)-locally robust if $\chi(x, f, \mathcal{D}_x)$ if (ϵ, λ)-locally Lipschitz continuous. That is, $\forall x'$ such that $||x - x'||_2 \leq \epsilon, ||\chi(x, f, \mathcal{D}_x) - \chi(x, f, \mathcal{D}'_x)||_2 \leq \lambda ||x - x'||_2$.*

Lipschitz Gradient During Training. Def. 3 shows that the key to enforce similar input gradients, which should eventually lead to similar distributional influence, we need to have Lipschitz gradient. One way to capture the continuity of the gradient is to analyze the maximum eigenvalue of the corresponding Hessian matrix. The approximation comes from the fact we need to remove other higher order terms in the Taylor's expansion of a function. Towards this end, we propose Smooth Surface Regularization (SSR) in Def. 4 to penalize the maximum eigenvalue of the Hessian during the training time to improve the continuity of the model's gradient.

Definition 4 (Smooth Surface Regularization (SSR)) *Given data pairs $(\mathbf{x}, y)$ drawn from a distribution $\mathcal{D}$, the training objective of SSR is given by $\min_{\boldsymbol{\theta}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathcal{L}((\mathbf{x}, y); \boldsymbol{\theta}) + \beta s \max_i \xi_i]$ where $\boldsymbol{\theta}$ is the parameter vector of the the model f, $\max_i \xi_i$ is the largest eigenvalue of the Hessian matrix $\tilde{\mathbf{H}}_{\mathbf{x}}$ of the regular training loss $\mathcal{L}$ (e.g. Cross Entropy) w.r.t to the input. β is a hyper-parameter for the penalty level and s ensures the scale of the regularization term is comparable to regular loss.*

Results. We measure the cosine distance between attributions before and after two attribution attacks, Top-K [30] and Manipulate [58] in Fig. 3. Lower cosine distance scores indicate better attribution robustness. We summarize the findings: 1) compared with results on the same model with natural training, SSR provides much better robustness nearly on all the metrics for all attribution methods; 2) SSR provides comparable and sometimes

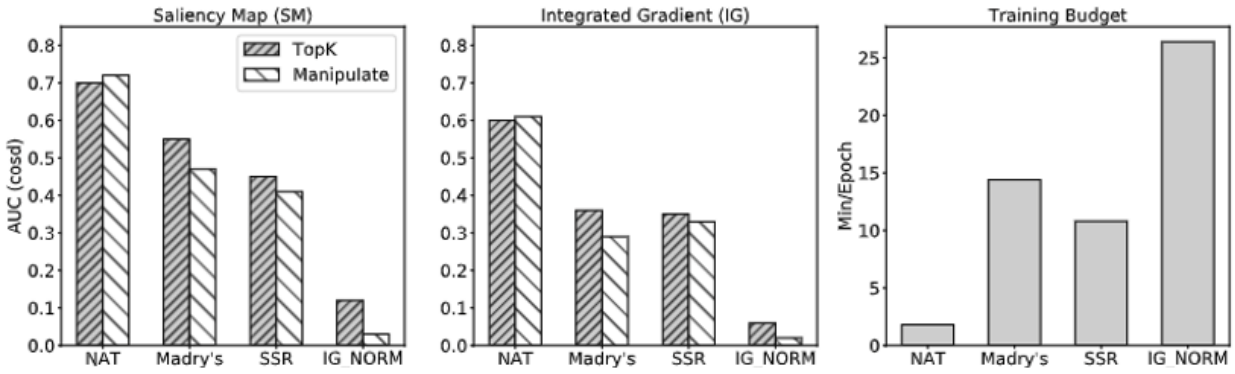


Figure 3: Consine distance (cosd) of two attribution attacks, Top-k and Manipulate attack, on CIFAR-10 with different training algorithm – NAT: natural training without any regularization's; Madry's [48] and IG-NORM [57]. Lower cosd indicates better attribution robustness.

even better robustness compared to Madry's adversarial training [48]; and 3) Though IG-NORM [57] provides the best performance, it has high costs of training time and accuracy, while SSR and Madry's training have lower costs.

Related Work

Smoothing the geometry with Hessians has been widely adopted for different purpose. We use Singla et al' s Closed-form Formula given that it is the first approach that returns Hessian approximation for ReLU networks [59]. There are other approaches that approximate the Hessian's spectrum norm, e.g. by using Frobenius norm and the finite difference of Hessian-vector product [60] and by regularizing the spectrum norms of weights [61]. Besides, more recent work has also demonstrated an even stronger correlation between training objectives towards robust predictions and attributions [62]. On the other hand, activation functions also play an important role in learning smoothed functions as pointed in the literature [58, 29].

2.3 Counterfactual Explanations and Model Lipschitzness

Previous sections have thoroughly discussed the connection between gradient-based attribution approaches and their connection to the adversarial robustness through Lipschitz

continuity. In our recent work, *Consistent Counterfactuals for Deep Models* [33], we find the similar connections also exist for another line of explanation method: counterfactual examples [63, 64, 65, 66, 67]. A counterfactual example is often considered as a related (and usually nearby) point that produces a desired (and usually opposite) outcome, e.g., the nearest potential accepted loan application to a rejected loan application.

Counterfactuals are often discussed under the assumption that the model on which they will be used is static, but in deployment models may be periodically retrained or fine-tuned. We thus study the consistency of model prediction on counterfactual examples in deep networks under small changes to initial training conditions.

Consistent Counterfactual from Consistent Boundaries. We analyze the differences between models with changes such as random initialization by studying the differences that arise in their decision boundaries. In order to capture information about the decision boundaries in analytical form, we leverage *distributional influence* (Def. 2). Our discussion in Section 2.1 have suggested that input gradients are normal vectors to decision boundaries, which may not be the nearest but are in the vicinity introduced by the distribution of interest. Therefore, the change of *distributional influence* can capture the change of decision boundaries. Notice that we are discussing a different setup here where we measure the similarity of distributional influence when varying the input in Section 2.1 and 2.2; however, this section considers the similarity when varying the trainable parameters. We now directly summarize our finding from this perspective: the change of distributional influence is upper-bounded by a constant $\lambda \sim O(K/\sigma)$ where K is the local Lipschitz constant of the model's output and σ is the probit output.

Robust Neighbor Search. One natural take-away from the observation above is that one can find another counterfactual example x_c nearby such that: 1) x_c has higher probit output and 2) the model is more locally Lipschitz around x_c. Directly optimizing over the local Lipschitz constants may encounter with vacuous second-order derivatives in ReLU networks.

Invalidation Rate

Dataset	German Credit		Seizure		CTG		Warfarin		HELOC		Taiwanese Credit	
Methods	LOO	RS	LOO	RS	LOO	RS	LOO	RS	LOO	RS	LOO	RS
Min. ℓ_1 [63]	0.41	0.56	-	-	0.07	0.29	0.44	0.35	0.30	0.43	0.30	0.78
+SNS	0.00	0.07	-	-	**0.00**	0.01	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.04
Min. ℓ_2 [63]	0.36	0.56	0.64	0.77	0.48	0.49	0.35	0.3	0.55	0.61	0.27	0.72
+SNS	0.00	**0.06**	**0.02**	0.13	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.04**
Min. ϵ PGD [48]	0.28	0.61	0.94	0.94	0.04	0.09	0.10	0.12	0.04	0.11	0.04	0.24
+SNS	**0.00**	0.12	0.04	0.16	**0.00**	**0.00**	0.01	0.02	**0.00**	**0.00**	**0.00**	0.11
Looveren et al. [64]	0.25	0.40	0.48	0.54	0.11	0.18	0.26	0.25	0.25	0.34	0.29	0.53
Pawelczyk et al. [68]	0.20	0.35	0.16	**0.11**	**0.00**	0.06	0.02	0.01	0.05	0.15	0.02	0.20

Table 3: The consistency of counterfactuals measured by invalidation rate [33]. Lower invalidation rates are more desirable. Results are aggregated over 100 networks for two different retraining method: random seed (RS) and leave-one-out change to the training set (LOO).

The local Lipschitz constant is the supreme of the gradient norm for some neighborhood. In this work, we use a lower-bound of the actual local Lispchitz constant by only considering the gradient norms for points on a linear path. Our lower-bound approximation further leads to our proposed method, Stable Neighbor Search (SNS) in Def. 5.

Definition 5 (Stable Neighbor Search (SNS)) *Given a starting counterfactual x for a network $f(x)$, its* stable neighbor *x_c of radius ϵ is the solution to the following objective: $x_c = \arg\max_{||x'-x||_2 \leq \epsilon} \int_0^1 \sigma(tx')dt$ where $\sigma(\cdot)$ is the sigmoid function.*

Results. We measure the *invalidation rate*, the percentage of invalid counterfactuals, under two small changes to the training process, i.e. random seed (RS) and leave-one-out change to the training set (LOO), in Table 3. SNS achieves the lowest invalidation rate across all datasets in both LOO and RS experiments, except for on the Seizure dataset with RS variations, where there is a two-point difference in the invalidation rate. SNS generates counterfactuals with *no* invalidation on CTG, Warfarin, and Heloc, and no invalidation over LOO differences on German Credit and Taiwanese Credit.

Related Work

Counterfactual examples enjoy popularity in the research literature [69, 63, 70, 71, 64, 65, 72, 66, 67, 73, 74], especially in the wake of legislation increasing legal requirements on explanations of machine learning models [75, 76]. Previous work investigating the problem of counterfactual invalidation [68, 77] has pointed to increasing counterfactual cost as a potential mitigation strategy in linear models; however, this insight might not be always true for deep networks given the non-linearity of decision boundaries. That is, moving away from one boundary can lead to another one.

3 Explainbility and Generalization

3.1 Generalization Gap

In this section, we examine the connection between the explainability with another axis of conceptual soundness, i.e., generalization. As humans can easily generalize from given instances to unseen data, it is expected that a deep model will learn similar concepts as humans do. From a perspective of learning theory, recent work aims to analytical bound the generalization gap, i.e. the difference between the training loss and the test loss, among which PAC-Bayesian [34, 35, 36] has shown promising tightness when evaluated in practice [37]. Def. 6 describes McAllester's way [34] of formalizing the PAC-Bayesian bound for the generalization gap.

Definition 6 (PAC-Bayesian Bound [34]) *Suppose P is the prior distribution of the parameters of a network independent of the data and W is the corresponding posterior distribution of parameters learned from the training set S with m samples, respectively. Let L be the 1-0 error of a classifier. For any $\delta \in (0, 1]$, and the test distribution D,*

$$\mathbb{E}_W \left[\mathbb{E}_{(x,y)\sim D}[L((x, y); W)]\right] \leq \mathbb{E}_W \left[\mathbb{E}_{(x,y)\sim S}[L((x, y); W)]\right] + O(KL(W||P), \frac{1}{\delta}, \frac{1}{m}) \quad (1)$$

Although PAC-Bayesian bounds the generalization error over the posterior distribution of the parameters instead of a static set of trained weights, its insight into closing the gap lies in the residual terms $O(KL(W||P), \frac{1}{\delta}, \frac{1}{m})$. With a closer look, Foret el al. [38] derives a trainable objective to explicitly minimize the generalization gap, which includes minimizing the maximum training loss when the current weights vary in a small ℓ_2 ball and the ℓ_2 norm of the weights. Foret el al. [38]'s objective therefore motivates our proposed work in this section.

3.2 Proposed Work: Does Generalization Indicate Explainability

The aforementioned objective is directly related to *smoothness* and *continuity* of the training loss in the parameter space, since similar objectives in training models with robust predictions and attributions have been shown to improve the *smoothness* and *continuity* of the model's output in the input space in Section 2. The proposed work thus aims to investigate the following research questions.

RQ1 How does the gradient update in adversarial training connect and differ from closing the generalization gap using Foret el al.'s objective?

RQ2 To follow up with RQ1, do smoothing techniques in the input space, i.e. spectral norm regularization, smooth the parameter space?

RQ3 Finally, do models that generalize better provide feature attributions that are more aligned with domain-related concepts?

Answers to RQ1 and RQ2 are important steps towards the answer to RQ3 as we have observed that robust models do provide feature attributions that are aligned with expected concepts. The answers to the above question can further lead to a training strategy that enforces both generalization and adversarial robustness through the lens of gradient regularization. This work will be done as the internship project in Google for the summer of 2022.

4 Explainability and Domain-Specific Concepts

Section 2 and 3 have focused on *robustness* and *generalization*, two axes of achieving *conceptual soundness* that are universal for many deep learning applications. In addition to the aforementioned metrics, when deep models are deployed to provide a specific service, more specific concepts often exist and are often more important to learn from the data. For example, in a graph-related task, it is necessary to utilize the graph structure because information does not always come from node features. In this section, we dive into several specific domains to 1) adapt attribution methods with domain-specific knowledge (Section 4.1); 2) improve the models with insights from explanations (Section 4.2); and 3) enforce the model to learn with a domain-specific theory (Section 4.3).

4.1 Faithful Graph Explanations

As feature attributions have been widely applied in tasks other than image classifications, capturing the importance of the input feature is often not enough to evaluate whether the model is capable of capturing some specific domain-related concepts. This section focuses on our recent work, *Faithful Explanations for Deep Graph Models* [42], that generalizes feature attributions to discover the importance of connectivity in graph data[1]. That is, our objective is to faithfully attribute the output of Graph Neural Network (GNN) over the edges of the input data.

Graph Neural Network (GNN). We first give a brief introduction to Graph Neural Networks (GNNs). A GNN takes a tuple of $(X \in \mathbb{R}^{n \times d}, A \in \mathbb{R}_{+}^{n \times n})$ as input, where X is the node feature matrix with n nodes and d features, and the connectivity between nodes is stored with the adjacency matrix A. To pass information between nodes, graph convolution [79] is one of the most popular options. Finally, the output of a GNN is task-dependent and in this

[1] On the other hand, we also generalize distributional influence for discovering language contextualization in another recent work [78].

	BA-shape			Cora			CiteSeer		
Neighborhood	$\mathcal{U}(X, 0.2r)$	$\mathcal{U}(A, 0.5)$	$\mathcal{B}(A, 0.5)$	$\mathcal{U}(X, 0.2r)$	$\mathcal{U}(A, 0.5)$	$\mathcal{B}(A, 0.5)$	$\mathcal{U}(X, 0.2r)$	$\mathcal{U}(A, 0.5)$	$\mathcal{B}(A, 0.5)$
GNNExpl [80]	-	0.26	1.29	0.07	3.81	0.53	0.03	3.25	0.60
GNNExpl (soft)	-	0.28	1.35	0.02	0.09	0.11	0.01	0.17	0.21
PGExpl [81]	-	0.25	0.68	-	1.35	0.75	-	0.29	0.48
SM [22]	-	0.19	1.57	0.02	7.26×10^{-4}	0.38	1.40×10^{-3}	3.70×10^{-4}	0.23
IG (zero) [19]	-	1.59	6.49	0.03	0.21	0.74	2.16×10^{-3}	0.10	0.43
IG (random)	-	0.29	1.48	0.03	3.93×10^{-3}	0.40	0.02	1.90×10^{-3}	0.26
Linear [83]	-	**0.13**	**0.33**	0.09	2.3×10^{-3}	0.45	0.04	7.86×10^{-4}	0.16
KEC (ours)	-	**0.13**	0.50	$\mathbf{3.80 \times 10^{-4}}$	$\mathbf{4.25 \times 10^{-4}}$	**0.06**	$\mathbf{6.54 \times 10^{-4}}$	$\mathbf{1.80 \times 10^{-4}}$	**0.04**

Table 4: General Unfaithfulness $\Delta(p)$ [42] for different explanation methods when node and edges are perturbed separately. $\mathcal{U}(X, 0.2r)$ corresponds to perturbations to node features while $\mathcal{U}(A, 0.5)$ and $\mathcal{B}(A, 0.5)$ correspond to perturbations to edges. Lower scores are better.

work, we consider node classification. Therefore, we write $y = \arg\max_c \{e_v^\top f(X, A)\}_c$ where f is a stack of graph convolution and dense layers, separated by ReLU activations, and $e_v^\top$ selects the prediction of the target node.

Non-linear Explanations for GNN. The key contribution of our work is to provide a non-linear and interpretable function that approximates the GNN's output in a neighborhood of edge features (with the ability to generalize to node features), whereas the prior work either only captures the GNN's output with a linear function [20, 22, 19] or less faithful non-linear ones [80, 81, 82]. Formally, we introduce KEC in Def. 7. Finally, given a GNN and its KEC explanation $p(X, A)$, we compute $\nabla_A p(X, A)$ to obtain the importance of each edge. The faithfulness of KEC w.r.t the target GNN model is guaranteed by the trainable parameter w in Def. 7, which is optimized to minimize *general unfaithfulness*, i.e., the difference between the change of $p(X, A)$ in a reference neighborhood and the change of original GNN in the same neighborhood.

Definition 7 (K-hop Model with a Convolutional Core (KEC)) *Suppose $F(X, A)$ has M graph convolution layers, a k-hop local difference model $p(X, A)$ to explain F is defined as $p(X, A) = \sum_{k=1}^{M} e^\top N(A^k) X w_k$ where $N(\cdot)$ is the normalization for the adjacency matrix, and $w_k (k = 1, ..., M)$ are trainable parameters.*

Results. We compare the faithfulness of the proposed method KEC in capturing the importance of the node and edge compared to other explanation methods in Table 4 where lower scores indicate a better faithfulness. We make the following observations from the table: 1) KEC is uniformly better than all baselines on real datasets, i.e. Cora and Citeseer, while it matches or is slightly worse than Linear on BA-shapes; 2) Saliency Map (SM) is much faithful than most other baselines for explaining the graph models while Integrated Gradient (IG) is really sensitive to the choice of baselines. A common choice of baseline, a zero vector, from the vision task does not appear to be a reasonable baseline in the graph case.

Related Work

The notation of *general unfaithfulness* is based on the INFD score [83], which is a special case for gradient-based attributions that provide a linear local difference model, as shown in our work [42]. The baseline explanation methods chosen in this work are *subgraph explanations* [80, 81, 82], which aims to search for a subset of nodes and edges that account for the behavior of the model. Another recent work [84] also finds IG and GradCAM [23] to be more correlated with the model's performance from the perspective of perturbing the training set of underlying GNN, compared to subgraph explanations. However, KEC has been shown to even outperform IG in our work.

4.2 Proposed Work: Improving Anomaly Detection by Diving into Explanations

Anomaly detection as one of the target tasks in applying GNN in real-word cases. We have conducted a case study using faithful explanations to locate anomalous sensor readings from the secure water treatment system (SWAT) [85] dataset. We aim to attribute the model's prediction to flagging the anomaly in the system over edge features, and visualizations are shown in Fig. 4. We plot the visualization of three attack instances explained by different

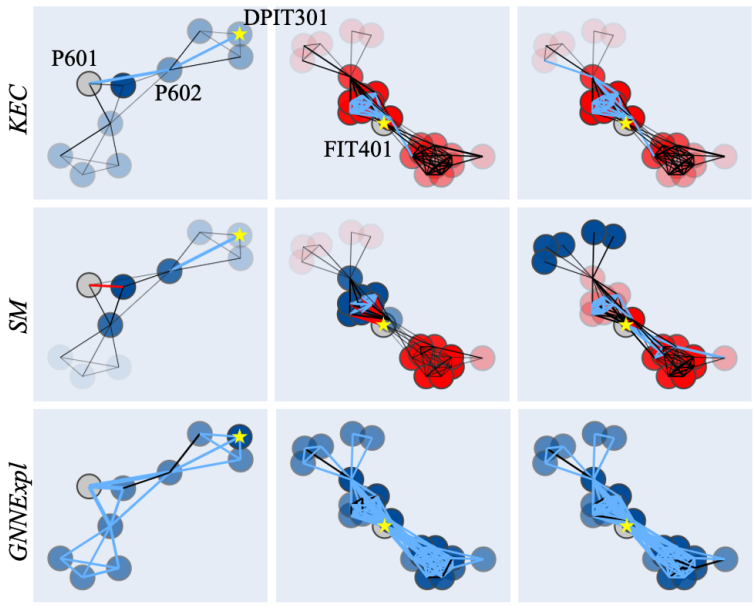


Figure 4: Visualization of graph explanations for SWAT anomaly detection using different methods for 3 attacks (left to right). Grey nodes are abnormal nodes reported by GNN (P601, FIT401, FIT401), while yellow stars are nodes under attack in ground truth (DPIT301, FIT401, FIT401). The importance of nodes and edges is colored with red (positive) and blue (negative) and their magnitudes are shown by opacity (for nodes) and width (for edges).

methods. Images of the same column are from the same attack. The actual node under attack is marked with yellow starts, whereas the model predictions are colored gray. The blue nodes and edges are positive influences, which makes the model believe that the gray node has an anomalous reading, while the red nodes are negative. Among the three attacks, we find several interesting insights: (1) The model incorrectly attributes nodes for attack #1. The attack spoofs the pressure sensor (DPIT301) readings to repeatedly trigger the backwash process (controlled by pump P602). As a result, the model predicts that a correlated pump (P601) should also have a low speed reading well below its normal operating range. Looking at the visualizations, KEC clearly shows the influence of DPIT301–P602–P601 with blue edges. (2) Both attacks #2 and #3 directly set the value of the flow meter FIT401 as 0 to shut down subsequent processes. Since they are the same attacks, their explanations should reflect similarity as well. Surprisingly, SM generates inconsistent explanations for these two attacks. We find that the disagreement mainly focuses on the top left group w.r.t. the stared

node, which provides us a direction to investigate the model's behavior.

Building on the observations above, we therefore propose this work to incorporate the take-aways from KEC to improve the anomaly detection on SWAT and other similar systems. In particular, our goal is to find ways to update the adjacency matrix to move the nodes closer that appear to be more related in KEC. We are also interested in extracting a rule-based model from a graph model. For example, a soft decision tree [86] by placing nodes with higher KEC influence closer to the root.

4.3 On-going Work: Explainability and Constraints Solving

The working logic from Section 4.1 to Section 4.2 is to first extract explanations from graph models. Secondly, we improve the model's performance with the aid of explanations. In this section, we discuss our on-going work that aims to directly enforce a particular theory into the training of deep models. We constrain the concepts that the model can learn from the data distribution. As a result, the model becomes more transparent by construction. Our approach is two-fold: 1) the model extracts the features that can satisfy the provided theory; and then 2) the model makes the inference with the given theory.

Our approach to enforce a theory into the training process is to add a combinatorial constraint solver, e.g. Z3 [43], as a differentiable layer into the network. In this work, we begin with a theory that encodes the Boolean satisfaction problem, e.g., $x_1 \wedge x_2$, and a corresponding solver verifies if there is an assignment of the variables that satisfy the problem, i.e., `SAT`. Propagating gradients over a constraint solver has received increasing attention in recent work [87, 88]. Distinctively, our work aims to use *constructive* feedback, e.g., unsat cores, from the solver to update the feature extractor part of the network when the solver needs to output `UNSAT` for the problem. Direct applications to our work include the addition of digits in images, Sudoku, the travel salesman problem, and many other problems that require a logical way of encoding the problem, which has been shown to be extremely hard if the network is monitored directly with the desired output [88].

References

[1] M. T. Ribeiro, S. Singh, and C. 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*, pp. 1135–1144, 2016.

[2] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pp. 80–89, IEEE, 2018.

[3] R. Wang, X. Wang, and D. I. Inouye, "Shapley explanation networks," in *ICLR*, 2021.

[4] Y. Wang and X. Wang, "Self-interpretable model with transformationequivariant interpretation," *ArXiv*, vol. abs/2111.04927, 2021.

[5] D. Alvarez-Melis and T. Jaakkola, "Towards robust interpretability with self-explaining neural networks," in *NeurIPS*, 2018.

[6] E. Dai and S. Wang, *Towards Self-Explainable Graph Neural Network*. 2021.

[7] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang, "Concept bottleneck models," in *International Conference on Machine Learning*, pp. 5338–5348, PMLR, 2020.

[8] D. Rajagopal, V. Balachandran, E. H. Hovy, and Y. Tsvetkov, "Selfexplain: A self-explaining architecture for neural text classifiers," in *EMNLP*, 2021.

[9] G. Erion, J. D. Janizek, P. Sturmfels, S. M. Lundberg, and S.-I. Lee, "Improving performance of deep learning models with axiomatic attribution priors and expected gradients," *Nature machine intelligence*, 2021.

[10] S. Jayaram and E. Allaway, "Human rationales as attribution priors for explainable stance detection," in *EMNLP*, 2021.

[11] L. Rieger, C. Singh, W. J. Murdoch, and B. Yu, "Interpretations are useful: penalizing explanations to align neural networks with prior knowledge," in *ICML*, 2020.

[12] P. Schramowski, W. Stammer, S. Teso, A. Brugger, X. Shao, H.-G. Luigs, A.-K. Mahlein, and K. Kersting, "Making deep neural networks right for the right scientific reasons by interacting with their explanations," *Nature Machine Intelligence*, vol. 2, pp. 476–486, 2020.

[13] K. Leino, Rshih32, D. Gopinath, Anupam Datta, Shayak Sen, MacKlinkachorn, Kaiji Lu, and Zifan Wang, "truera/trulens: Trulens," 2021.

[14] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson, "Captum: A unified and generic model interpretability library for pytorch," 2020.

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[16] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[17] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," in *BMVC*, 2018.

[18] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3449–3457, 2017.

[19] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3319–3328, JMLR. org, 2017.

[20] K. Leino, S. Sen, A. Datta, M. Fredrikson, and L. Li, "Influence-directed explanations for deep convolutional networks," in *2018 IEEE International Test Conference (ITC)*, pp. 1–8, IEEE, 2018.

[21] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," 2017.

[22] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 2013.

[23] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, p. 336–359, Oct 2019.

[24] M. T. Ribeiro, S. Singh, and C. 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, San Francisco, CA, USA, August 13-17, 2016*, pp. 1135–1144, 2016.

[25] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," in

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp. 24–25, 2020.

[26] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, vol. abs/1412.6572, 2015.

[27] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[28] H. Wang, X. Wu, Z. Huang, and E. P. Xing, "High-frequency component helps explain the generalization of convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[29] Z. Wang, M. Fredrikson, and A. Datta, "Robust models are more interpretable because attributions look normal," 2021.

[30] A. Ghorbani, A. Abid, and J. Y. Zou, "Interpretation of neural networks is fragile," in *AAAI*, 2019.

[31] A.-K. Dombrowski, M. Alber, C. J. Anders, M. Ackermann, K.-R. Müller, and P. Kessel, "Explanations can be manipulated and geometry is to blame," in *NeurIPS*, 2019.

[32] Z. Wang, H. Wang, S. Ramkumar, P. Mardziel, M. Fredrikson, and A. Datta, "Smoothed geometry for robust attribution," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds.), 2020.

[33] E. Black, Z. Wang, and M. Fredrikson, "Consistent counterfactuals for deep models," in *International Conference on Learning Representations*, 2022.

[34] D. A. McAllester, "Some pac-bayesian theorems," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, (New York, NY, USA), p. 230–234, Association for Computing Machinery, 1998.

[35] A. Maurer, "A note on the pac bayesian theorem," 2004.

[36] O. Catoni, "Pac-bayesian supervised classification: The thermodynamics of statistical learning," 2007.

[37] Y. Jiang*, B. Neyshabur*, H. Mobahi, D. Krishnan, and S. Bengio, "Fantastic generalization measures and where to find them," in *International Conference on Learning Representations*, 2020.

[38] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," in *International Conference on Learning Representations*, 2021.

[39] S. Harada, H. Akita, M. Tsubaki, Y. Baba, I. Takigawa, Y. Yamanishi, and H. Kashima, "Dual graph convolutional neural network for predicting chemical networks," *BMC Bioinformatics*, vol. 21, no. 3, p. 94, 2020.

[40] W. Fan, Y. Ma, Q. Li, Y. He, Y. E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," *The World Wide Web Conference*, 2019.

[41] G. Dong, M. Tang, Z. Wang, J. Gao, S. Guo, L. Cai, R. Gutierrez, B. Campbell, L. E. Barnes, and M. Boukhechba, "Graph neural networks in iot: A survey," 2022.

[42] Z. Wang, Y. Yao, C. Zhang, H. Zhang, Y. Kang, C. Joe-Wong, M. Fredrikson, and A. Datta, "Faithful explanations for deep graph models," 2022.

[43] L. De Moura and N. Bjørner, "Z3: An efficient smt solver," in *Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, TACAS'08/ETAPS'08, p. 337–340, Springer-Verlag, 2008.

[44] F. Croce, M. Andriushchenko, and M. Hein, "Provable robustness of relu networks via maximization of linear regions," *AISTATS 2019*, 2019.

[45] C. Etmann, S. Lunz, P. Maass, and C. Schoenlieb, "On the connection between adversarial robustness and saliency map interpretability," in *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[46] M. Jordan, J. Lewis, and A. Dimakis, "Provable certificates for adversarial examples: Fitting a ball in the union of polytopes," in *NeurIPS*, 2019.

[47] A. Fromherz, K. Leino, M. Fredrikson, B. Parno, and C. Păsăreanu, "Fast geometric projections for local robustness certification," in *International Conference on Learning Representations (ICLR)*, 2021.

[48] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[49] D. Pan, X. Li, and D. Zhu, "Explaining deep neural network models with adversarial gradient integration," in *Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[50] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International Conference on Machine Learning*, pp. 3145–3153, PMLR, 2017.

[51] B. Kim, M. Wattenberg, J. Gilmer, C. J. Cai, J. Wexler, F. Viégas, and R. Sayres, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)," in *ICML*, 2018.

[52] S. Ghalebikesabi, L. Ter-Minassian, K. Diaz-Ordaz, and C. C. Holmes, "On locality of local explanation models," *arxiv*, vol. abs/2106.14648, 2021.

[53] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st international conference on neural information processing systems*, pp. 4768–4777, 2017.

[54] C.-H. Chang, E. Creager, A. Goldenberg, and D. Duvenaud, "Explaining image classifiers by counterfactual generation," in *ICLR*, 2019.

[55] A. Dhurandhar, P. Chen, R. Luss, C.-C. Tu, P.-S. Ting, K. Shanmugam, and P. Das, "Explanations based on the missing: Towards contrastive explanations with pertinent negatives," in *NeurIPS*, 2018.

[56] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee, "Counterfactual visual explanations," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 2376–2384, 2019.

[57] J. Chen, X. Wu, V. Rastogi, Y. Liang, and S. Jha, "Robust attribution regularization," in *Advances in Neural Information Processing Systems 32*, 2019.

[58] A.-K. Dombrowski, M. Alber, C. Anders, M. Ackermann, K.-R. Müller, and P. Kessel, "Explanations can be manipulated and geometry is to blame," in *Advances in Neural Information Processing Systems 32*, 2019.

[59] S. Singla, E. Wallace, S. Feng, and S. Feizi, "Understanding impacts of high-order loss approximations and features in deep learning interpretation," in *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research.

[60] S.-M. Moosavi-Dezfooli, A. Fawzi, J. Uesato, and P. Frossard, "Robustness via curvature regularization, and vice versa," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[61] Y. Yoshida and T. Miyato, "Spectral norm regularization for improving the generalizability of deep learning," 2017.

[62] P. Chalasani, J. Chen, A. R. Chowdhury, X. Wu, and S. Jha, "Concise explanations of neural networks using adversarial training," in *International Conference on Machine Learning*, pp. 1383–1391, PMLR, 2020.

[63] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the gdpr," *Harvard Journal of Law & Technology*, vol. 31, no. 2, pp. 841–887, 2018.

[64] A. Van Looveren and J. Klaise, "Interpretable counterfactual explanations guided by prototypes," *arXiv preprint arXiv:1907.02584*, 2019.

[65] D. Mahajan, C. Tan, and A. Sharma, "Preserving causal constraints in counterfactual explanations for machine learning classifiers," *arXiv preprint arXiv:1912.03277*, 2019.

[66] S. Verma, J. Dickerson, and K. Hines, "Counterfactual explanations for machine learning: A review," *arXiv preprint arXiv:2010.10596*, 2020.

[67] M. Pawelczyk, K. Broelemann, and G. Kasneci, "Learning model-agnostic counterfactual explanations for tabular data," in *Proceedings of The Web Conference 2020*, pp. 3126–3132, 2020.

[68] M. Pawelczyk, K. Broelemann, and G. Kasneci, "On counterfactual explanations under predictive multiplicity," in *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, Proceedings of Machine Learning Research, 2020.

[69] K. Sokol and P. A. Flach, "Counterfactual explanations of machine learning predictions: opportunities and challenges for ai safety," in *SafeAI at AAAI*, 2019.

[70] M. T. Keane and B. Smyth, "Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable ai (xai)," in *International Conference on Case-Based Reasoning*, pp. 163–178, Springer, 2020.

[71] S. Dandl, C. Molnar, M. Binder, and B. Bischl, "Multi-objective counterfactual explanations," in *International Conference on Parallel Problem Solving from Nature*, pp. 448–469, Springer, 2020.

[72] L. Yang, E. M. Kenny, T. L. J. Ng, Y. Yang, B. Smyth, and R. Dong, "Generating plausible counterfactual explanations for deep transformers in financial text classification," *arXiv preprint arXiv:2010.12512*, 2020.

[73] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, and P. Das, "Explanations based on the missing: Towards contrastive explanations with pertinent negatives," *arXiv preprint arXiv:1802.07623*, 2018.

[74] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti, "Local rule-based explanations of black box decision systems," *arXiv preprint arXiv:1805.10820*, 2018.

[75] M. E. Kaminski, "The right to explanation, explained," *Berkeley Tech. LJ*, vol. 34, p. 189, 2019.

[76] GDPR, "European parliament and council of european union (2016) regulation (eu) 2016/679.." `https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN`, 2016.

[77] K. Rawal, E. Kamar, and H. Lakkaraju, "Can i still trust you?: Understanding the impact of distribution shifts on algorithmic recourses," 2021.

[78] K. Lu, Z. Wang, P. Mardziel, and A. Datta, "Influence patterns for explaining information flow in bert," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[79] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[80] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "Gnnexplainer: Generating explanations for graph neural networks," *Advances in neural information processing systems*, vol. 32, p. 9240, 2019.

[81] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang, "Parameterized explainer for graph neural network," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[82] E. Alsentzer, S. G. Finlayson, M. M. Li, and M. Zitnik, "Subgraph neural networks," 2021.

[83] C.-K. Yeh, C.-Y. Hsieh, A. Suggala, D. I. Inouye, and P. K. Ravikumar, "On the (in)fidelity and sensitivity of explanations," in *Advances in Neural Information Processing Systems*, 2019.

[84] B. Sanchez-Lengeling, J. Wei, B. Lee, E. Reif, P. Wang, W. Qian, K. McCloskey, L. Colwell, and A. Wiltschko, "Evaluating attribution for graph neural networks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[85] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *International conference on critical information infrastructures security*, pp. 88–99, Springer, 2016.

[86] N. Frosst and G. Hinton, "Distilling a neural network into a soft decision tree."

[87] P.-W. Wang, P. Donti, B. Wilder, and Z. Kolter, "Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver," in *International Conference on Machine Learning*, pp. 6545–6554, PMLR, 2019.

[88] M. V. Pogančić, A. Paulus, V. Musil, G. Martius, and M. Rolinek, "Differentiation of blackbox combinatorial solvers," in *International Conference on Learning Representations*, 2020.