Concise Explanations of Neural Networks using Adversarial Training

Prasad Chalasani [1] **Jiefeng Chen** [2] **Amrita Roy Chowdhury** [2] **Somesh Jha** [1 2] **Xi Wu** [3]

Abstract

We show new connections between adversarial learning and explainability for deep neural networks (DNNs). One form of explanation of the output of a neural network model in terms of its input features, is a vector of feature-attributions. Two desirable characteristics of an attribution-based explanation are: (1) *sparseness*: the attributions of irrelevant or weakly relevant features should be negligible, thus resulting in *concise* explanations in terms of the significant features, and (2) *stability*: it should not vary significantly within a small local neighborhood of the input. Our first contribution is a theoretical exploration of how these two properties (when using attributions based on Integrated Gradients, or IG) are related to adversarial training, for a class of 1-layer networks (which includes logistic regression models for binary and multi-class classification); for these networks we show that (a) adversarial training using an ℓ_∞-bounded adversary produces models with sparse attribution vectors, and (b) natural model-training while encouraging stable explanations (via an extra term in the loss function), is equivalent to adversarial training. Our second contribution is an empirical verification of phenomenon (a), which we show, somewhat surprisingly, occurs *not only in 1-layer networks, but also DNNs trained on standard image datasets*, and extends beyond IG-based attributions, to those based on DeepSHAP: adversarial training with ℓ_∞-bounded perturbations yields significantly sparser attribution vectors, with little degradation in performance on natural test data, compared to natural training. Moreover, the sparseness of the attribution vectors is significantly better than that achievable via ℓ_1-regularized natural training.

[1]XaiPient [2]University of Wisconsin (Madison) [3]Google. Correspondence to: Prasad Chalasani <pchalasani@gmail.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

1. Introduction

Despite the recent dramatic success of deep learning models in a variety of domains, two serious concerns have surfaced about these models.

Vulnerability to Adversarial Attacks: We can abstractly think of a neural network model as a function $F(\boldsymbol{x})$ of a d-dimensional input vector $\boldsymbol{x} \in \mathbb{R}^d$, and the range of F is either a discrete set of class-labels, or a continuous set of class probabilities. Many of these models can be foiled by an adversary who imperceptibly (to humans) alters the input $\boldsymbol{x}$ by adding a perturbation $\delta \in \mathbb{R}^d$ so that $F(\boldsymbol{x}+\delta)$ is very different from $F(\boldsymbol{x})$ (Szegedy et al., 2013; Goodfellow et al., 2014; Papernot et al., 2015; Biggio et al., 2013). *Adversarial training* (or *adversarial learning*) has recently been proposed as a method for training models that are robust to such attacks, by applying techniques from the area of Robust Optimization (Madry et al., 2017; Sinha et al., 2018). The core idea of adversarial training is simple: we define a set S of allowed perturbations $\delta \in \mathbb{R}^d$ that we want to "robustify" against (e.g. S could be the set of δ where $||\delta||_\infty \leq \varepsilon$), and perform model-training using Stochastic Gradient Descent (SGD) exactly as in natural training, except that each training example x is perturbed adversarially, i.e. replaced by $x + \delta^*$ where $\delta^* \in S$ maximizes the example's loss-contribution.

Explainability: One way to address the well-known lack of explainability of deep learning models is *feature attribution*, which aims to explain the output of a model $F(\boldsymbol{x})$ as an attribution vector $A^F(\boldsymbol{x})$ of the contributions from the *features* $\boldsymbol{x}$. There are several feature-attribution techniques in the literature, such as *Integrated Gradients* (IG) (Sundararajan et al., 2017), *DeepSHAP* (Lundberg & Lee, 2017), and *LIME* (Ribeiro et al., 2016). For such an explanation to be human-friendly, it is highly desirable (Molnar, 2019) that the attribution-vector is *sparse*, i.e., only the features that are truly predictive of the output $F(\boldsymbol{x})$ should have significant contributions, and irrelevant or weakly-relevant features should have negligible contributions. A sparse attribution makes it possible to produce a *concise* explanation, where only the input features with significant contributions are included. For instance, if the model F is used for a loan approval decision, then various stakeholders (like customers,

data-scientists and regulators) would like to know the reason for a specific decision in simple terms. In practice however, due to artifacts in the training data or process, the attribution vector is often not sparse and irrelevant or weakly-relevant features end up having significant contributions (Tan et al., 2013). Another desirable property of a good explanation is *stability*: the attribution vector should not vary significantly within a small local neighborhood of the input x. Similar to the lack of concise explainability, natural training often results in explanations that lack stability (Alvarez-Melis & Jaakkola, 2018).

Our paper shows new connections between adversarial robustness and the above-mentioned desirable properties of explanations, namely conciseness and stability. Specifically, let $\tilde{F}$ be an adversarially trained version of a classifier F, and for a given input vector $\boldsymbol{x}$ and attribution method A, let $A^F(\boldsymbol{x})$ and $A^{\tilde{F}}(\boldsymbol{x})$ denote the corresponding attribution vectors. The central research question this paper addresses is:

$$\text{Is } A^{\tilde{F}}(\boldsymbol{x}) \text{ sparser and more stable than } A^F(\boldsymbol{x})?$$

The main contributions of our paper are as follows:

Theoretical Analysis of Adversarial Training: Our first set of results show via a *theoretical* analysis that $\ell_\infty(\varepsilon)$-adversarial training 1-layer networks tends to produce sparse attribution vectors for IG, which in turn leads to concise explanations. In particular, under some assumptions, we show (Theorems 3.1 and E.1) that for a general class of convex loss functions (which includes popular loss functions used in 1-layer networks, such as logistic and hinge loss, used for binary or multi-class classification), and adversarial perturbations δ satisfying $||\delta||_\infty \leq \varepsilon$, the weights of "weak" features are on average more aggressively shrunk toward zero than during natural training, and the rate of shrinkage is proportional to the amount by which ε exceeds a certain measure of the "strength" of the feature. This shows that $\ell_\infty(\varepsilon)$-adversarial training tends to produce sparse *weight vectors* in popular 1-layer models. In Section 4 we show (Lemma 4.1) a closed form formula for the IG vector of 1-layer models, that makes it clear that in these models, sparseness of the *weight vector* directly implies sparseness of the *IG vector*.

Empirically Demonstrate Attribution Sparseness: In Section 6 we *empirically* demonstrate that this "sparsification" effect of $\ell_\infty(\varepsilon)$-adversarial training holds not only for 1-layer networks (e.g. logistic regression models), but also for Deep Convolutional Networks used for image classification, and extends beyond IG-based attributions, to those based on DeepSHAP. Specifically, we show this phenomenon via experiments applying $\ell_\infty(\varepsilon)$-adversarial training to (a) Convolutional Neural Networks on public benchmark image datasets MNIST (LeCun & Cortes, 2010) and Fashion-MNIST (Xiao et al., 2017), and (b) logistic regression models on the Mushroom and Spambase tabular datasets from the UCI Data Repository (Dheeru & Karra Taniskidou, 2017). In all of our experiments, we find that it is possible to choose an ℓ_∞ bound ε so that adversarial learning under this bound produces attribution vectors that are sparse on average, *with little or no drop in performance on natural test data*. A visually striking example of this effect is shown in Figure 1 (the Gini Index, introduced in Section 6, measures the sparseness of the map).

It is natural to wonder whether a traditional *weight-regularization* technique such as ℓ_1-regularization can produce models with sparse attribution vectors. In fact, our experiments show that for logistic regression models, ℓ_1-regularized training does yield attribution vectors that are on average significantly sparser compared to attribution vectors from natural (un-regularized) model-training, and the sparseness improvement is almost as good as that obtained with $\ell_\infty(\varepsilon)$-adversarial training. This is not too surprising given our result (Lemma 4.1) that implies a direct link between sparseness of *weights* and sparseness of *IG vectors*, *for 1-layer models*. Intriguingly, this does *not* carry over to DNNs: for multi-layer models (such as the ConvNets we trained for the image datasets mentioned above) we find that with ℓ_1-regularization, the sparseness improvement is significantly inferior to that obtainable from $\ell_\infty(\varepsilon)$-adversarial training (when controlling for model accuracy on natural test data), as we show in Table 1, Figure 2 and Figure 3. Thus it appears that for DNNs, the *attribution-sparseness* that results from adversarial training is not necessarily related to *sparseness of weights*.

Connection between Adversarial Training and Attribution Stability: We also show theoretically (Section 5) that training 1-layer networks naturally, while encouraging stability of explanations (via a suitable term added to the loss function), is in fact equivalent to adversarial training.

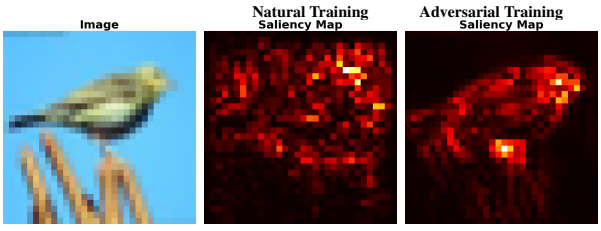

Figure 1: Both models correctly predict "Bird", but the IG-based saliency map of the adversarially trained model is much sparser than that of the naturally trained model.

2. Setup and Assumptions

For ease of understanding, we consider the case of binary classification for the rest of our discussion in the main paper. We assume there is a distribution $\mathcal{D}$ of data points $(\boldsymbol{x}, y)$ where $\boldsymbol{x} \in \mathbb{R}^d$ is an input feature vector, and $y \in \{\pm 1\}$ is its true label[1]. For each $i \in [d]$, the i'th component of $\boldsymbol{x}$ represents an input feature, and is denoted by x_i. The model is assumed to have learnable parameters ("weights") $\boldsymbol{w} \in \mathbb{R}^d$, and for a given data point $(\boldsymbol{x}, y)$, the *loss* is given by some function $\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w})$. *Natural model training* [2] consists of minimizing the expected loss, known as *empirical risk*:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w})]. \tag{1}$$

We sometimes assume the existence of an $\ell_\infty(\varepsilon)$-*adversary* who may perturb the input example $\boldsymbol{x}$ by adding a vector $\delta \in \mathbb{R}^d$ whose ℓ_∞-norm is bounded by ε; such a perturbation δ is referred to as an $\ell_\infty(\varepsilon)$-perturbation. For a given data point $(\boldsymbol{x}, y)$ and a given loss function $\mathcal{L}(.)$, an $\ell_\infty(\varepsilon)$-*adversarial* perturbation is a δ^* that maximizes the *adversarial loss* $\mathcal{L}(\boldsymbol{x}+\delta^*, y; \boldsymbol{w})$.

Given a function $F : \mathbb{R}^d \to [0, 1]$ representing a neural network, an input vector $\boldsymbol{x} \in \mathbb{R}^d$, and a suitable baseline vector $\boldsymbol{u} \in \mathbb{R}^d$, an *attribution* of the prediction of F at input $\boldsymbol{x}$ relative to $\boldsymbol{u}$ is a vector $A^F(\boldsymbol{x}, \boldsymbol{u}) \in \mathbb{R}^d$ whose i'th component $A^F_i(\boldsymbol{x}, \boldsymbol{u})$ represents the "contribution" of x_i to the prediction $F(\boldsymbol{x})$. A variety of *attribution methods* have been proposed in the literature (see (Arya et al., 2019) for a survey), but in this paper we will focus on two of the most popular ones: Integrated Gradients (Sundararajan et al., 2017), and DeepSHAP (Lundberg & Lee, 2017). When discussing a specific attribution method, we will denote the IG-based attribution vector as $\text{IG}^F(\boldsymbol{x}, \boldsymbol{u})$, and the DeepSHAP-based attribution vector as $\text{SH}^F(\boldsymbol{x}, \boldsymbol{u})$. In all cases we will drop the superscript F and/or the baseline vector $\boldsymbol{u}$ when those are clear from the context.

The aim of *adversarial training*(Madry et al., 2017) is to train a model that is *robust* to an $\ell_\infty(\varepsilon)$-adversary (i.e. performs well in the presence of such an adversary), and consists of minimizing the expected $\ell_\infty(\varepsilon)$-adversarial loss:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\max_{||\delta||_\infty \le \varepsilon} \mathcal{L}(\boldsymbol{x}+\delta, y; \boldsymbol{w})]. \tag{2}$$

In the expectations (1) and (2) we often drop the subscript under $\mathbb{E}$ when it is clear that the expectation is over $(\boldsymbol{x}, y) \sim \mathcal{D}$.

Some of our theoretical results make assumptions regarding the form and properties of the loss function $\mathcal{L}$, the properties of its first derivative. For the sake of clarify, we highlight these assumptions (with mnemonic names) here for ease of future reference.

[1] It is trivial to convert -1/1 labels to 0/1 labels and vice versa

[2] Also referred to as *standard training* by (Madry et al., 2017)

Assumption LOSS-INC. *The loss function is of the form* $\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w}) = g(-y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)$ *where* g *is a non-decreasing function.*

Assumption LOSS-CVX. *The loss function is of the form* $\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w}) = g(-y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)$ *where* g *is non-decreasing, almost-everywhere differentiable, and convex.*

Section B.1 in the Supplement shows that these Assumptions are satisfied by popular loss functions such as logistic and hinge loss. Incidentally, note that for any differentiable function g, g is convex if and only if its first-derivative g' is non-decreasing, and we will use this property in some of the proofs.

Assumption FEAT-TRANS. *For each* $i \in [d]$, *if* x'_i *is the feature in the original dataset,* x_i *is the* ***translated*** *version of* x'_i *defined by* $x_i = x'_i - [E(x'_i|y = 1) + E(x'_i|y = -1)]/2$.

In Section B.2 (Supplement) we show that this mild assumption implies that for each feature x_i there is a constant a_i such that

$$\mathbb{E}(x_i|y) = a_i y \tag{3}$$

$$\mathbb{E}(yx_i) = \mathbb{E}[\mathbb{E}(yx_i|y)] = \mathbb{E}[y^2 a_i] = a_i \tag{4}$$

$$\mathbb{E}(yx_i|y) = y\mathbb{E}[x_i|y] = y^2 a_i = a_i \tag{5}$$

For any $i \in [d]$, we can think of $\mathbb{E}(yx_i)$ as the *degree of association*[3] between feature x_i and label y. Since $\mathbb{E}(yx_i) = a_i$ (Eq. 4), we refer to a_i as the *directed strength*[4] of feature x_i, and $|a_i|$ is referred to as the *absolute strength* of x_i. In particular when $|a_i|$ is large (small) we say that x_i is a *strong* (*weak*) feature.

2.1. Averaging over a group of features

For our main theoretical result (Theorem 3.1), we need a notion of weighted average defined as follows:

Definition WTD-AV. *Given a quantity* q_i *defined for each feature-index* $i \in [d]$, *a subset* $S \subset [d]$ *(where* $|S| \ge 1$*) of feature-indices, and a feature weight-vector* $\mathbf{w}$ *with* $w_i \neq 0$ *for at least one* $i \in S$, *the* ***w-weighted average of*** q ***over*** S *is defined as*

$$q_S^{\mathbf{w}} := \frac{\sum_{i\in S} w_i q_i}{\sum_{i\in S} |w_i|} \tag{6}$$

Note that the quantity $w_i q_i$ can be written as $|w_i| \operatorname{sgn}(w_i) q_i$, so $q_S^{\mathbf{w}}$ is essentially a $|w_i|$-weighted average of $\operatorname{sgn}(w_i) q_i$ over $i \in S$.

For our result we will use the above $\mathbf{w}$-weighted average definition for two particular quantities q_i. The first one is

[3] When the features are standardized to have mean 0, $\mathbb{E}(yx_i)$ is in fact the covariance of y and x_i.

[4] This is related to the feature "robustness" notion introduced in (Ilyas et al., 2019)

$q_i := a_i$, the directed strength of feature x_i (Eq. 4). Intuitively, the quantity $\text{sgn}(w_i)\mathbb{E}(yx_i) = a_i \,\text{sgn}(w_i)$ captures the *aligned strength* of feature x_i in the following sense: if this quantity is large and positive (large and negative), it indicates both that the current weight w_i of x_i is aligned (misaligned) with the directed strength of x_i, *and* that this directed strength is large. Thus $a_S^{\mathbf{w}}$ represents an average of the aligned strength over the feature-group S.

The second quantity for which we define the above $\mathbf{w}$-weighted average is $q_i := \overline{\Delta}_i$, where $\overline{\Delta}_i := -\mathbb{E}[\partial\mathcal{L}/\partial w_i]$ represents the *expected* SGD update (over random draws from the data distribution) of the weight w_i, given the loss function $\mathcal{L}$, for a unit learning rate (details are in the next Section). The quantity $\text{sgn}(w_i)\overline{\Delta}_i$ has a natural interpretation, analogous to the above interpretation of $a_i \,\text{sgn}(w_i)$: a large positive (large negative) value of $\text{sgn}(w_i)\overline{\Delta}_i$ corresponds to an *large expansion* (*large shrinkage*), in expectation, of the weight w_i away from zero magnitude (toward zero magnitude). Thus the $\mathbf{w}$-weighted average $\overline{\Delta}_S^{\mathbf{w}}$ represents the $|w_i|$-weighted average of this effect over the feature-group S.

3. Analysis of SGD Updates in Adversarial Training

One way to understand the characteristics of the weights in an adversarially-trained neural network model, is to analyze how the weights evolve during adversarial training under Stochastic Gradient Descent (SGD) optimization. One of the main results of this work is a theoretical characterization of the weight updates during a single SGD step, when applied to a randomly drawn data point $(\boldsymbol{x}, y) \sim \mathcal{D}$ that is subjected to an $\ell_\infty(\varepsilon)$-adversarial perturbation.

Although the holy grail would be to do this for general DNNs (and we expect this will be quite difficult) we take a first step in this direction by analyzing *single-layer networks* for binary or multi-class classification, where each weight is associated with an input feature. Intriguingly, our results (Theorem 3.1 for binary classification and E.1 for multi-class classification in the Supplement) show that for these models, $\ell_\infty(\varepsilon)$-adversarial training tends to selectively reduce the weight-magnitude of *weakly relevant* or *irrelevant* features, and does so much more aggressively than natural training. In other words, natural training can result in models where many weak features have significant weights, whereas adversarial training would tend to push most of these weights close to zero. The resulting model weights would thus be more sparse, and the corresponding IG-based attribution vectors would on average be more sparse as well (since in linear models, sparse weights imply sparse IG vectors; this is a consequence of Lemma 4.1) compared to naturally-trained models.

Our experiments (Sec. 6) show that indeed for logistic regression models (which satisfy the conditions of Theorem 3.1), adversarial training leads to sparse IG vectors. Interestingly, our extensive experiments with Deep Convolutional Neural Networks on public image datasets demonstrate that this phenomenon extends to DNNs as well, and to attributions based on DeepSHAP, even though our theoretical results only apply to 1-layer networks and IG-based attributions.

As a preliminary, it is easy to show the following expressions related to the $\ell_\infty(\varepsilon)$-adversarial perturbation δ^* (See Lemmas 2 and 3 in Section C of the Supplement): For loss functions satisfying Assumption LOSS-INC, the $\ell_\infty(\varepsilon)$-adversarial perturbation δ^* is given by:

$$\delta^* = -y \,\text{sgn}(\boldsymbol{w})\varepsilon, \tag{7}$$

the corresponding $\ell_\infty(\varepsilon)$-adversarial loss is

$$\mathcal{L}(\boldsymbol{x} + \delta^*, y;\ \boldsymbol{w}) = g(\varepsilon||\,\boldsymbol{w}\,||_1 - y\langle \boldsymbol{w}, \boldsymbol{x}\rangle), \tag{8}$$

and the gradient of this loss w.r.t. a weight w_i is

$$\frac{\partial\mathcal{L}(\boldsymbol{x} + \delta^*, y;\ \boldsymbol{w})}{\partial w_i} = \\ -\,g'(\varepsilon||\,\boldsymbol{w}\,||_1 - y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)\ (yx_i - \text{sgn}(w_i)\varepsilon). \tag{9}$$

In our main result, the expectation of the g' term in (9) plays an important role, so we will use the following notation:

$$\overline{g'} := \mathbb{E}\big[g'(\varepsilon||\,\boldsymbol{w}\,||_1 - y\langle \boldsymbol{w}, \boldsymbol{x}\rangle)\big], \tag{10}$$

and by Assumption LOSS-INC, $\overline{g'}$ **is non-negative**.

Ideally, we would like to understand the nature of the weight-vector $\boldsymbol{w}^*$ that minimizes the expected adversarial loss (2). This is quite challenging, so rather than analyzing the *final* optimum of (2), we instead analyze how an SGD-based optimizer for (2) *updates* the model weights $\boldsymbol{w}$. We assume an idealized SGD process: (a) a data point $(\boldsymbol{x}, y)$ is drawn from distribution $\mathcal{D}$, (b) $\boldsymbol{x}$ is replaced by $\boldsymbol{x}' = \boldsymbol{x} + \delta^*$ where δ^* is an $\ell_\infty(\varepsilon)$-adversarial perturbation with respect to the loss function $\mathcal{L}$, (c) each weight w_i is updated by an amount $\Delta w_i = -\partial\mathcal{L}(\boldsymbol{x}', y; \boldsymbol{w})/\partial w_i$ (assuming a unit learning rate to avoid notational clutter). We are interested in the *expectation* $\overline{\Delta}_i := \mathbb{E}\Delta w_i = -\mathbb{E}[\partial\mathcal{L}(\boldsymbol{x}', y; \mathbf{w})/\partial w_i]$, in order to understand how a weight w_i evolves *on average* during a single SGD step. Where there is a *conditionally independent* feature subset $S \in [d]$ (i.e. the features in S are conditionally independent of the rest given the label y), our main theoretical result characterizes the behavior of $\overline{\Delta}_i$ for $i \in S$, and the corresponding $\mathbf{w}$-weighted average $\overline{\Delta}_S^{\mathbf{w}}$:

Theorem 3.1 (Expected SGD Update in Adversarial Training)**.** *For any loss function $\mathcal{L}$ satisfying Assumption LOSS-CVX, a dataset $\mathcal{D}$ satisfying Assumption FEAT-TRANS, a*

subset S of features that are conditionally independent of the rest given the label y, if a data point $(\boldsymbol{x}, y)$ is randomly drawn from $\mathcal{D}$, and $\boldsymbol{x}$ is perturbed to $\boldsymbol{x}' = \boldsymbol{x} + \delta^*$, where δ^* is an $\ell_\infty(\varepsilon)$-adversarial perturbation, then during SGD using the $\ell_\infty(\varepsilon)$-adversarial loss $\mathcal{L}(\boldsymbol{x}', y; \boldsymbol{w})$, the expected weight-updates $\overline{\Delta}_i := \mathbb{E}\Delta w_i$ for $i \in S$ and the corresponding $\mathbf{w}$-weighted average $\overline{\Delta}_S^{\mathbf{w}}$ satisfy the following properties:

1. *If $w_i = 0\ \forall i \in S$, then for each $i \in S$,*

$$\overline{\Delta}_i = \overline{g'}\ a_i, \tag{11}$$

2. *and otherwise,*

$$\overline{\Delta}_S^{\mathbf{w}} \leq \overline{g'}(a_S^{\mathbf{w}} - \varepsilon), \tag{12}$$

and equality holds in the limit as $w_i \to 0\ \forall i \in S$,

where $\overline{g'}$ is the expectation in (10), $a_i = \mathbb{E}(x_i y)$ is the directed strength of feature x_i from Eq. (4), and $a_S^{\mathbf{w}}$ is the corresponding $\mathbf{w}$-weighted average over S.

For space reasons, a detailed discussion of the implications of this result is presented in Sec. D.2 of the Supplement, but here we note the following. Recalling the interpretation of the $\mathbf{w}$-weighted averages $a_S^{\mathbf{w}}$ and $\overline{\Delta}_S^{\mathbf{w}}$ in Section 2.1, we can interpret the above result as follows. For any conditionally independent feature subset S, if the weights of all features in S are zero, then by Eq. (11), an SGD update causes, on average, each of these weights w_i to grow (from 0) in a direction consistent with the directed feature-strength a_i (since $\overline{g'} \geq 0$ as noted above). If at least one of the features in S has a non-zero weight, (12) implies $\overline{\Delta}_S^{\mathbf{w}} < 0$, i.e., an aggregate shrinkage of the weights of features in S, if either of the following hold: (a) $a_S^{\mathbf{w}} < 0$, i.e., the weights of features in S are mis-aligned on average, or (b) the weights of features in S are aligned on average, i.e., $a_S^{\mathbf{w}}$ is positive, but dominated by ε, i.e. the features S are *weakly correlated* with the label. In the latter case the weights of features in S are (in aggregate and in expectation) aggressively pushed toward zero, and this aggressiveness is proportional to the extent to which ε dominates $a_S^{\mathbf{w}}$. A partial generalization of the above result for the multi-class setting (for a single conditionally-independent feature) is presented in Section E (Theorem E.1) of the Supplement.

4. FEATURE ATTRIBUTION USING INTEGRATED GRADIENTS

Theorem 3.1 showed that $\ell_\infty(\varepsilon)$-adversarial training tends to shrink the *weights* of features that are "weak" (relative to ε). We now show a link between weights and *explanations*, specifically explanations in the form of a vector of feature-attributions given by the *Integrated Gradients* (IG) method (Sundararajan et al., 2017), which is defined as follows: Suppose $F : \mathbb{R}^d \to \mathbb{R}$ is a real-valued function of an input vector. For example F could represent the output of a neural network, or even a loss function $\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w})$ when the label y and weights $\boldsymbol{w}$ are held fixed. Let $\boldsymbol{x} \in \mathbb{R}^d$ be a specific input, and $\boldsymbol{u} \in \mathbb{R}^d$ be a baseline input. The IG is defined as the path integral of the gradients along the straight-line path from the baseline $\boldsymbol{u}$ to the input $\boldsymbol{x}$. The IG along the i'th dimension for an input $\boldsymbol{x}$ and baseline $\boldsymbol{u}$ is defined as:

$$\text{IG}_i^F(\boldsymbol{x}, \boldsymbol{u}) := (x_i - u_i) \times \int_{\alpha=0}^{1} \partial_i F(\boldsymbol{u} + \alpha(\boldsymbol{x} - \boldsymbol{u}))d\alpha, \tag{13}$$

where $\partial_i F(\boldsymbol{z})$ denotes the gradient of $F(\boldsymbol{v})$ along the i'th dimension, at $\boldsymbol{v} = \boldsymbol{z}$. The vector of all IG components $\text{IG}_i^F(\boldsymbol{x}, \boldsymbol{u})$ is denoted as $\text{IG}^F(\boldsymbol{x}, \boldsymbol{u})$. Although we do not show $\boldsymbol{w}$ explicitly as an argument in the notation $\text{IG}^F(\boldsymbol{x}, \boldsymbol{u})$, it should be understood that the IG depends on the model weights $\boldsymbol{w}$ since the function F depends on $\boldsymbol{w}$.

The following Lemma (proved in Sec. F of the Supplement) shows a closed form exact expression for the $\text{IG}^F(\boldsymbol{x}, \boldsymbol{u})$ when $F(\boldsymbol{x})$ is of the form

$$F(\boldsymbol{x}) = A(\langle \boldsymbol{w}, \boldsymbol{x} \rangle), \tag{14}$$

where $\boldsymbol{w} \in \mathbb{R}^d$ is a vector of weights, A is a differentiable scalar-valued function, and $\langle \boldsymbol{w}, \boldsymbol{x} \rangle$ denotes the dot product of $\boldsymbol{w}$ and x. Note that this form of F could represent a single-layer neural network with any differentiable activation function (e.g., logistic (sigmoid) activation $A(\boldsymbol{z}) = 1/[1 + \exp(-\boldsymbol{z})]$ or Poisson activation $A(\boldsymbol{z}) = \exp(\boldsymbol{z})$), or a differentiable loss function, such as those that satisfy Assumption LOSS-INC for a fixed label y and weight-vector $\boldsymbol{w}$. For brevity, we will refer to a function of the form (14) as representing a "1-Layer Network", with the understanding that it could equally well represent a suitable loss function.

Lemma 4.1 (IG Attribution for 1-layer Networks)**.** *If $F(\boldsymbol{x})$ is computed by a 1-layer network (14) with weights vector $\boldsymbol{w}$, then the Integrated Gradients for all dimensions of $\boldsymbol{x}$ relative to a baseline $\boldsymbol{u}$ are given by:*

$$\text{IG}^F(\boldsymbol{x}, \boldsymbol{u}) = [F(\boldsymbol{x}) - F(\boldsymbol{u})] \frac{(\boldsymbol{x} - \boldsymbol{u}) \odot \boldsymbol{w}}{\langle \boldsymbol{x} - \boldsymbol{u}, \boldsymbol{w} \rangle}, \tag{15}$$

where the $\odot$ operator denotes the entry-wise product of vectors.

Thus for 1-layer networks, the IG of each feature is essentially proportional to the feature's fractional contribution to the logit-change $\langle \boldsymbol{x} - \boldsymbol{u}, \boldsymbol{w} \rangle$. This makes it clear that in such models, if the weight-vector $\boldsymbol{w}$ is sparse, then the IG vector will also be correspondingly sparse.

5. Training with Explanation Stability is equivalent to Adversarial Training

Suppose we use the IG method described in Sec. 4 as an explanation for the output of a model $F(\boldsymbol{x})$ on a specific input $\boldsymbol{x}$. A desirable property of an explainable model is that the explanation for the value of $F(\boldsymbol{x})$ is *stable*(Alvarez-Melis & Jaakkola, 2018), i.e., does not change much under small perturbations of the input $\boldsymbol{x}$. One way to formalize this is to say the following *worst-case* ℓ_1-norm of the change in IG should be small:

$$\max_{\boldsymbol{x}' \in N(\boldsymbol{x},\varepsilon)} ||\operatorname{IG}^F(\boldsymbol{x}', \boldsymbol{u}) - \operatorname{IG}^F(\boldsymbol{x}, \boldsymbol{u})||_1, \tag{16}$$

where $N(\boldsymbol{x}, \varepsilon)$ denotes a suitable ε-neighborhood of $\boldsymbol{x}$, and $\boldsymbol{u}$ is an appropriate baseline input vector. If the model F is a single-layer neural network, it would be a function of $\langle \boldsymbol{w}, \boldsymbol{x} \rangle$ for some weights $\boldsymbol{w}$, and typically when training such networks the loss is a function of $\langle \boldsymbol{w}, \boldsymbol{x} \rangle$ as well, so we would not change the essence of (16) much if instead of F in each IG, we use $\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w})$ for a fixed y; let us denote this function by $\mathcal{L}_y$. Also intuitively, $||\operatorname{IG}^{\mathcal{L}_y}(\boldsymbol{x}', \boldsymbol{u}) - \operatorname{IG}^{\mathcal{L}_y}(\boldsymbol{x}, \boldsymbol{u})||_1$ is not too different from $||\operatorname{IG}^{\mathcal{L}_y}(\boldsymbol{x}', \boldsymbol{x})||_1$. These observations motivate the following definition of *Stable-IG Empirical Risk*, which is a modification of the usual empirical risk (1), with a regularizer to encourage stable IG explanations:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\Big[\, \mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w}) + \max_{||\boldsymbol{x}' - \boldsymbol{x}||_\infty \leq \varepsilon} ||\operatorname{IG}^{\mathcal{L}_y}(\boldsymbol{x}, \boldsymbol{x}')||_1 \,\Big]. \tag{17}$$

The following somewhat surprising result is proved in Section G of the Supplement.

Theorem 5.1 (Equivalence of Stable IG and Adversarial Robustness)**.** *For loss functions $\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w})$ satisfying Assumption LOSS-CVX, the augmented loss inside the expectation (17) equals the $\ell_\infty(\varepsilon)$-adversarial loss inside the expectation (2), i.e.*

$$\mathcal{L}(\boldsymbol{x}, y; \boldsymbol{w}) + \max_{||\boldsymbol{x}' - \boldsymbol{x}||_\infty \leq \varepsilon} ||\operatorname{IG}^{\mathcal{L}_y}(\boldsymbol{x}, \boldsymbol{x}')||_1 = \max_{||\delta||_\infty \leq \varepsilon} \mathcal{L}(\boldsymbol{x} + \delta, y; \boldsymbol{w}) \tag{18}$$

This implies that for loss functions satisfying Assumption LOSS-CVX, minimizing the Stable-IG Empirical Risk (17) is equivalent to minimizing the expected $\ell_\infty(\varepsilon)$-adversarial loss. In other words, for this class of loss functions, *natural model training while encouraging IG stability is equivalent to $\ell_\infty(\varepsilon)$-adversarial training!* Combined with Theorem 3.1 and the corresponding experimental results in Sec 6, this equivalence implies that, for this class of loss functions, and data distributions satisfying Assumption FEAT-TRANS, the explanations for the models produced by $\ell_\infty(\varepsilon)$-adversarial training are both *concise* (due to the sparseness of the models), and *stable*.

6. Experiments

6.1. Hypotheses

Recall that one implication of Theorem 3.1 is the following: For 1-layer networks where the loss function satisfies Assumption LOSS-CVX, $\ell_\infty(\varepsilon)$-adversarial training tends to more-aggressively prune the *weight-magnitudes* of "weak" features compared to natural training. In Sec. 4 we observed that a consequence of Lemma 4.1 is that for 1-layer models the sparseness of the weight vector implies sparseness of the IG vector. Thus a reasonable conjecture is that, for 1-layer networks, $\ell_\infty(\varepsilon)$-adversarial training leads to models with sparse attribution vectors in general (whether using IG or a different method, such as DeepSHAP). We further conjecture that this sparsification phenomenon extends to practical multi-layer Deep Neural Networks, not just 1-layer networks, and that this benefit can be realized without significantly impacting accuracy on natural test data. Finally, we hypothesize that the resulting sparseness of *attribution vectors* is better than what can be achieved by a traditional *weight regularization* technique such as L1-regularization, for a comparable level of natural test accuracy.

6.2. Measuring Sparseness of an Attribution Vector

For an attribution method A, we quantify the sparseness of the attribution vector $A^F(\boldsymbol{x}, \boldsymbol{u})$ using the *Gini Index* applied to the vector of absolute values $A^F(\boldsymbol{x}, \boldsymbol{u})$. For a vector $\boldsymbol{v}$ of non-negative values, the Gini Index, denoted $G(\boldsymbol{v})$ (defined formally in Sec. I in the Supplement), is a metric for sparseness of $\boldsymbol{v}$ that is known (Hurley & Rickard, 2009) to satisfy a number of desirable properties, and has been used to quantify sparseness of weights in a neural network (Guest & Love, 2017). The Gini Index by definition lies in [0,1], and a higher value indicates more sparseness.

Since the model F is clear from the context, and the baseline vector $\boldsymbol{u}$ are fixed for a given dataset, we will denote the attribution vector on input $\boldsymbol{x}$ simply as $A(\boldsymbol{x})$, and our measure of sparseness is $G(|A(\boldsymbol{x})|)$, which we denote for brevity as $G[A](\boldsymbol{x})$, and refer to informally as the *Gini of* A, where A can stand for IG (when using IG-based attributions) or SH (when using DeepSHAP for attribution). As mentioned above, one of our hypotheses is that the sparseness of attributions of models produced by $\ell_\infty(\varepsilon)$-adversarial training is much better than what can be achieved by natural training using ℓ_1-regularization, for a comparable level of accuracy. To verify this hypothesis we will compare the sparseness of attribution vectors resulting from three types of models: (a) n-model: *naturally-trained* model with no adversarial perturbations and no ℓ_1-regularization, (b) a-model: *$\ell_\infty(\varepsilon)$-adversarially trained* model, and (c) l-model: naturally trained model with *ℓ_1-regularization* strength $\lambda > 0$. For an attribution method A, we denote the Gini indices $G[A](\boldsymbol{x})$ resulting from these models respec-

tively as $G^n[A](\boldsymbol{x})$, $G^a[A](\boldsymbol{x};\ \varepsilon)$ and $G^l[A](\boldsymbol{x};\ \lambda)$.

In several of our datasets, individual feature vectors are already quite sparse: for example in the MNIST dataset, most of the area consists of black pixels, and in the Mushroom dataset, after 1-hot encoding the 22 categorical features, the resulting 120-dimensional feature-vector is sparse. On such datasets, even an n-model can achieve a "good" level of sparseness of attributions in the absolute sense, i.e. $G^n[A](\boldsymbol{x})$ can be quite high. Therefore for all datasets we compare the *sparseness improvement* resulting from an a-model relative to an n-model, with that from an l-model relative to a n-model. Or more precisely, we will compare the two quantities defined below, for a given attribution method A:

$$dG^a[A](\boldsymbol{x};\ \varepsilon) := G^a[A](\boldsymbol{x};\ \varepsilon) - G^n[A](\boldsymbol{x}), \tag{19}$$

$$dG^l[A](\boldsymbol{x};\ \lambda) := G^l[A](\boldsymbol{x};\ \lambda) - G^n[A](\boldsymbol{x}). \tag{20}$$

The above quantities define the IG sparseness improvements for a *single* example $\boldsymbol{x}$. It will be convenient to define the overall sparseness improvement from a model, as measured on a test dataset, by averaging over all examples $\boldsymbol{x}$ in that dataset. We denote the corresponding *average* sparseness metrics by $G^a[A](\varepsilon)$, $G^l[A](\lambda)$ and $G^n[A]$ respectively. We then define the *average sparseness improvement* of an a-model and l-model as:

$$dG^a[A](\varepsilon) := G^a[A](\varepsilon) - G^n[A], \tag{21}$$

$$dG^l[A](\lambda) := G^l[A](\lambda) - G^n[A]. \tag{22}$$

We can thus re-state our hypotheses in terms of this notation: For each of the attribution methods $A \in \{IG, SH\}$, the average sparseness improvement $dG^a[A](\varepsilon)$ resulting from type-a models is high, and is significantly higher than the average sparseness improvement $dG^l[A](\lambda)$ resulting from type-l models.

6.3. Results

We ran experiments on five standard public benchmark datasets: three image datasets MNIST, Fashion-MNIST, and CIFAR-10, and two tabular datasets from the UCI Data Repository: Mushroom and Spambase. Details of the datasets and training methodology are in Sec. J.1 of the Supplement. The code for all experiments is at this repository: `https://github.com/jfc43/advex`.

For each of the two tabular datasets (where we train logistic regression models), for a given model-type (a, l or n), we found the average Gini index of the attribution vectors is virtually identical when using IG or DeepSHAP. This is not surprising: as pointed out in (Ancona et al., 2017), DeepSHAP is a variant of DeepLIFT, and for simple linear models, DeepLIFT gives a very close approximation of IG. To avoid clutter, we therefore omit DeepSHAP-based results on the tabular datasets. Table 1 shows a summary of some results on the above 5 datasets, and Fig. 2 and 3 display results graphically [5]

Table 1: Results on 5 datasets. For each dataset, "a" indicates an $\ell_\infty(\varepsilon)$-adversarially trained model with the indicated ε, and "l" indicates a naturally trained model with the indicated ℓ_1-regularization strength λ. The **attr** column indicates the feature attribution method (IG or DeepSHAP). Column **dG** shows the average sparseness improvements of the models relative to the baseline naturally trained model, as measured by the $dG^a[A](\varepsilon)$ and $dG^l[A](\lambda)$ defined in Eqs. (21, 22). Column **AcDrop** indicates the drop in accuracy relative to the baseline model.

dataset	attr	model	dG	AcDrop
MNIST	IG	a ($\varepsilon = 0.3$)	0.06	0.8%
	IG	l ($\lambda = 0.01$)	0.004	0.4%
	SHAP	a ($\varepsilon = 0.3$)	0.06	0.8%
	SHAP	l ($\lambda = 0.01$)	0.007	0.4%
Fashion -MNIST	IG	a ($\varepsilon = 0.1$)	0.06	4.7%
	IG	l ($\lambda = 0.01$)	0.008	3.4%
	SHAP	a ($\varepsilon = 0.1$)	0.08	4.7%
	SHAP	l ($\lambda = 0.01$)	0.003	3.4%
CIFAR-10	IG	a ($\varepsilon = 1.0$)	0.081	0.57%
	IG	l ($\lambda = 10^{-5}$)	0.022	1.51%
Mushroom	IG	a ($\varepsilon = 0.1$)	0.06	2.5%
	IG	l ($\lambda = 0.02$)	0.06	2.6%
Spambase	IG	a ($\varepsilon = 0.1$)	0.17	0.9%
	IG	l ($\lambda = 0.02$)	0.15	0.1%

These results make it clear that for comparable levels of accuracy, the sparseness of attribution vectors from $\ell_\infty(\varepsilon)$-adversarially trained models is much better than the sparseness from natural training with ℓ_1-regularization. The effect is especially pronounced in the two image datasets. The effect is less dramatic in the two tabular datasets, for which we train logistic regression models. Our discussion at the end of Sec. 4 suggests a possible explanation.

In the Introduction we gave an example of a *saliency map* (Simonyan et al., 2013; Baehrens et al., 2010) (Fig. 1) to dramatically highlight the sparseness induced by adversarial training. We show several more examples of saliency maps in the supplement (Section J.4).

[5] The official implementation of DeepSHAP (`https://github.com/slundberg/shap`) doesn't support the network we use for CIFAR-10 well, so we do not show results for this combination.

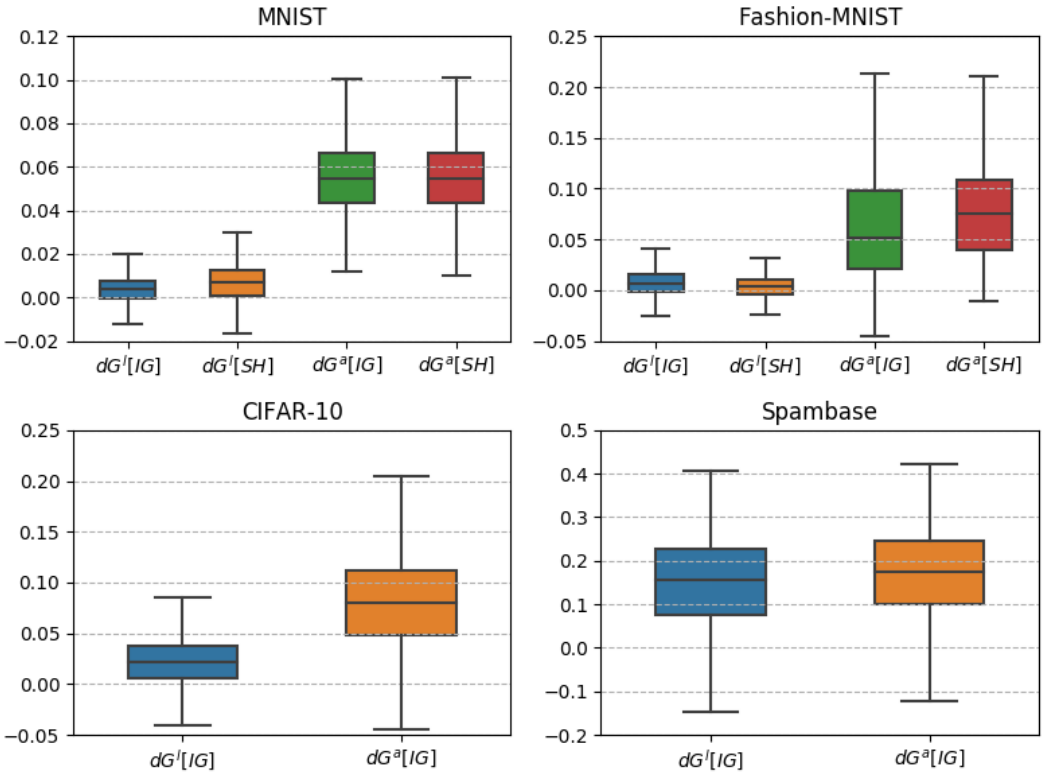

Figure 2: Boxplot of pointwise sparseness-improvements from adversarially trained models ($dG^a[A](\boldsymbol{x}, \varepsilon)$) and naturally trained models with ℓ_1-regularization ($dG^l[A](\boldsymbol{x}, \lambda)$), for attribution methods $A \in \{IG, SH\}$.

7. Related Work

In contrast to the growing body of work on defenses against adversarial attacks (Yuan et al., 2017; Madry et al., 2017; Biggio et al., 2013) or explaining adversarial examples (Goodfellow et al., 2014; Tsipras et al., 2018), the focus of our paper is the connection between adversarial robustness and explainability. We view the process of adversarial training as a tool to produce more explainable models. A recent series of papers (Tsipras et al., 2018; Ilyas et al., 2019) essentially argues that adversarial examples exist because standard training produces models are heavily reliant on highly predictive but *non-robust features* (which is similar to our notion of "weak" features in Sec 3) which are vulnerable to an adversary who can "flip" them and cause performance to degrade. Indeed the authors of (Ilyas et al., 2019) touch upon some connections between explainability and robustness, and conclude, "*As such, producing human-meaningful explanations that remain faithful to underlying models cannot be pursued independently from the training of the models themselves*", by which they are implying that good explainability may require intervening in the model-training procedure itself; this is consistent with our findings. We discuss other related work in the Supplement Section A.

8. Conclusion

We presented theoretical and experimental results that show a strong connection between adversarial robustness (under ℓ_∞-bounded perturbations) and two desirable properties of model explanations: conciseness and stability. Specifically, we considered model explanations in the form of feature-attributions based on the Integrated Gradients (IG) and DeepSHAP techniques. For 1-layer models using a popular family of loss functions, we theoretically showed that $\ell_\infty(\varepsilon)$-adversarial training tends to produce *sparse* and *stable* IG-based attribution vectors. With extensive experiments on benchmark tabular and image datasets, we demonstrated that the "attribution sparsification" effect extends to Deep Neural Networks, when using two popular attribution methods. Intriguingly, especially in DNN models for image classification, the attribution sparseness from natural training with ℓ_1-regularization is much inferior to that achievable via $\ell_\infty(\varepsilon)$-adversarial training. Our theoretical results are a first step in explaining some of these phenomena.

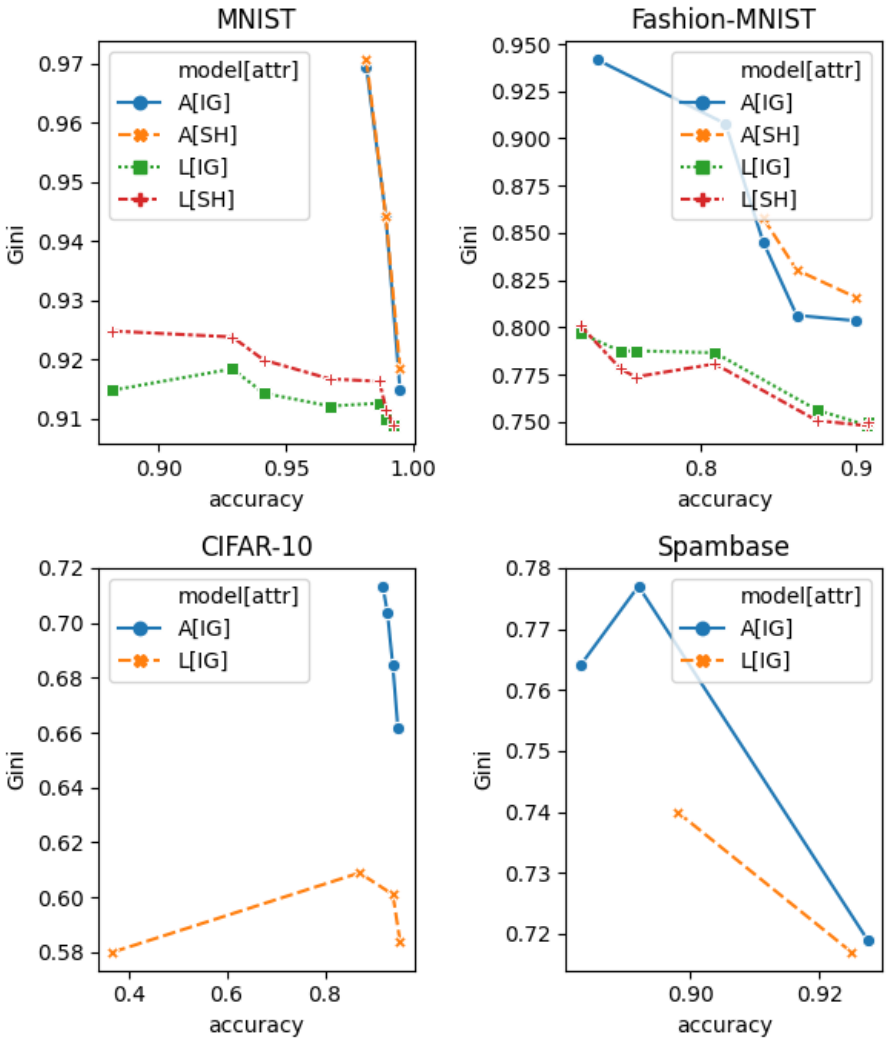

Figure 3: For four benchmark datasets, each line-plot labeled $M[A]$ shows the tradeoff between Accuracy and Attribution Sparseness achievable by various combinations of models M and attribution methods A. $M = A$ denotes $\ell_\infty(\varepsilon)$-adversarial training, and the plot shows the accuracy/sparseness for various choices of ε. $M = L$ denotes ℓ_1-regularized natural training, and the plot shows accuracy/sparseness for various choices of ℓ_1-regularization parameter λ. $A = IG$ denotes the IG attribution method, whereas $A = SH$ denotes DeepSHAP. Attribution sparseness is measured by the Average Gini Index over the dataset ($G^a[A](\varepsilon)$ and $G^l[A](\lambda)$, for adversarial training and ℓ_1-regularized natural training respectively). In all cases, and especially in the image datasets, adversarial training achieves a significantly better accuracy vs attribution sparseness tradeoff.