Do Input Gradients Highlight Discriminative Features?

Harshay Shah[*]
Microsoft Research India
harshay@google.com

Prateek Jain[*]
Microsoft Research India
prajain@google.com

Praneeth Netrapalli[*]
Microsoft Research India
pnetrapalli@google.com

Abstract

Post-hoc gradient-based interpretability methods [1, 2] that provide instance-specific explanations of model predictions are often based on assumption (`A`): *magnitude of input gradients—gradients of logits with respect to input—noisily highlight discriminative task-relevant features*. In this work, we test the validity of assumption (`A`) using a three-pronged approach:

1. We develop an evaluation framework, `DiffROAR`, to test assumption (`A`) on four image classification benchmarks. Our results suggest that (i) input gradients of standard models (i.e., trained on original data) may grossly violate (`A`), whereas (ii) input gradients of adversarially robust models satisfy (`A`) reasonably well.
2. We then introduce `BlockMNIST`, an `MNIST`-based semi-real dataset, that *by design* encodes a priori knowledge of discriminative features. Our analysis on `BlockMNIST` leverages this information to validate as well as characterize differences between input gradient attributions of standard and robust models.
3. Finally, we theoretically prove that our empirical findings hold on a simplified version of the `BlockMNIST` dataset. Specifically, we prove that input gradients of standard one-hidden-layer MLPs trained on this dataset do not highlight instance-specific "signal" coordinates, thus grossly violating (`A`).

Our findings motivate the need to formalize and test common assumptions in interpretability in a falsifiable manner [3]. We believe that the `DiffROAR` framework and `BlockMNIST` datasets serve as sanity checks to audit interpretability methods; code and data available at `https://github.com/harshays/inputgradients`.

1 Introduction

Interpretability methods that provide instance-specific explanations of model predictions are often used to identify biased predictions [4], debug trained models [5], and aid decision-making in high-stakes domains such as medical diagnosis [6, 7]. A common approach for providing instance-specific explanations is *feature attribution*. Feature attribution methods rank or score input coordinates, or features, in the order of their *purported* importance in model prediction; coordinates achieving the top-most rank or score are considered most important for prediction, whereas those with the bottom-most rank or score are considered least important.

Input gradient attributions. Ranking input coordinates based on the *magnitude of input gradients* is a fundamental feature attribution technique [8, 1] that undergirds well-known methods such as SmoothGrad [2] and Integrated Gradients [9]. Given instance x and a trained model θ with prediction $\hat{y}$ on x, the input gradient attribution scheme (i) computes the input gradient $\nabla_x \text{Logit}_\theta(x, \hat{y})$ of the logit [2] of the predicted label $\hat{y}$ and (ii) ranks the input coordinates in *decreasing* order of their input gradient magnitude. Below we explicitly characterize the underlying intuitive assumption behind input gradient attribution methods:

[*] Part of the work completed after joining Google Research India

[2] In Appendix C, we show that our results also hold for input gradients taken w.r.t. the *loss*

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

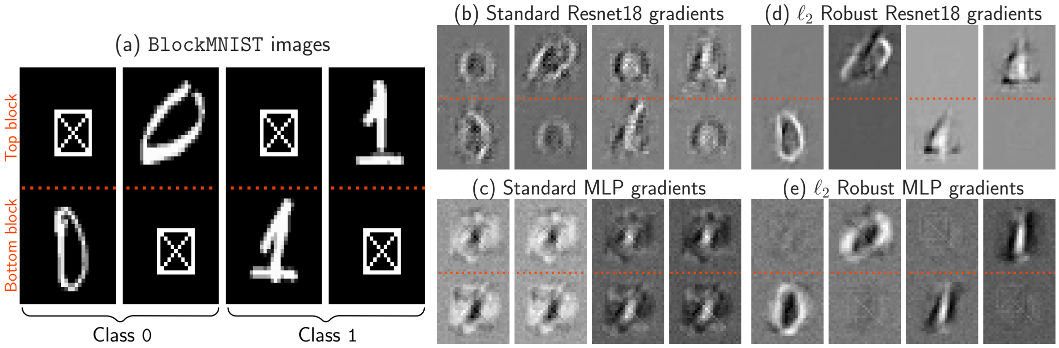
(a) BlockMNIST images
Top block
Bottom block
Class 0
Class 1
(b) Standard Resnet18 gradients
(c) Standard MLP gradients
(d) ℓ_2 Robust Resnet18 gradients
(e) ℓ_2 Robust MLP gradients

Figure 1: Experiments on `BlockMNIST` dataset. (a) Four representative images from class 0 & class 1 in `BlockMNIST` dataset; every image consists of a *signal* and *null* block that are randomly placed as the *top* or *bottom* block. The *signal* block, containing the `MNIST` digit, determines the image class. The *null* block, containing the square patch, does not encode any information of the image class. For these four images, subplots (b-e) show the input gradients of standard Resnet18, standard MLP, ℓ_2 robust Resnet18 (ϵ=2) and ℓ_2 robust MLP (ϵ=4) respectively. The plots clearly show that input gradients of standard `BlockMNIST` models highlight the signal block *and the non-discriminative null block*, thereby violating (`A`). In contrast, input gradients of adversarially robust models exclusively highlight the signal block, suppress the null block, and satisfy (`A`). Please see Section 5 for details.

Assumption (`A`): *Coordinates with larger input gradient magnitude are more relevant for model prediction compared to coordinates with smaller input gradient magnitude.*

Sanity-checking attribution methods. Several attribution methods [10] are based on input gradients and explicitly or implicitly assume an appropriately modified version of (`A`). For example, Integrated Gradients [9] aggregate input gradients of linearly interpolated points, SmoothGrad [2] averages input gradients of points perturbed using gaussian noise, and Guided Backprop [11] modifies input gradients by zeroing out negative values at every layer during backpropagation. Surprisingly, unlike vanilla input gradients, popular methods that output attributions with better visual quality fail simple sanity checks that are indeed expected out of any valid attribution method [12, 13]. On the other hand, while vanilla input gradients pass simple sanity checks, Hooker et al. [14] suggest that they produce estimates of feature importance that are no better than a random designation of feature importance.

Do input gradients satisfy assumption (`A`)? Since (`A`) is necessary for input gradients attributions to accurately reflect model behavior, we introduce an evaluation framework, `DiffROAR`, to analyze whether input gradient attributions satisfy assumption (`A`) on real-world datasets. While `DiffROAR` adopts the remove-and-retrain (`ROAR`) methodology [14], `DiffROAR` is more appropriate for testing the validity of assumption (`A`) because it directly compares top-ranked features against bottom-ranked features. We apply `DiffROAR` to evaluate input gradient attributions of MLPs & CNNs trained on multiple image classification datasets. Consistent with the message in Hooker et al. [14], our experiments indicate that input gradients of standard models (i.e., trained on original data) can grossly violate (`A`) (see Section 4). Furthermore, we also observe that unlike standard models, adversarially trained models [15] that are robust to ℓ_2 and ℓ_∞ perturbations satisfy (`A`) in a consistent manner.

Probing input gradient attributions using `BlockMNIST`. Our empirical findings mentioned above strongly suggest that standard models grossly violate (`A`). However, without knowledge of ground-truth discriminative features learned by models trained on real data, *conclusively* testing (`A`) remains elusive. In fact, this is a key shortcoming of the remove-and-retrain (`ROAR`) framework. So, to further verify and better understand our empirical findings, we introduce an `MNIST`-based semi-real dataset, `BlockMNIST`, that *by design* encodes a priori knowledge of ground-truth discriminative features. `BlockMNIST` is based on the principle that for different inputs, discriminative and non-discriminative features may occur in different parts of the input. For example, in an object classification task, the object of interest can occur in different parts of the image (e.g., top-left, center, bottom-right etc.) for different images. As shown in Figure 1(a), `BlockMNIST` images consist of a *signal* block and a *null* block that are randomly placed at the top or bottom. The *signal* block contains the `MNIST` digit that determines the class of the image, whereas the *null* block, contains a square patch with two diagonals that has no information about the label. This a priori knowledge of ground-truth discriminative features in BlockMNIST data allows us to (i) validate our empirical findings vis-a-vis input gradients of standard and robust models (see fig. 1) and (ii) identify *feature leakage* as a reason that potentially explains why input gradients violate (A) in practice. Here, feature leakage refers to the phenomenon wherein given an instance, its input gradients highlight the location of discriminative features in the given instance *as well as* in other instances that are present in the dataset. For example, consider the first BlockMNIST image in fig. 1(a), in which the signal is placed in the bottom block. For this image, as shown in fig. 1(b,c), input gradients of standard models incorrectly highlight the top block *because* there are *other* instances in the BlockMNIST dataset which have signal in the top block.

Rigorously demonstrating feature leakage. In order to concretely verify as well as understand feature leakage more thoroughly, we design a simplified version of BlockMNIST that is amenable to theoretical analysis. On this dataset, we first rigorously demonstrate that input gradients of standard one-hidden-layer MLPs exhibit feature leakage in the infinite-width limit and then discuss how feature leakage results in input gradient attributions that clearly violate assumption (A).

Paper organization: Section 2 discusses related work and section 3 presents our evaluation framework, DiffROAR, to test assumption (A). Section 4 employs DiffROAR to evaluate input gradient attributions on four image classification datasets. Section 5 analyzes BlockMNIST data to differentially characterize input gradients of standard and robust models using feature leakage. Section 6 provides theoretical results on a simplified version on BlockMNIST that shed light on how feature leakage results in input gradients that violate assumption (A). Our code, along with the proposed datasets, is publicly available at https://github.com/harshays/inputgradients.

2 Related work

Due to space constraints, we only discuss directly related work and defer the rest to Appendix A.

Sanity checks for explanations. Several explanation methods that provide feature attributions are often primarily evaluated using inherently subjective visual assessments [1, 2]. Unsurprisingly, recent "sanity checks" show that sole reliance on visual assessment is misleading, as attributions can lack fidelity and inaccurately reflect model behavior. Adebayo et al. [12] and Kindermans et al. [13] show that unlike input gradients [8], other popular methods—guided backprop [16], gradient $\odot$ input [17], integrated gradients [9]—output explanations which lack fidelity on image data, as they remain invariant to model and label randomization. Similarly, Yang and Kim [18] use custom image datasets to show that several explanation methods are more likely to produce false positive explanations than vanilla input gradients. Moreover, several explanation methods based on modified backpropagation do not pass basic sanity checks [19, 20, 21]. To summarize, well-known gradient-based attribution methods that seek to mitigate gradient saturation [9, 22], discontinuity [23], and visual noise [16] surprisingly fare worse than vanilla input gradients on multiple sanity checks.

Evaluating explanation fidelity. The black-box nature of neural networks necessitates frameworks that evaluate the fidelity or "correctness" of post-hoc explanations *without* knowledge of ground-truth features learned by trained models. Modification-based evaluation frameworks [24, 25, 26] gauge explanation fidelity by measuring the change in model performance after masking input coordinates that a given explanation method considers most (or least) important. However, due to distribution shifts induced by input modifications, one cannot *conclusively* attribute changes in model performance to the fidelity of instance-specific explanations [27]. The remove-and-retrain (ROAR) framework [14] accounts for distribution shifts by evaluating the performance of models *retrained* on train data masked using post-hoc explanations. Surprisingly, contrary to findings obtained via sanity checks, experiments with the ROAR framework show that multiple attribution methods, *including* vanilla input gradients, are no better than model-independent *random* attributions that lack explanatory power [14]. Therefore, motivated by the central role of vanilla input gradients in attribution methods, we augment the ROAR framework to understand when and why input gradients violate assumption (A).

Effect of adversarial robustness. Adversarial training [15] not only leads to robustness to ℓ_p adversarial attacks [28], but also leads to perceptually-aligned feature representations [29], and improved visual quality of input gradients [30]. Recent works hypothesize that adversarial training improves the visual quality of input gradients by suppressing irrelevant features [31] and promoting sparsity and stability [32] in explanations. Kim et al. [33] use the ROAR framework to conjecture that adversarial training "tilts" input gradients to better align with the data manifold. In this work, we use experiments on real-world data and theory on data with features known *a priori* in order to differentially characterize input gradients of standard and robust models vis-a-vis assumption (A).

3 DiffROAR evaluation framework

In this section, we introduce our evaluation framework, DiffROAR, to probe the extent to which instance-specific explanations, or feature attributions, highlight discriminative features in practice. Specifically, our framework, DiffROAR, builds upon the remove-and-retrain (ROAR) methodology [14] to test whether feature attribution methods satisfy assumption (A) on real-world datasets.

Setting. We consider the standard classification setting; Each data point $(x^{(i)}, y^{(i)})$, where instance $x^{(i)} \in \mathbb{R}^d$ and label $y^{(i)} \in \mathcal{Y}$ for some label set $\mathcal{Y}$, is drawn independently from a distribution $\mathcal{D}$ on $\mathbb{R}^d \times \mathcal{Y}$. Given dataset $\{(x^{(i)}, y^{(i)})\}$ where $i \in [n]{:=}\{1, \cdots, n\}$, $x_j^{(i)}$ denotes the j^{th} coordinate of $x^{(i)}$. Note that we also refer to the d coordinates of instance $x^{(i)}$ as *features* interchangeably.

Attribution schemes. A *feature attribution* scheme $\mathcal{A} : \mathbb{R}^d \to \{\sigma : \sigma \text{ is a permutation of } [d]\}$ maps a d-dimensional instance x to a permutation, or ordering, $\mathcal{A}(x) : [d] \to [d]$ of its coordinates. For example, the *input gradient attribution* scheme takes as input instance x & predicted label $\hat{y}$ and outputs an ordering $[d]$ that ranks coordinates in decreasing order of their input gradient magnitude. That is, coordinate j is ranked ahead of coordinate k if the magnitude of the j^{th} coordinate of $\nabla_x \text{Logit}_\theta(x, \hat{y})$ is larger than that of the k^{th} coordinate.

Unmasking schemes. Given instance x and a subset $S \subseteq [d]$ of coordinates, the *unmasked* instance x^S zeroes out all coordinates that are not in subset S: $x_j^S = x_j$ if $j \in S$ and 0 if $j \notin S$. An *unmasking scheme* $A : \mathbb{R}^d \to \{S : S \subseteq [d]\}$ simply maps instance x to a subset $A(x) \subseteq [d]$ of coordinates that can be used to obtain unmasked instance $x^{A(x)}$. Any attribution scheme $\mathcal{A}$ naturally induces *top-k* and *bottom-k* unmasking schemes, $\mathcal{A}_k^{\text{top}}$ and $\mathcal{A}_k^{\text{bot}}$, which output k coordinates with the top-most and bottom-most attributions in $\mathcal{A}(x)$ respectively. In other words, given attribution scheme $\mathcal{A}$ and level k, the top-k and bottom-k unmasking schemes, $\mathcal{A}_k^{\text{top}}$ and $\mathcal{A}_k^{\text{bot}}$, can be defined as follows:

$$\mathcal{A}_k^{\text{top}}(x) {:=} \{\mathcal{A}(x)_j : j \le k\},$$
$$\mathcal{A}_k^{\text{bot}}(x) {:=} \{\mathcal{A}(x)_j : d - k < j \le d\}.$$

For example, Figure 2 depicts an image x and its top-k unmasked variant $x^{\mathcal{A}_k^{\text{top}}(x)}$. In this case, the attribution scheme $\mathcal{A}$ assigns higher rank to pixels in the foreground. So, the top-25% unmasking operation, $x^{\mathcal{A}_{25\%}^{\text{top}}(x)}$, highlights the monkey by retaining pixels with top-25% attribution ranks and zeroing out the remaining pixels that correspond to the green background.

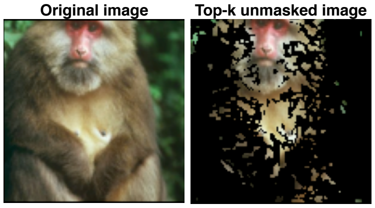

Figure 2: Pictorial example of a top-25% unmasked image.

Predictive power of unmasking schemes. The *predictive power* of an unmasking scheme A with respect to model architecture M (e.g., resnet18) can be defined as the best classification accuracy that can be attained by training a model with architecture M on unmasked instances that are obtained via unmasking scheme A. More formally, it can defined as follows:

$$\texttt{PredPower}_M(A) := \sup_{f \in M, f:\mathbb{R}^d \to \mathcal{Y}} \mathbb{E}_{\mathcal{D}} \left[\mathbb{1} \left[f(x^{A(x)}) = y \right] \right].$$

Due to masking-induced distribution shifts, models with architecture M that are trained using original data cannot be plugged in to estimate $\texttt{PredPower}_M(A)$. The ROAR framework [14] sidesteps this issue by *retraining* models on unmasked data, as similar model architectures tend to learn "similar" classifiers [34, 35, 36, 37]. Therefore, we employ the ROAR framework to estimate $\texttt{PredPower}_M(A)$ in two steps. First, we use unmasking scheme A to obtain *unmasked* train and test datasets that comprise data points of the form $(x^{A(x)}, y)$. Then, we *retrain* a new model with the same architecture M on unmasked train data and evaluate its accuracy on unmasked test data.

DiffROAR evaluation metric to test assumption (A). Recall that an attribution scheme $\mathcal{A}$ maps an instance x to a permutation of its coordinates that reflects the order of *estimated* importance in model prediction. An attribution scheme that satisfies assumption (A) must place coordinates that are more important for model prediction higher up in the the attribution order. More formally, given attribution scheme $\mathcal{A}$, architecture M and level k, we define DiffROAR as the difference between the predictive power of top-k and bottom-k unmasking schemes, $\mathcal{A}_k^{\text{top}}$ and $\mathcal{A}_k^{\text{bot}}$:

$$\texttt{DiffROAR}_M(\mathcal{A}, k) = \texttt{PredPower}_M(\mathcal{A}_k^{\text{top}}) - \texttt{PredPower}_M(\mathcal{A}_k^{\text{bot}}) \tag{1}$$

Interpreting the `DiffROAR` metric. The sign of the `DiffROAR` metric indicates whether the given attribution scheme satisfies or violates assumption (`A`). For example, $\texttt{DiffROAR}_M(\mathcal{A}, \cdot) < 0$ implies that $\mathcal{A}$ violates assumption (`A`) , as coordinates with *higher* attribution ranks have *worse* predictive power with respect to architecture M. Similarly, the magnitude of the `DiffROAR` metric quantifies the extent to which the ordering in attribution scheme $\mathcal{A}$ separates the most and least discriminative coordinates into two disjoint subsets. For example, a *random* attribution scheme $\mathcal{A}_r$, which outputs attributions $\mathcal{A}_r(x)$ chosen uniformly at random from all permutations of $[d]$, neither highlights nor suppresses discriminative features; $\mathbb{E}[\texttt{DiffROAR}_M(\mathcal{A}_r, k)] = 0$ for any architecture M.

On testing assumption (`A`). To verify (`A`) for a given attribution scheme $\mathcal{A}$, it is necessary to evaluate whether input coordinates with *higher* attribution rank are *more* important for model prediction than coordinates with *lower* rank. Consequently, the `ROAR`-based metric in Hooker et al. [14], which essentially computes the top-k predictive power, is not sufficient to test whether attribution methods satisfy assumption (`A`). Therefore, as discussed above, `DiffROAR` tests (`A`) by comparing the top-k predictive power, $\texttt{PredPower}_M(\mathcal{A}_k^{\text{top}})$, to the bottom-$k$ predictive power, $\texttt{PredPower}_M(\mathcal{A}_k^{\text{bot}})$, using multiple values of k.

4 Testing assumption (`A`) on image classification benchmarks

In this section, we use `DiffROAR` to evaluate whether input gradient attributions of standard and adversarially robust MLPs and CNNs trained on four image classification benchmarks satisfy assumption (`A`). We first summarize the experiment setup and then describe key empirical findings.

Datasets and models. We consider four benchmark image classification datasets: SVHN [38], Fashion MNIST [39], CIFAR-10 [40] and ImageNet-10 [41]. ImageNet-10 is an open-sourced variant (`https://github.com/MadryLab/robustness/`) of Imagenet [41], with $80,000$ images grouped into 10 super-classes. ImageNet-10 enables us to test assumption (`A`) on Imagenet without the computational overload of training models on the 1000-way ILSVRC classification task [42]. We evaluate input gradient attributions of standard and adversarially trained two-hidden-layer MLPs and Resnets [43]. We obtain ℓ_2 and ℓ_∞ ϵ-robust models with perturbation budget ϵ using PGD adversarial training [15]. Unless mentioned otherwise, we train models using stochastic gradient descent (SGD), with momentum 0.9, batch size 256, ℓ_2 regularization 0.0005 and initial learning rate 0.1 that decays by a factor of 0.75 every 20 epochs. Additionally, we use standard data augmentation and train models for at most 500 epochs, stopping early if cross-entropy loss on training data goes below 0.001. Appendix C.1 provides additional details about the datasets and trained models.[3]

Estimating `DiffROAR` on real data. We compute the evaluation metric, $\texttt{DiffROAR}_M(\mathcal{A}, k)$, on real datasets in four steps, as follows. First, we train a standard or robust model with architecture M on the original dataset and obtain its input gradient attribution scheme $\mathcal{A}$. Second, as outlined in Section 3, we use attribution scheme $\mathcal{A}$ and level k (i.e., fraction of pixels to be unmasked) to extract the top-k and bottom-k unmasking schemes: $\mathcal{A}_k^{\text{top}}$ and $\mathcal{A}_k^{\text{bot}}$. Third, we apply $\mathcal{A}_k^{\text{top}}$ and $\mathcal{A}_k^{\text{bot}}$ on the original train & test datasets to obtain top-k and bottom-k unmasked datasets respectively. Finally, to compute $\texttt{DiffROAR}_M(\mathcal{A}, k)$ via eq. (1), we estimate top-k and bottom-k predictive power, $\texttt{PredPower}_M(\mathcal{A}_k^{\text{top}})$ and $\texttt{PredPower}_M(\mathcal{A}_k^{\text{bot}})$, by *retraining new models* with architecture M on top-k and bottom-k unmasked datasets respectively. Also, note that we (a) average the `DiffROAR` metric over five runs for each model and unmasking fraction or level k and (b) unmask individual image pixels without grouping them channel-wise.

Experiment setup. Now, we analyze the `DiffROAR` metric as a function of the unmasking fraction $k \in \{5, 10, 20, \dots, 100\}\%$ in order to evaluate whether input gradient attributions of models trained on four image classification benchmarks satisfy assumption (`A`). In particular, as shown in Figure 3, we use `DiffROAR` to analyze input gradients of standard and adversarially robust two-hidden-layer MLPs on SVHN & Fashion MNIST, Resnet18 on ImageNet-10, and Resnet50 on CIFAR-10. In order to calibrate our findings, we compare input gradient attributions of these models to two natural baselines: model-agnostic *random* attributions and input-agnostic attributions of linear models.

Input gradients of standard models. Input gradient attributions of standard MLPs trained on SVHN satisfy assumption (`A`), as the `DiffROAR` metric in Figure 3(a) is positive for all values of level k < 100%. However, in Figure 3(b), the `DiffROAR` curves of standard MLPs trained on

[3] Code publicly available at `https://github.com/harshays/inputgradients`

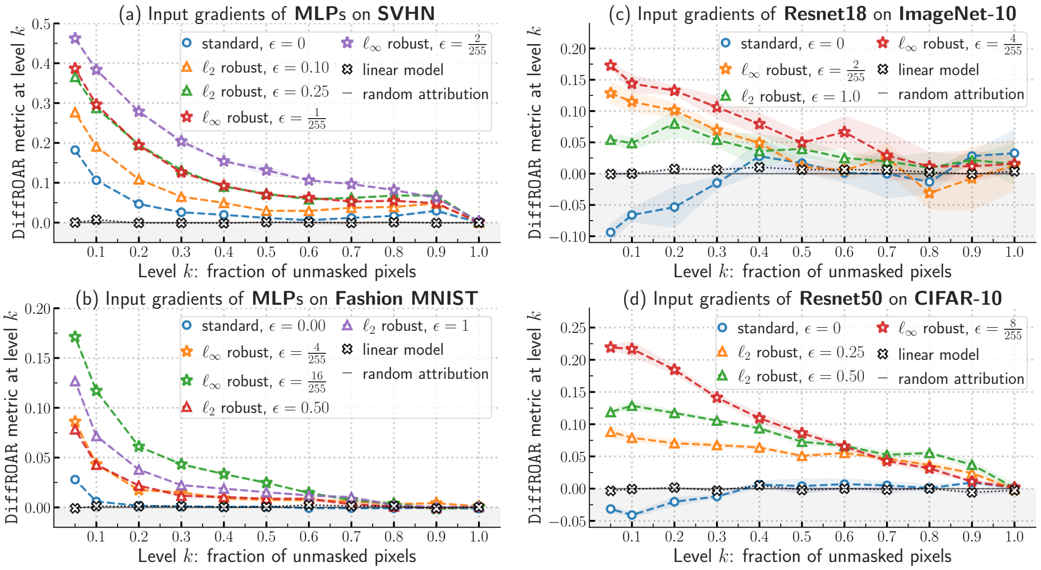

Figure 3: DiffROAR plots for input gradient attributions of standard and adversarially robust two-hidden-layer MLPs on (a) SVHN & (b) Fashion MNIST, (c) Resnet18 on ImageNet-10 and (d) Resnet50 on CIFAR-10. Subplot (a) indicates that adversarially robust MLPs consistently and considerably outperform standard MLPs on the DiffROAR metric for all $k < 100\%$. Subplot (b) shows that for most unmasking fractions k, standard MLPs trained on Fashion MNIST, unlike robust MLPs, fare no better than model-agnostic random attributions and input-agnostic attributions of linear models. Subplots (c) and (d) show that when $k < 40\%$, standard Resnet models trained on CIFAR-10 and ImageNet-10 grossly violate (A), thereby implying that coordinates with top-most gradient attribution rank have worse predictive power than coordinates with bottom-most rank. In stark contrast, input gradients of Resnets that are robust to ℓ_2 and ℓ_∞ adversarial perturbations satisfy assumption (A) reasonably well. We observe that increasing the perturbation budget ϵ during adversarial training amplifies the magnitude of DiffROAR for every k across all four image classification benchmarks.

Fashion MNIST indicate that input gradient attributions, consistent with findings in Hooker et al. [14], can fare no better than model-agnostic random attributions and input-agnostic attributions of linear models vis-a-vis assumption (A). Furthermore, and rather surprisingly, the shaded area in Figure 3(c) and Figure 3(d) shows that when level $k < 40\%$, DiffROAR curves of standard Resnets trained on CIFAR-10 and Imagenet-10 are consistently *negative* and perform considerably worse than model-agnostic and input-agnostic baseline attributions. These results strongly suggest that on CIFAR-10 and Imagenet-10, input gradients of standard Resnets grossly violate assumption (A) and suppress discriminative features. In other words, coordinates with *larger* gradient magnitude have *worse* predictive power than coordinates with *smaller* gradient magnitude.

Input gradients of robust models. Models that are ϵ-robust to ℓ_2 and ℓ_∞ adversarial perturbations fare considerably better than standard models on the DiffROAR metric. For example, in Figure 3(a), when level k equals 10%, robust MLPs trained on SVHN outperform standard MLPs on the DiffROAR metric by roughly 10-30%. The DiffROAR curves of adversarially robust MLPs in Figure 3 are positive at every level $k < 100\%$, which strongly suggests that input gradient attributions of robust MLPs satisfy assumption (A). Similarly, robust resnet50 models trained on CIFAR-10 and ImageNet-10 satisfy assumption (A) reasonably well and, unlike standard resnet50 models, starkly highlight discriminative features. Furthermore, we observe that increasing the perturbation budget ϵ in ℓ_2 or ℓ_∞ PGD adversarial training [15] amplifies the magnitude of DiffROAR across k and for all four datasets. That is, the adversarial perturbation budget ϵ determines the extent to which input gradients differentiates the most and least discriminative coordinates into two disjoint subsets.

Additional results. In Appendix C, we show that our DiffROAR results are robust to choice of model architecture & SGD hyperparameters during retraining and also hold for input gradients taken with respect to cross-entropy loss. Additionally, while DiffROAR *without retraining* gives qualitatively similar results, they are not as consistent across architectures as with retraining, particularly for small unmasking fraction k that induce non-trivial distribution shifts.

5 Analyzing input gradient attributions using BlockMNIST data

To verify whether input gradients satisfy assumption (A) more thoroughly, we introduce and perform experiments on BlockMNIST, an MNIST-based dataset that *by design* encodes a priori knowledge of ground-truth discriminative features.

BlockMNIST **dataset design**: The design of the BlockMNIST dataset is based on two intuitive properties of real-world object classification tasks: (i) for different images, the object of interest may appear in different parts of the image (e.g., top-left, bottom-right); (ii) the object of interest and the rest of the image often share low-level patterns such as edges that are not informative of the label on their own. We replicate these aspects in BlockMNIST instances, which are vertical concatenations of two 28×28 *signal* and *null* image blocks that are randomly placed at the top or bottom with equal probability. The *signal* block is an MNIST image of digit 0 or digit 1, corresponding to class 0 or 1 of the BlockMNIST image respectively. On the other hand, the *null* block in every BlockMNIST image, independent of its class, contains a square patch made of two horizontal, vertical, and slanted lines, as shown in Figure 1(a). It is important to note that unlike the MNIST signal block that is fully predictive of the class, the non-discriminative null block contains no information about the class. Standard as well as adversarially robust models trained on BlockMNIST data attain 99.99% test accuracy, thereby implying that model predictions are indeed based solely on the signal block for any given instance. We further verify this by noting that the predictions of trained model remain unchanged on almost every instance even when all pixels in the null block are set to zero.

Do standard and robust models satisfy (A)? As discussed above, unlike the null block that has no task-relevant information, the MNIST digit in the signal block entirely determines the class of any given BlockMNIST image. Therefore, in this setting, we can restate assumption (A) as follows: *Do input gradient attributions highlight the signal block over the null block?* Surprisingly, as shown in Figure 1(b,c), input gradient attributions of standard MLP and Resnet18 models highlight the signal block *as well as* the non-discriminative null block. In stark contrast, subplots (d) and (e) in Figure 1 show that input gradient attributions of ℓ_2 robust MLP and Resnet18 models exclusively highlight MNIST digits in the signal block and clearly suppress the square patch in the null block. These results validate our findings on real-world datasets by showing that unlike standard models, adversarially robust models satisfy (A) on BlockMNIST data.

Feature leakage hypothesis: Recall that the discriminative signal block in BlockMNIST images is randomly placed at the top or bottom with equal probability. Given our results in Figure 1, we hypothesize that when discriminative features vary across instances (e.g., signal block at top vs. bottom), input gradients of standard models not only highlight instance-specific features but also *leak* discriminative features from other instances. We term this hypothesis *feature leakage*. To test our hypothesis, we leverage the modular structure in BlockMNIST to construct a slightly modified version, BlockMNIST-Top, wherein the location of the MNIST signal block is fixed at the top for all instances (see fig. 4). In this setting, in contrast to results on BlockMNIST, input gradients of *standard* Resnet18 and MLP models trained on BlockMNIST-Top satisfy assumption (A). Specifically, when the signal block is fixed at the top, input gradient attributions in Figure 4(b, c) clearly highlight the signal block and suppress the null block, thereby supporting our feature leakage hypothesis. Based on our BlockMNIST experiments, we believe that understanding *how* adversarial robustness mitigates feature leakage is an interesting direction for future work.

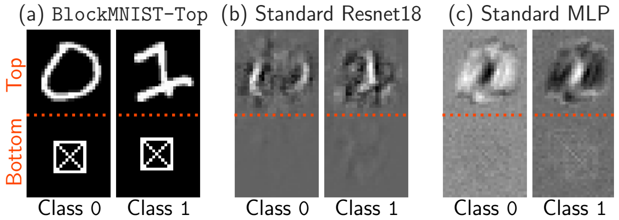

Figure 4: (a) In BlockMNIST-Top images, the signal & null blocks are fixed at the top & bottom respectively. In contrast to results on BlockMNIST in fig. 1, input gradients of standard (b) Resnet18 and (c) MLP trained on BlockMNIST-Top highlight discriminative features in the signal block, suppress the null block, and satisfy (A).

Additional results. In Appendix D.1, we (i) visualize input gradients of several BlockMNIST and BlockMNIST-Top images, (ii) introduce a quantitative proxy metric to compare feature leakage between standard and robust models, (iii) show that our findings are fairly robust to the choice and number of classes in BlockMNIST data, and (iv) evaluate feature leakage in five feature attribution methods. We also provide experiments that falsify hypotheses vis-a-vis input gradients and assumption (A) that we considered in addition to feature leakage.

6 Feature leakage in input gradient attributions

To understand the extent of feature leakage more thoroughly, we introduce a simplified version of the `BlockMNIST` dataset that is amenable to theoretical analysis. We rigorously show that input gradients of standard one-hidden-layer MLPs do not differentiate instance-specific features from other task-relevant features that are not pertinent to the given instance.

Dataset: Given dimension of each block $\widetilde{d}$, feature vector $u^* \in \mathbb{R}^{\widetilde{d}}$ with $\|u^*\| = 1$, number of blocks d and noise parameter η, we will construct input instances of dimension $\widetilde{d} \cdot d$. More concretely, a sample $(x, y) \in \mathbb{R}^{\widetilde{d} \cdot d} \times \{\pm 1\}$ from the distribution $\mathcal{D}$ is generated as follows:

$$\begin{aligned} & y = \pm 1 \text{ with probability 0.5 and} \\ & x = [\eta g_1, \quad \eta g_2, \quad \ldots, \quad y u^* + \eta g_j, \quad \ldots, \quad \eta g_d] \text{ with } j \text{ chosen at random from } [d/2] \end{aligned} \tag{2}$$

where each $g_i \in \mathbb{R}^{\widetilde{d}}$ is drawn uniformly at random from the unit ball. For simplicity, we take d to be even so that $d/2$ is an integer. We can think of each x as a concatenation of d $\widetilde{d}$-dimensional blocks $\{x_1, \ldots, x_d\}$. The first $d/2$ blocks, $\{1, \ldots, d/2\}$, are *task-relevant*, as every example (x, y) contains an instance-specific *signal* block $x_i = y u^* + \eta g_i$ for some $i \in [d/2]$ that is informative of its label y. Given instance x, we use $j^*(x)$ to denote the unique instance-specific signal block such that $x_{j^*(x)} = y u^* + \eta g_{j^*(x)}$. On the other hand, *noise* blocks $\{d/2 + 1, \ldots, d\}$ do not contain task-relevant signal for any instance x. At a high level, the instance-specific signal block $j^*(x)$ and noise blocks $\{d/2 + 1, \ldots, d\}$ in instance x correspond to the discriminative `MNIST` digit and the null square patch in `BlockMNIST` images respectively. For example, each row in Figure 5(a) illustrates an instance x where $d = 10, \widetilde{d} = 1, \eta = 0$ and $u^* = 1$.

Model: We consider one-hidden layer MLPs with ReLU nonlinearity in the infinite-width limit. More concretely, for a given width m, the network is parameterized by $R \in \mathbb{R}^{m \times \widetilde{d} \cdot d}$, $b \in \mathbb{R}^m$ and $w \in \mathbb{R}^m$. Given an input instance $(x, y) \in \mathbb{R}^{\widetilde{d}d} \times \{\pm 1\}$, the output score (or logit) f and cross-entropy (CE) loss $\mathcal{L}$ are given by:

$$f((w, R, b), x) := \langle w, \phi(Rx + b) \rangle, \quad \mathcal{L}((w, R, b), (x, y)) := \log\left(1 + \exp\left(-y \cdot f((w, R, b), x)\right)\right).$$

where $\phi(t) := \max(0, t)$ denotes the ReLU function. A remarkable set of recent results [44, 45, 46, 47] show that as $m \to \infty$, the training procedure is equivalent to gradient descent (GD) on an infinite dimensional Wasserstein space. In the Wasserstein space, the network can be interpreted as a probability distribution ν over $\mathbb{R} \times \mathbb{R}^{\widetilde{d} \cdot d} \times \mathbb{R}$ with output score f and cross entropy loss $\mathcal{L}$ defined as:

$$f(\nu, x) := \mathbb{E}_{(w, r, b) \sim \nu}\left[w \phi(\langle r, x \rangle + b)\right], \quad \mathcal{L}(\nu, (x, y)) := \log\left(1 + \exp\left(-y \cdot f(\nu, x)\right)\right). \tag{3}$$

Theoretical analysis: Our approach leverages the recent result in Chizat and Bach [48], which shows that if GD in the Wasserstein space $\mathcal{W}^2\left(\mathbb{R} \times \mathbb{R}^{\widetilde{d}d} \times \mathbb{R}\right)$ on $\mathbb{E}_{\mathcal{D}}\left[\mathcal{L}(\nu, (x, y))\right]$ converges, it does so to a max-margin classifier given by:

$$\nu^* := \underset{\nu \in \mathcal{P}\left(\mathbb{S}^{d\widetilde{d}+1}\right)}{\arg\max} \; \min_{(x,y) \sim \mathcal{D}} y \cdot f(\nu, x), \tag{4}$$

where $\mathbb{S}^{d\widetilde{d}+1}$ denotes the surface of the Euclidean unit ball in $\mathbb{R}^{\widetilde{d}d+2}$, and $\mathcal{P}\left(\mathbb{S}^{d\widetilde{d}+1}\right)$ denotes the space of probability distributions over $\mathbb{S}^{d\widetilde{d}+1}$. Intuitively, our main result shows that on any data point $(x, y) \sim \mathcal{D}$, the input gradient magnitude of the max-margin classifier ν^* is *equal* over all task-relevant blocks $\{1, \ldots, d/2\}$ and zero on the remaining *noise* blocks $\{d/2 + 1, \ldots, d\}$.

Theorem 1. *Consider distribution $\mathcal{D}$ (2) with $\eta < \frac{1}{10d}$. There exists a max-margin classifier ν^* for $\mathcal{D}$ in Wasserstein space (i.e., training both layers of FCN with $m \to \infty$) given by (4), such that for all $\forall$ $(x, y) \sim \mathcal{D}$: (i) $\left\|\left(\nabla_x \mathcal{L}(\nu^*, (x, y))\right)_j\right\| = c > 0$ for every $j \in [d/2]$ and (ii) $\left\|\left(\nabla_x \mathcal{L}(\nu^*, (x, y))\right)_j\right\| = 0$ for every $j \in \{d/2 + 1, \cdots, d\}$, where $\left(\nabla_x \mathcal{L}(\nu^*, (x, y))\right)_j$ denotes the j^{th} block of the input gradient $\nabla_x \mathcal{L}(\nu^*, (x, y))$.*

Theorem 1 guarantees the *existence* of a max-margin classifier such that the input gradient magnitude for any given instance is (i) a non-zero constant on each of the first $d/2$ task-relevant blocks, and

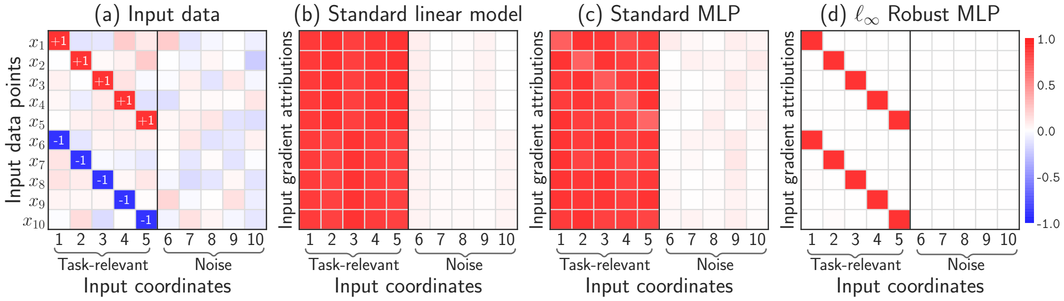

Figure 5: Input gradients of linear models and standard & robust MLPs trained on data from eq. (2) with $d = 10, \widetilde{d} = 1, \eta = 0$ and $u^* = 1$. (a) Each row in corresponds to an instance x, and the highlighted coordinate denotes the signal block $j^*(x)$ & label y. (b) Linear models suppress noise coordinates but lack the expressive power to highlight instance-specific signal $j^*(x)$, as their input gradients in subplot (b) are identical across all examples. (c) Despite the expressive power to highlight instance-specific signal coordinate $j^*(x)$, input gradients of standard MLPs exhibit feature leakage (see Theorem 1) and violate (A) as well. (d) In stark contrast, input gradients of adversarially trained MLPs suppress feature leakage and starkly highlight instance-specific signal coordinates $j^*(x)$.

(ii) equal to zero on the remaining $^d/_2$ *noise* blocks that do not contain any information about the label. However, input gradients fail at highlighting the *unique instance-specific signal* block over the remaining *task-relevant* blocks. This clearly demonstrates feature leakage, as input gradients for any given instance also highlight task-relevant features that are, in fact, *not specific* to the given instance. Therefore, input gradients of standard one-hidden-layer MLPs do not highlight instance-specific discriminative features and grossly violate assumption (A). In Appendix F, we present additional results that demonstrate that adversarially trained one-hidden-layer MLPs can suppress feature leakage and satisfy assumption (A).

Empirical results: Now, we supplement our theoretical results by evaluating input gradients of linear models as well as standard & robust one-hidden-layer ReLU MLPs with width 10000 on the dataset shown in Figure 5. Note that all models obtain 100% test accuracy on this linearly separable dataset, a simplified version of BlockMNIST that is obtained via eq. 2 with $d = 10, \widetilde{d} = 1, \eta = 0$ and $u^* = 1$. Due to insufficient expressive power, linear models have input-agnostic gradients that suppress all five noise coordinates, but do not differentiate the instance-specific signal coordinate from the remaining task-relevant coordinates. Consistent with Theorem 1, even standard MLPs, which are expressive enough to have input gradients that correctly highlight instance-specific coordinates, apply equal weight on all five task-relevant coordinates and violate (A) due to feature leakage. On the other hand, Figure 5(c) shows that the same MLP architecture, if robust to ℓ_∞ adversarial perturbations with norm 0.35, satisfies (A) by clearly highlighting the instance-specific signal coordinate over all other noise *and* task-relevant coordinates

7 Discussion and conclusion

In this work, we took a three-pronged approach to investigate the validity of a key assumption made in several popular post-hoc attribution methods: (A) *coordinates with larger input gradient magnitude are more relevant for model prediction compared to coordinates with smaller input gradient magnitude*. Through (i) evaluation on real-world data using our DiffROAR framework, (ii) empirical analysis on BlockMNIST data that encodes information of ground-truth discriminative features, and (iii) a rigorous theoretical study, we present strong evidence to suggest that standard models do not satisfy assumption (A). In contrast, adversarially robust models satisfy (A) in a consistent manner. Furthermore, our analysis in Section 5 and Section 6 indicates that *feature leakage* sheds light on why input gradients of standard models tend to violate (A). We provide additional discussion in Appendix B.

This work exclusively focused on “vanilla" input gradients due to their fundamental significance in *feature attribution*. A similarly thorough investigation that analyzes other commonly-used attribution methods is an interesting avenue for future work. Another interesting avenue for further analyses is to understand how adversarial training mitigates feature leakage in input gradient attributions.

References

[1] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[2] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[3] Matthew L. Leavitt and Ari S. Morcos. Towards falsifiable interpretability research. *ArXiv*, abs/2010.12016, 2020.

[4] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

[5] Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 700–712. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/075b051ec3d22dac7b33f788da631fd4-Paper.pdf`.

[6] John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11):e1002683, 2018.

[7] Gregor Stiglic, Primoz Kocbek, Nino Fijacko, Marinka Zitnik, Katrien Verbert, and Leona Cilar. Interpretability of machine learning-based prediction models in healthcare. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(5):e1379, 2020.

[8] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

[9] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[10] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.

[11] J Springenberg, Alexey Dosovitskiy, Thomas Brox, and M Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*, 2015.

[12] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in Neural Information Processing Systems*, 31: 9505–9515, 2018.

[13] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer, 2019.

[14] Sara Hooker, D. Erhan, P. Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *NeurIPS*, 2019.

[15] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJzIBfZAb`.

[16] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[17] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

[18] Mengjiao Yang and Been Kim. Benchmarking attribution methods with relative feature importance. *arXiv preprint arXiv:1907.09701*, 2019.

[19] Leon Sixt, Maximilian Granz, and Tim Landgraf. When explanations lie: Why many modified BP attributions fail. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9046–9057. PMLR, 13–18 Jul 2020. URL `http://proceedings.mlr.press/v119/sixt20a.html`.

[20] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in)fidelity and sensitivity of explanations. In *Advances in Neural Information Processing Systems*, volume 32, pages 10967–10978, 2019.

[21] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Sy21R9JAW`.

[22] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Gradients of counterfactuals. *arXiv preprint arXiv:1611.02639*, 2016.

[23] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017.

[24] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015. doi: 10.1371/journal.pone.0130140. URL `https://doi.org/10.1371/journal.pone.0130140`.

[25] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K. Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673, 2017. doi: 10.1109/TNNLS.2016.2599820.

[26] Leila Arras, Ahmed Osman, Klaus-Robert Müller, and Wojciech Samek. Evaluating recurrent neural network explanations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 113–126, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4813. URL `https://www.aclweb.org/anthology/W19-4813`.

[27] Richard Tomsett, D. Harborne, S. Chakraborty, Prudhvi Gurram, and A. Preece. Sanity checks for saliency metrics. *ArXiv*, abs/1912.01451, 2020.

[28] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

[29] Shibani Santurkar, Dimitris Tsipras, Brandon Tran, Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Image synthesis with a single (robust) classifier. *Advances in Neural Information Processing Systems*, 32, 2019.

[30] A. Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *AAAI*, 2018.

[31] Beomsu Kim, Junghoon Seo, Seunghyun Jeon, Jamyoung Koo, J. Choe, and Taegyun Jeon. Why are saliency maps noisy? cause of and solution to noisy saliency maps. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4149–4157, 2019.

[32] P. Chalasani, J. Chen, A. Chowdhury, S. Jha, and X. Wu. Concise explanations of neural networks using adversarial training. In *ICML*, 2020.

[33] Beomsu Kim, Junghoon Seo, and Taegyun Jeon. Bridging adversarial robustness and gradient interpretability. *ArXiv*, abs/1903.11626, 2019.

[34] Guy Hacohen, Leshem Choshen, and Daphna Weinshall. Let's agree to agree: Neural networks share classification order on real datasets. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3950–3960. PMLR, 13–18 Jul 2020. URL `http://proceedings.mlr.press/v119/hacohen20a.html`.

[35] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent learning: Do different neural networks learn the same representations? In Dmitry Storcheus, Afshin Rostamizadeh, and Sanjiv Kumar, editors, *Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges at NIPS 2015*, volume 44 of *Proceedings of Machine Learning Research*, pages 196–212, Montreal, Canada, 11 Dec 2015. PMLR. URL `http://proceedings.mlr.press/v44/li15convergent.html`.

[36] Dimitris Kalimeris, Gal Kaplun, Preetum Nakkiran, Benjamin Edelman, Tristan Yang, Boaz Barak, and Haofeng Zhang. Sgd on neural networks learns functions of increasing complexity. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/b432f34c5a997c8e7c806a895ecc5e25-Paper.pdf`.

[37] Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9573–9585. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/6cfe0e6127fa25df2a0ef2ae1067d915-Paper.pdf`.

[38] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[39] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[40] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[44] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33): E7665–E7671, 2018.

[45] Lénaïc Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 3040–3050, 2018.

[46] Grant M Rotskoff and Eric Vanden-Eijnden. Trainability and accuracy of neural networks: An interacting particle system approach. *arXiv preprint arXiv:1805.00915*, 2018.

[47] Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of deep neural networks. *Mathematics of Operations Research*, 2021.

[48] Lénaïc Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338. PMLR, 2020.

[49] A. Ghorbani, Abubakar Abid, and James Y. Zou. Interpretation of neural networks is fragile. In *AAAI*, 2019.

[50] Juyeon Heo, Sunghwan Joo, and T. Moon. Fooling neural network interpretations via adversarial model manipulation. In *NeurIPS*, 2019.

[51] Ann-Kathrin Dombrowski, Maximilian Alber, Christopher J. Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché Buc, Emily B. Fox, and Roman Garnett, editors, *NeurIPS*, pages 13567–13578, 2019. URL `http://dblp.uni-trier.de/db/conf/nips/nips2019.html#DombrowskiAAAMK19`.

[52] Naman Bansal, Chirag Agarwal, and Anh M Nguyen. Sam: The sensitivity of attribution methods to hyperparameters. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 11–21, 2020.

[53] Mayank Singh, Nupur Kumari, P. Mangla, Abhishek Sinha, V. Balasubramanian, and Balaji Krishnamurthy. Attributional robustness training using input-gradient spatial alignment. In *ECCV*, 2020.

[54] H. Lakkaraju, Nino Arsov, and Osbert Bastani. Robust and stable black box explanations. In *ICML*, 2020.

[55] E. Chu, Deb Roy, and Jacob Andreas. Are visual explanations useful? a case study in model-in-the-loop prediction. *ArXiv*, abs/2007.12248, 2020.

[56] Forough Poursabzi-Sangdeh, D. Goldstein, J. Hofman, Jennifer Wortman Vaughan, and H. Wallach. Manipulating and measuring model interpretability. *ArXiv*, abs/1802.07810, 2018.

[57] Danish Pruthi, Bhuwan Dhingra, Livio Baldini Soares, M. Collins, Zachary C. Lipton, Graham Neubig, and William W. Cohen. Evaluating explanations: How much do explanations from the teacher aid students? *ArXiv*, abs/2012.00893, 2020.

[58] Peijie Chen, Chirag Agarwal, and Anh Nguyen. The shape and simplicity biases of adversarially robust imagenet-trained cnns. *arXiv preprint arXiv:2006.09373*, 2020.

[59] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14274–14285, 2020.

[60] Thomas Tanay and Lewis Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690*, 2016.

[61] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

[62] Ali Shafahi, W Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? *arXiv preprint arXiv:1809.02104*, 2018.

[63] Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya Razenshteyn. Adversarial examples from computational constraints. In *International Conference on Machine Learning*, pages 831–840. PMLR, 2019.

[64] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *arXiv preprint arXiv:2007.08489*, 2020.

[65] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*, 2019.

[66] Adi Shamir, Odelia Melamed, and Oriel BenShmuel. The dimpled manifold model of adversarial examples in machine learning. *arXiv preprint arXiv:2106.10151*, 2021.

[67] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks (2013). *arXiv preprint arXiv:1311.2901*, 2013.

[68] Lénaïc Chizat. Personal communication, 2021.

Appendices

The supplementary material is organized as follows. We first discuss additional related work Section 2. Appendix B provides additional discussion. Appendix C describes additional experiments based on the `DiffROAR` framework to analyze the fidelity of input gradient attributions on real-world datasets. In Appendix D, we provide additional experiments on feature leakage using `BlockMNIST`-based datasets. Then, Appendix E contains the proof of Theorem 1 and Appendix F discusses the effect of adversarial training on input gradients of models that are adversarially trained on a simplified version of `BlockMNIST` data. We plan to open-source our trained models, code primitives, and Jupyter notebooks soon, which can be used to reproduce our empirical results.

A Additional related work

In this section, we briefly describe works that analyze two properties of post-hoc instance-specific explanations that are related to explanation fidelity or "correctness". In particular, we outline recent works that study the *robustness* and *practical utility* of instance-specific explanation methods.

Robustness of explanations: Several commonly used instance-specific explanation methods lack robustness in practice. Ghorbani et al. [49] show that instance-specific explanations and exempler-based explanations are not robust to imperceptibly small adversarial perturbations to the input. Heo et al. [50] show that instance-specific explanations are highly vulnerable to adversarial *model* manipulations as well. Dombrowski et al. [51] show that explanations lack robustness to to *arbitrary* manipulations and show that non-robustness stems from geometric properties of neural networks. Bansal et al. [52] show that explanation methods are considerably sensitive to method-specific hyperparameters such as sample size, blur radius, and random seeds. Recent works promote robustness in explanations using smoothing [51] or variants of adversarial training [53, 54]

Utility of explanations: A recent line of work propose evaluation frameworks to assess the practical utility of post-hoc instance-specific explanation methods via proxy downstream tasks. Chu et al. [55] employ a randomized controlled trial to show that using explanation methods as additional information does not improve *human* accuracy on classification tasks. More generally, Poursabzi-Sangdeh et al. [56] analyze the effect of model transparency (e.g., number of input features, black-box vs. white-box) on the accuracy of human decisions with respect to the task and model. Similarly, Adebayo et al. [5] conduct a human subject study to show that subjects fail to identify defective models using attributions and instead primarily rely on model predictions. [57] formalize the "value" of explanations as the explanation utility (i.e., as side information) in a student-teacher learning framework. In contrast to the works above, we propose an evaluation framework, `DiffROAR`, to evaluate the fidelity, or "correctness", of explanations in classification tasks. In particular, using benchmark image classification tasks and synthetic data, we empirically and theoretically characterize input gradient attributions of standard as well as adversarially robust models.

Stability of explanations. Explanation stability and explanation correctness (also known as explanation fidelity) are two distinct desirable properties of explanations [20]. That is, stability does not imply fidelity. For example, an input-agnostic constant explanation is stable but lacks fidelity. Conversely, fidelity does not imply stability—if the underlying model is itself unstable, then any correct high-fidelity explanation of that model must also be unstable. Bansal et al. [52] and Chen et al. [58] identify and explain why input gradients of adversarially trained models are more stable compared to those of standard models. In contrast, our work focuses on identifying and explaining why input gradients of adversarially trained models have more fidelity compared to those of standard models. Furthermore, we also take the first step towards theoretically showing that adversarial robustness can provably improve input gradient fidelity in Appendix E.

B Additional discussion

Translation invariance in BlockMNIST models. Intuitively, CNNs are translation-invariant only if the object of interest is not closer to the boundary than the receptive field of the final layer; In BlockMNIST, the digits are either close to the top boundary or the bottom boundary. Given that the receptive field of Resnets is quite large, translation invariance would not hold in this case. This is further supported by recent work [59], which demonstrates that "CNNs can and will exploit the absolute spatial location by learning filters that respond exclusively to particular absolute locations by exploiting image boundary effects". We observe this phenomenon empirically in our BlockMNIST-Top experiments as well. That is, while models trained on BlockMNIST-Top data (i.e., MNIST digit in top block) attain 100% test accuracy on BlockMNIST-Top images, the accuracy of these models degrades to approximately 55% (i.e., 5% better than random chance) when evaluated on BlockMNIST-Bottom images, wherein the MNIST digit (signal) is placed in the bottom block.

Choice of removal operator in DiffROAR framework. Recall that in DiffROAR, the predictive power of a new model retrained on the unmasked dataset (i.e, data points after removal operation) is used to evaluate the fidelity of post-hoc explanation methods. Note that this approach employs retraining to account for and nullify distribution shifts induced by feature removal operators such as gaussian noise, zeros etc. Since the same removal operation is applied to unmask every image (across classes), the choice of removal operator has no effect on our DiffROAR results in Section 4. To verify this, we evaluated DiffROAR on CIFAR-10 with another removal operator in which pixels are masked/replaced by random gaussian noise (instead of zeros) and observed that the results do not change (i.e., same as in Figure 3).

Counterfactual changes vis-a-vis feature leakage. As evidenced in the BlockMNIST experiments, input gradient attributions of standard models incorporate counterfactual changes in the null block. While this phenomenon seems natural and "intuitive" in hindsight, it can be misleading in the context of feature attributions. For example, consider the typical use case for feature attributions: to highlight regions within the given instance/image that are most relevant for model prediction. Now, in the BlockMNIST setting, if input gradients leak digit-like features into the null block, then the feature attributions in the null block can be easily (mis)interpreted as the non-discriminative null patch being highly relevant for model prediction.

Comparison to results in Kim et al. [33]. Kim et al. [33] use the ROAR framework to conjecture that adversarial training "tilts" input gradients to better align with the data manifold. First, in contrast to Kim et al. [33], we thoroughly establish our DiffROAR results across datasets/architectures/hyper-parameters, revealing a significantly larger gap between the attribution quality of standard and adversarially robust models. Second, motivated by the boundary tilting hypothesis [60], Kim et al. [33] use a two-dimensional synthetic dataset to empirically show that the decision boundary of robust models aligns better with the vector between the two class-conditional means. However, this empirical evidence might be misleading, as Ilyas et al. [61] theoretically demonstrates that "this exact statement is not true beyond two dimensions" (pg. 15). Furthermore, several recent works have also provided concrete evidence to support alternative hypotheses [61, 37, 62, 63] for the existence of adversarial examples that counter the boundary tilting hypothesis that Kim et al. build upon. This discrepancy in these results motivates the need for a multipronged approach, which we adopt to empirically identify the feature leakage hypothesis using BlockMNIST and theoretically verify the hypothesis in Section 6.

Connections between adversarial robustness and data manifold: In the recent past, there have been several results showing unexpected benefits of adversarially trained models beyond adversarial robustness such as visually perceptible input gradients [29] and feature representations that transfer better [64]. One reason for this phenomenon widely considered in the literature [65, 66] is that the input data lies on a low dimensional manifold and unlike standard training, adversarial training encourages the decision boundary to lie on this manifold (i.e. alignment with data manifold). Our experiments and theoretical results on feature leakage suggest that this reasoning is indeed true for both the BlockMNIST and its simplified version presented in Section 6. Furthermore, we believe that the simplified version of BlockMNIST in eq. (2) can be used as a tool to thoroughly investigate both the benefits and potential drawbacks of adversarially trained models.

Why focus on input gradient attributions?. As discussed in Section 1, several feature attributions such as guided backprop [16] and integrated gradients [22] that output visually sharper saliency maps fail basic sanity checks such as model randomization and label randomization [12, 13, 20]. We focus on vanilla input gradient attributions for two key reasons: (i) vanilla input gradients pass both sanity checks mentioned above and (ii) the input gradient operation is the key building block of several feature attribution methods. Our experiments and theoretical analysis are specifically designed to identify and verify feature leakage in input gradient attributions of standard models.

Comparing `ROAR` and `DiffROAR`. The following questions below illustrate key differences between `ROAR` [14] and our work:

- *Does the framework verify assumption (`A`)?* In Hooker et al. [14], the `ROAR` framework essentially computes the top-k predictive power only, which is not sufficient to test assumption (`A`). In our paper, DiffROAR directly compares the top-k and bottom-k predictive power to test whether the given attribution method satisfies assumption (`A`).
- *Are the results in the paper conclusive?* Both, `ROAR` and `DiffROAR`, make a key assumption: models retrained on unmasked datasets learn the same features as the model trained on the original dataset. Although empirically supported [34, 35, 37], this assumption makes it difficult to conclusively test assumption (`A`). Therefore, we empirically (Section 5) and theoretically (Section 6) verify our `DiffROAR` findings in settings wherein ground-truth features are known a priori.
- *Does the work identify why standard input gradients violate (`A`)?* Hooker et al. [14] do not discuss why input gradients lack explanation fidelity. In our paper, we hypothesize feature leakage as the key reason for ineffectiveness of input gradients, and validate it with empirical as well as theoretical analysis on `BlockMNIST`-based data

Limitations of `ROAR` and `DiffROAR`. The major limitation of `ROAR` and `DiffROAR` is the key assumption that models retrained on unmasked datasets learn the same features as the model trained on the original dataset. In the absence of ground-truth features, this assumption is empirically supported by findings that suggest that different runs of models sharing the same architecture learn similar features [34, 35, 37]. Another limitation is that `ROAR`-based frameworks are not useful in the following setting. Consider a redundant dataset where features are either all negative (in which case label $y = 0$) or all positive (in which case label $y = 1$). In such cases, no feature is more or less informative than any other, so no information can be gained by ranking or removing input coordinates/features.

C Experiments on real-world datasets using DiffROAR

In this section, we first provide additional details about datasets, training, and performance of trained models vis-a-vis generalization and robustness. We also present top-k and bottom-k predictive power of input gradient unmasking schemes obtained via standard and robust models. Next, we show that our results on image classification benchmarks are robust to CNN architectures and SGD hyperparameters used during retraining. Then, we use DiffROAR to show that our results hold with input *loss* gradients, but *signed* input logit gradients do not satisfy assumption (A) for standard *or* robust models. Finally, we discuss DiffROAR results obtained without retraining and provide additional example images that are masked using input gradients of standard & robust models.

C.1 Additional details about DiffROAR experiments and trained models

We first provide additional details about standard and adversarial training, and describe the performance of trained models vis-a-vis generalization and robustness to ℓ_2 & ℓ_∞ perturbations.

Recall that we use DiffROAR to analyze input gradients of standard and adversarially robust two-hidden-layer MLPs on SVHN & Fashion MNIST, Resnet18 on ImageNet-10, and Resnet50 on CIFAR-10 in Figure 3. In these experiments, we train models using stochastic gradient descent (SGD), with momentum 0.9, batch size 256, ℓ_2 regularization 0.0005 and initial learning rate 0.1 that decays by a factor of 0.75 every 20 epochs; We obtain ℓ_2 and ℓ_∞ ϵ-robust models with perturbation budget ϵ using PGD adversarial training [15]. In PGD adversarial training, we use learning rate $\epsilon/4$, 8 steps of PGD and no random initialization in order to compute ϵ-norm ℓ_2 and ℓ_∞ perturbations. In both cases, we use standard data augmentation and train models for at most 500 epochs, stopping early if cross-entropy (standard or adversarial) loss on training data goes below 0.001. Unless mentioned otherwise, we set the depth and width of MLPs trained on real datasets to be 2 and 2× the input dimension respectively.

Figure 6 depicts standard test accuracy (i.e., when perturbation budget $\epsilon = 0$) and ϵ-robust test accuracy (for multiple values of ϵ) of standard as well as ℓ_2 and ℓ_∞ robust models trained on SVHN, Fashion MNIST, CIFAR-10 and ImageNet-10. Note that to estimate ϵ-robust test accuracy, we use PGD-based adversarial *test* examples, computed using 2× the number of PGD steps used during training. As expected, we observe that (i) compared to standard models, adversarially trained MLPs and CNNs attain significantly better robust test accuracy, (ii) models trained with larger perturbation budget are more robust to larger-norm adversarial perturbations at test time, and (iii) standard test accuracies (when $\epsilon = 0$) of adversarially trained models are worse than those of standard models.

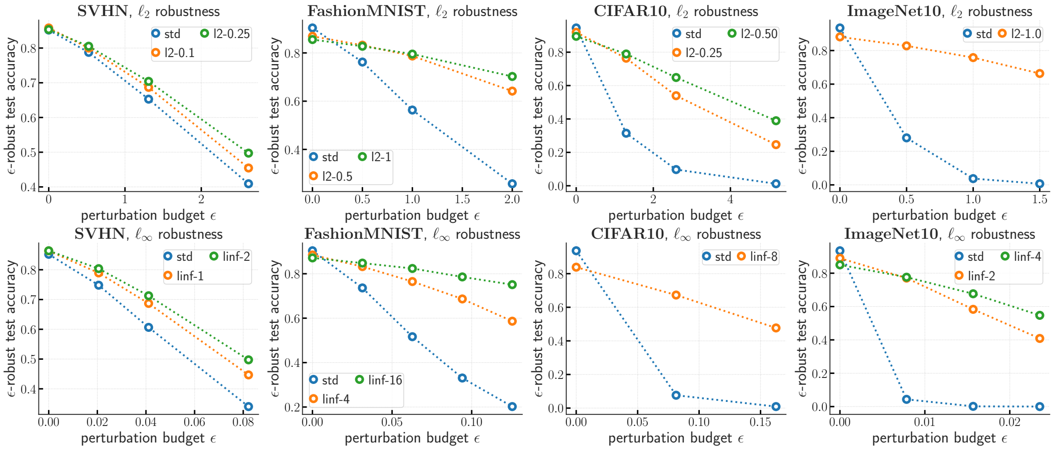

Figure 6: **Standard and ϵ-robust test accuracies** of MLPs trained on SVHN and Fashion MNIST, Resnet50 trained on CIFAR-10, and Resnet18 trained on ImageNet10. Details in Appendix C.1.

C.2 Top-k and bottom-k predictive power of input gradient attributions

Now, we describe the top-k and bottom-k predictive power curves for unmasking schemes of input gradients of standard and robust models. Recall that top-k predictive power simply estimates the test accuracy of models that are retrained on datasets wherein only coordinates with top-k (%) of the coordinates are unmasked in every image. The top and bottom rows in Figure 7 show how top-k and bottom-k predictive power of input gradient attributions of standard and robust models vary with unmasking fraction k respectively. The subplots in Figure 7 show that (i) decreasing the unmasking fraction k decreases top-k and bottom-k predictive power, and (ii) models retrained on attribution-masked datasets attain non-trivial unmasked test dataset accuracy even when a significant fraction of coordinates with the top-most and bottom-most attributions are masked.

As described in Section 3, for a given attribution scheme and unmasking fraction or level k, `DiffROAR` (see equation (1)) is positive when the top-k predictive power is greater than the bottom-k predictive power. The subplots in the first column indicate that standard models trained on Fashion MNIST do not satisfy assumption (`A`) because the top-k and bottom-k unmasking schemes are *equally* ineffective at masking discriminative features. Conversely, the difference between the top-k and bottom-k predictive power of input gradient attributions of robust models is significant. For example, in the second column, for the SVHN model adversarially trained with ℓ_∞ perturbations and budget $\epsilon = {}^2\!/_{255}$ (purple line), top-k predictive power is roughly 40% more than the bottom-k predictive power when $k = 5\%$. Furthermore, as shown in the third and fourth columns, the top-k and bottom-k curves of standard CNNs trained on CIFAR-10 and ImageNet-10 are "inverted", thereby explaining why `DiffROAR` is *negative* when unmasking fraction is roughly less than 40%.

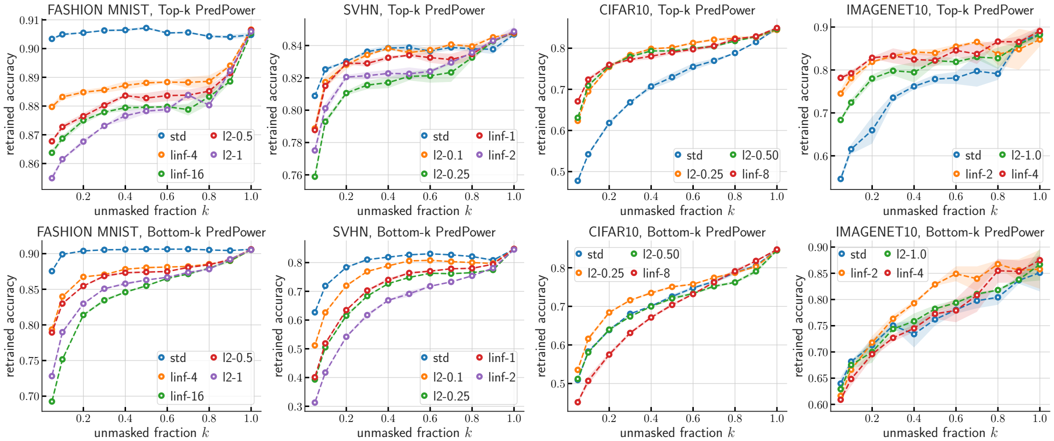

Figure 7: **Predictive power of top-k and bottom-k input gradient unmasking schemes** vs. unmasking fraction, or level, k for standard and adversarially robust models trained on 4 image classification benchmarks. Please see Appendix C.2 for details.

C.3 Effect of model architecture on `DiffROAR` results

Recall that in Section 4, we used the `DiffROAR` metric to evaluate whether input gradient attributions of models trained on real-world datasets satisfy or violate assumption (`A`). For CNNs, we evaluated input gradient attributions of standard Resnet50 and Resnet18 models trained on CIFAR-10 and Imagenet-10 respectively. In this section, we show that our empirical findings based on these architectures extend to three other commonly-used and well-known CNN architectures: Densenet121, InceptionV3, and VGG11.

As shown in Figure 8, the `DiffROAR` results support key empirical findings made using input gradients of Resnet models in Section 4: (i) standard models perform poorly, often no better or even worse than the random attribution baseline, and (ii) `DiffROAR` curves of adversarially robust models are positive and significantly better than that of the standard model. For example, for Densenet121, InceptionV3, and VGG11, when unmasking fraction $k = 20\%$, standard training yields input gradient attributions that attain `DiffROAR` scores roughly -5%, 2% and 1% respectively, whereas ℓ_∞ adversarial training with budget $\epsilon = {}^6\!/_{255}$ results in input gradients with `DiffROAR` metric roughly 15%.

C.4 Effect of SGD Hyperparameters on `DiffROAR` results

In this section, we show that `DiffROAR` results for input gradient attribu of standard and robust models are not sensitive to the choice of SGD hyperparameters used during retraining. In particular, we show that `DiffROAR` curves on CIFAR-10 are not sensitive to the learning rate, weight decay, or the momentum used in SGD to train models on top-k or bottom-k attribution-masked datasets. The

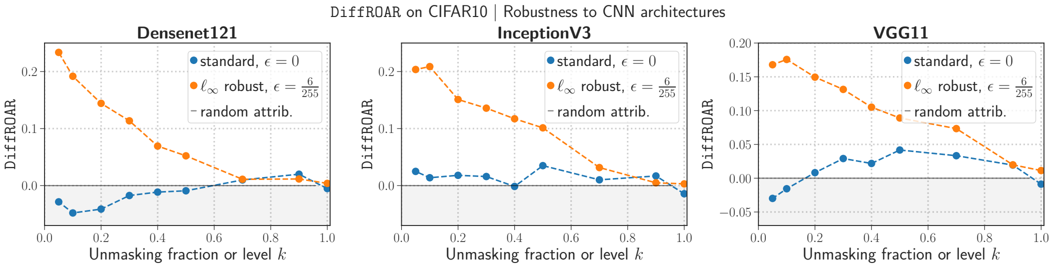

Figure 8: DiffROAR **results on input gradients of additional CNN architectures**. DiffROAR curves for three well-known NN architectures—Densenet121, InceptionV3, and VGG11—indicate that empirical findings vis-a-vis input gradients of standard and robust models (Section 4) are robust to choice of CNN architecture. Please see Appendix C.3 for details.

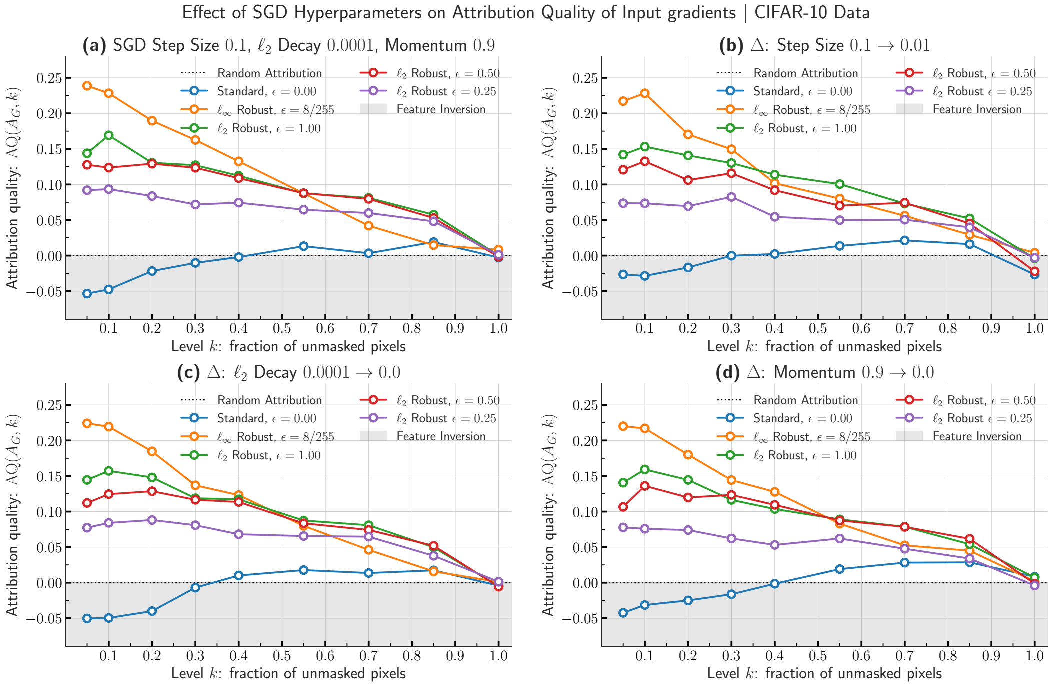

Figure 9: DiffROAR **robust to SGD hyperparameters in retraining**. DiffROAR curves for input gradients of standard and robust models trained on CIFAR-10 data show that our empirical findings presented in Section 4 are robust to SGD hyperparameters that are used in retraining. Specifically, we show that our findings vis-a-vis DiffROAR are not sensitive to changes in SGD hyperparameters such as learning rate, momentum, and weight decay that are used to retrain models on unmasked CIFAR-10 data. For example, the subplots above show that across multiple SGD hyperparameter values, when the fraction of unmasked pixels $k <$ 30-40%, standard models violate (A) whereas robust models satisfy (A). See Appendix C.4 for details.

four subplots in Figure 9 collectively show that decreasing learning rate from 0.1 to 0.01, weight decay from 0.0001 to 0, and momentum from 0.9 to 0 does not alter our findings: (i) input gradient attributions of standard models do not satisfy (A) when unmasking fraction k is roughly less than 30-40%; (ii) models that are robust to ℓ_2 and $\ell\infty$ perturbations consistently satisfy (A); (iii) increasing perturbation budget ϵ during PGD adversarial training increases DiffROAR metric for most values of unmasking fraction k. To summarize, our results based on the DiffROAR evaluation framework are robust to SGD hyperparameters used to retrain models on top-k and bottom-k unmasked datasets.

C.5 Evaluating input *loss* gradient attributions using DiffROAR

Recall that our experiments in Section 4 evaluate whether input gradients taken w.r.t. the logit of the predicted label satisfy or violate assumption (A) on image classification benchmarks. In this section, we show that our empirical findings generalize to input *loss* gradients—input gradients w.r.t loss

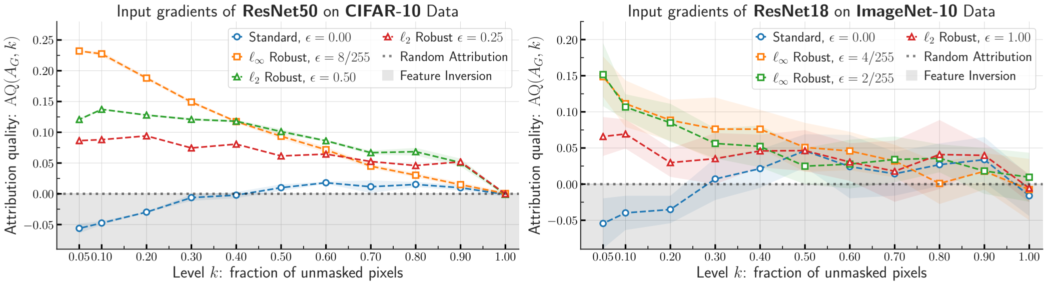

Figure 10: `DiffROAR` **results for input loss gradient attributions**. `DiffROAR` plots for input *loss* gradient attributions of standard and adversarially Resnet50 on CIFAR-10 and Resnet18 on ImageNet-10. In both subplots, standard models violate (`A`) when the fraction of unmasked pixels $k < 30\%$. That is, input coordinates that have the largest gradient magnitude are not as important performance-wise as the coordinates with smallest gradient magnitude. Conversely, $\{\ell_2, \ell_\infty\}$-adversarially trained models satisfy (`A`), as the `DiffROAR` metric is positive for all $k < 100\%$. Similar to our results with input logit gradients, we observe that increasing the perturbation budget ϵ during adversarial training amplifies the magnitude of `DiffROAR` for every k across all four image classification benchmarks.

(e.g., cross-entropy)—of standard and robust models evaluated on image classification benchmarks. Specifically, we apply `DiffROAR` to input *loss* gradients of standard and robust ResNet models trained on CIFAR-10 and ImageNet-10.

Figure 10 illustrates `DiffROAR` curves for input *loss* gradient attributions on CIFAR-10 and ImageNet-10 data. In both cases, we observe that (i) input loss gradient attributions of robust models, unlike those of standard models, satisfy (`A`) and (ii) PGD adversarial training with larger perturbation budget ϵ increases the `DiffROAR` metric in a consistent manner. Recall that the magnitude in `DiffROAR` quantifies the extent to which the attribution order separates discriminative and task-relevant features from features that are unimportant for model prediction; see Section 3 for more information about `DiffROAR`.

C.6 Evaluating *signed* input gradient attributions using `DiffROAR`

In addition to input loss gradient magnitude attributions and input logit gradient magnitude attributions, our results vis-a-vis `DiffROAR` evaluation on image classification benchmarks extend to *signed* input logit gradients as well. In signed input gradient attributions, input coordinates are ranked based on $\mathrm{sgn} x_i \cdot g_i$ where $\mathrm{sgn}(x_i)$ is the sign of input coordinate x_i and g_i is the signed input gradient value for input coordinate x_i.

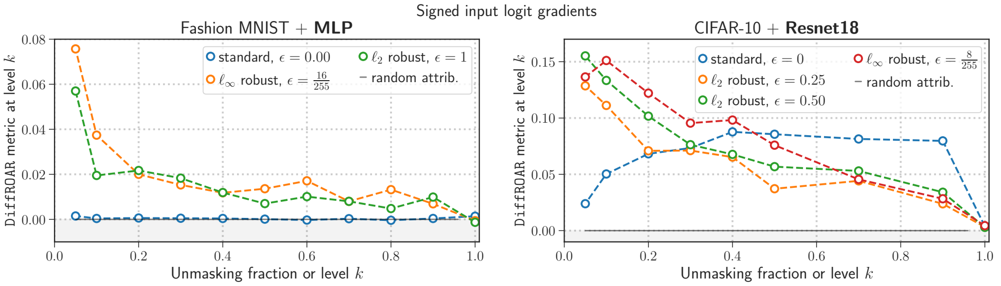

Figure 11: `DiffROAR` **results for signed input logit gradients**. `DiffROAR` results for attributions based on signed input gradients of standard and robust MLPs & CNNs trained on Fashion MNIST & CIFAR-10. See Appendix C.6 for details.

Figure 11 shows `DiffROAR` curves for attributions based on *signed* input gradients taken with respect to the logit of the predicted label. The left and right subplot evaluate `DiffROAR` for standard and robust (i) MLP trained on Fashion MNIST and (ii) Resnet18 models trained on CIFAR-10. Consistent with our findings in Section 4, while standard MLPs trained on Fashion MNIST fare no better than random attributions, signed input gradients of robust MLPs attain positive `DiffROAR` scores for all $k < 100\%$ and perform considerably better than gradients of standard MLPs. Similarly, based on the `DiffROAR` metric, when $k < 50\%$, while signed input gradients of standard Resnet18 models perform better than absolute logit and loss gradients, signed input gradients of robust Resnet18 models continue to fare better than standard models.

C.7 The role of retraining in DiffROAR evaluation

Figure 12 shows the results on DiffROAR without retraining on the masked datasets. As we can see from the figures, the trends are not consistent across model architectures and datasets, possibly due to varying levels of distribution shift. For this reason, we employ DiffROAR with retraining as described in Section 3.

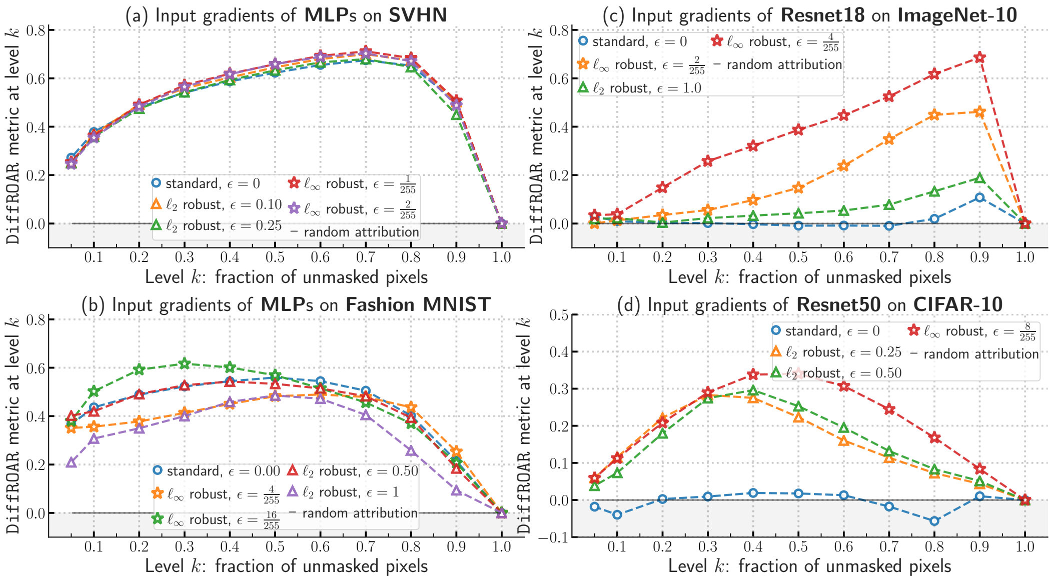

Figure 12: DiffROAR **results without retraining**. While we observe that standard models violate (A) while adversarially trained models satisfy (A) for the Resnet models, we see that both standard and adversarially trained models satisfy (A) for MLP models, showing that this evaluation methodology does not yield consistent results across model architectures/datasets. Further, the DiffROAR metric may be unrealiable for small unmasking fractions since this incurs heavy distribution shift. Consequently, we employ DiffROAR after retraining on the new unmasked data.

C.8 Imagenet-10 images unmasked using input gradients attributions of Resnet18 models

Recall that in Section 4, we showed that unlike input gradients of standard models, robust models consistently satisfy assumption (A). That is, input gradients of robust models highlight discriminative features, whereas input gradients of standard models tend to highlight non-discriminative features and suppress discriminative task-relevant features. In this section, we qualitatively substantiate these findings by visualizing ImageNet-10 images that are unmasked using top-k and bottom-k input gradient attributions of standard and robust Resnet18 models. *Please note that the following visual assessments are only meant to* qualitatively *support findings made in Section 4 using the evaluation framework described in Section 3.* As discussed in Section 3, if input gradients attain high-magnitude DiffROAR score, images unmasked using top-k attributions should highlight discriminative features, whereas images unmasked using bottom-k should highlight non-discriminative features.

We make two observations using Figure 13 that qualitatively support our empirical findings in Section 4. First, we observe that images unmasked using top-k gradient attributions of robust models tend to highlight salient aspects of images (e.g., shape of fruit or face of monkey in Figure 13), whereas bottom-k attributions often mask salient aspects of images either completely or partially. Second, images unmasked using top-k and bottom-k attributions using input gradients of standard models exhibit visual commonalities, supporting the fact that for standard models, DiffROAR is close to 0 for multiple values of k.

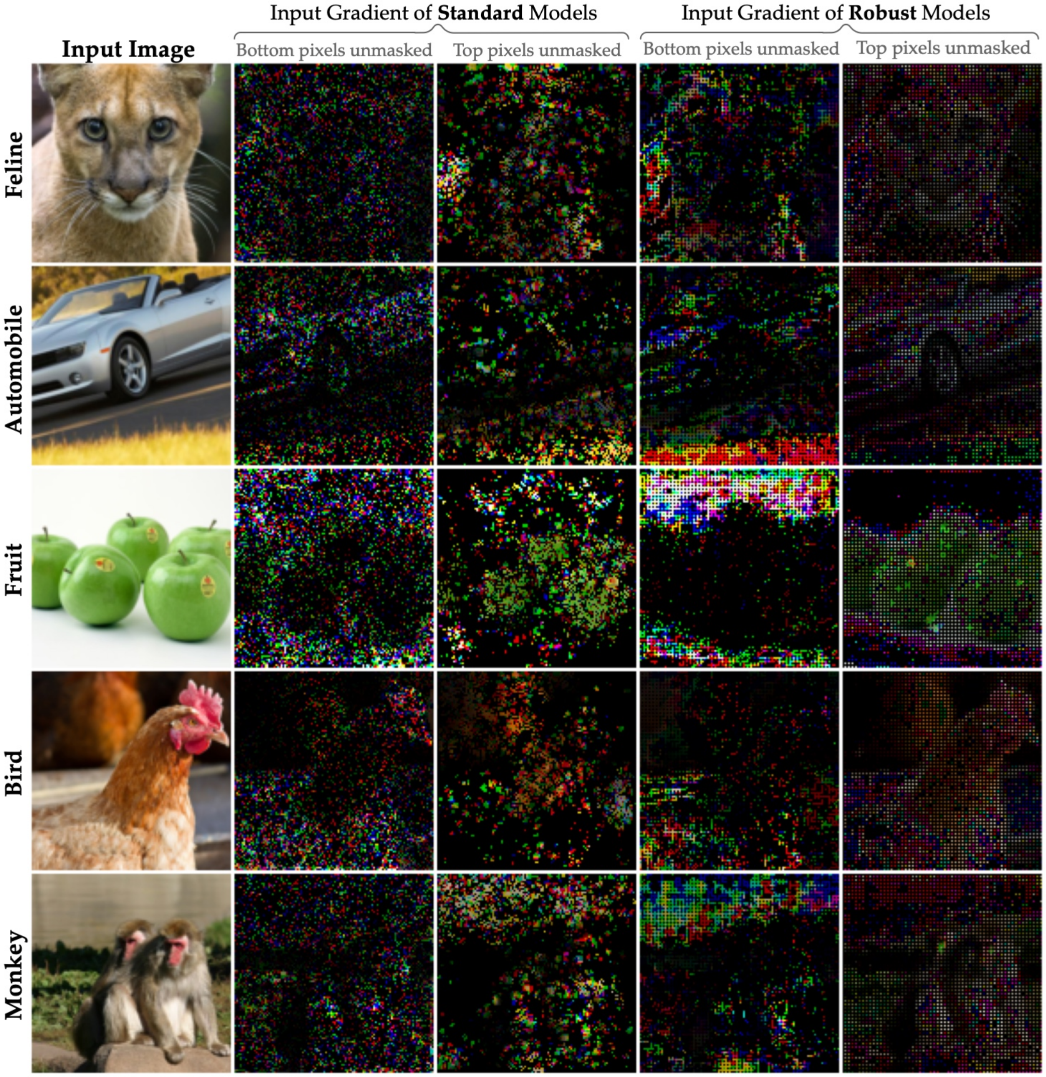

Figure 13: **ImageNet10 images unmasked using input gradient attributions**. Visualizing ImageNet-10 images that are unmasked using unmasking fraction, or level, $k = 15\%$ using input gradient attributions of standard and ℓ_∞-robust Resnet18 models. Top-k unmasked images (i.e., images in which *only* "top" gradient attributions are unmasked) and bottom-k unmasked images, attained via input gradients of standard models, share visual commonalities, suggestive of poor attribution quality. Unlike bottom-k unmasked images, images unmasked using top-k attributions of robust models' input gradients highlight salient aspects of images. See Appendix C.8 for details.

D Additional experiments on feature leakage and `BlockMNIST` data

In this section, we first provide additional evidence that supports the feature leakage hypothesis in the setting used in Section 5: `BlockMNIST` data with `MNIST` digits 0 and 1 corresponding to the signal block in class 0 and class 1 respectively. Then, we show that our results vis-a-vis feature leakage and `BlockMNIST` are robust to the choice of `MNIST` digits used in the signal block as well as the number of classes in the `BlockMNIST` classification task. Finally, we end with a brief description of experiments that we conducted in order to test another hypothesized cause to understand why input gradients of standard models tend to violate (`A`).

D.1 Additional analysis to demonstrate feature leakage in `BlockMNIST` data

In this section, we provide (i) additional examples of `BlockMNIST` images and inputs gradients of standard and robust models, (ii) additional examples of `BlockMNIST-Top` images and input gradients, and (iii) describe a *proxy* metric to measure feature leakage in `BlockMNIST`-based data.

Figure 14 shows 40 `BlockMNIST` images in the first row and their corresponding input gradients for standard and robust MLPs and Resnet18 models in the subsequent rows. We observe that input gradient attributions of standard MLP and Resnet18 models consistently highlight the signal block *as well as* the non-discriminative null block for all images. On the other hand, input gradient attributions of ℓ_2 robust MLP and Resnet18 models exclusively highlight `MNIST` digits in the signal block and clearly suppress the square patch in the null block. These results further substantiate our results in Figure 3 by showing that unlike standard models, adversarially robust models satisfy (`A`) on `BlockMNIST` data. Figure 17 provides 20 `BlockMNIST-Top` images in the first row and the corresponding input gradients of standard MLP and Resnet models in the subsequent rows. As shown in Figure 16, in this setting, in contrast to results on `BlockMNIST`, input gradients of *standard* Resnet18 and MLP models trained on `BlockMNIST-Top` satisfy assumption (`A`).

We further substantiate these findings using a *proxy* metric to quantitatively measure feature leakage in `BlockMNIST`-based datasets. As discussed in Section 5, in the `BlockMNIST` setting, we can restate assumption (`A`) as follows: *Do input gradient attributions highlight the signal block over the null block?* We measure the extent to which input gradients of a given trained model satisfies assumption (`A`) by evaluating the fraction of top-k attributions that are placed in the null block. In Figure 16, we show that the fraction of top-k attributions in the null block, when averaged over all images in the test dataset, is significantly greater for standard MLPs & CNNs than for robust MLP & CNNs. In Figure 17, we show that input gradient attributions of standard models trained on `BlockMNIST-Top` place significantly fewer attributions in the null block, compared to attributions of standard models trained on `BlockMNIST`. In both cases, the proxy metric further validates our findings vis-a-vis input gradients of standard & robust models and feature leakage.

D.2 Effect of choice and number of classes in `BlockMNIST` data

In this section, we show that our analysis on `BlockMNIST`-based datasets in Section 5 is robust to the choice and number of classes in `BlockMNIST` data. In particular, we reproduce our empirical findings vis-a-vis feature leakage and input gradient attributions of standard vs. robust models on three additional `BlockMNIST`-based tasks. In Figure 18 and Figure 19, we evaluate input gradients of standard and robust models trained on `BlockMNIST` and `BlockMNIST-Top` data, wherein the `MNIST` digits in class 0 and class 1 correspond to digits 2 and 4 (in the signal block) respectively. Similarly, in Figure 20 and Figure 21, we reproduce our empirical findings from Section 5 on `BlockMNIST` and `BlockMNIST-Top` data in which the `MNIST` digits in class 0 and class 1 correspond to digits 3 and 7 (in the signal block) respectively. In Figure 22 and Figure 23, we show that (i) input gradients of standard models violate assumption (`A`) due to feature leakage and (ii) adversarial training mitigates feature leakage on 10-class `BlockMNIST` and `BlockMNIST-Top` data, wherein each class $i\{0, \ldots, 9\}$ corresponds to `MNIST` digit i in the signal block.

D.3 Does randomness in initialization explain why input gradients violate (`A`)?

In this section, *we investigate whether the poor quality of input gradients in standard models is due to randomness retained from the initialization.* Figure 24 shows scatter plots of input gradient values over all pixels in all images before (x-axis) and after (y-axis) standard training on four image classification benchmarks. The results indicate that (i) the scale of gradients after training is at least an order of magnitude larger than those before training and (ii) the gradient values before and after training are uncorrelated. Together, these results suggest that random initialization does not have much of a role in determining the input gradients after training.

D.4 Do other feature attribution methods exhibit feature leakage?

In this section, we evaluate feature leakage in five feature attribution methods: Integrated Gradients [22], Layer-wise Relevance Propagation (LRP) [24], Guided Backprop [16], Smoothgrad [2] (with standard deviation $\sigma \in \{0.1, 0.3, 0.5\}$), and Occlusion [67] (with patch size $\rho \in \{5, 10\}$). First, we evaluate the aforementioned feature attribution methods on standard models trained on `BlockMNIST` data. As shown in Figure 25 and Figure 26, in addition to vanilla input gradients, all five feature attribution methods evaluated on standard MLPs and Resnet18 models highlight the `MNIST` signal block as well as the null block. Conversely, Figure 27 and Figure 28 show that when standard MLPs and Resnet18 models are trained on `BlockMNIST-Top` data, all feature attribution methods exclusively highlight the `MNIST` signal block. These results collectively indicate that similar to vanilla input gradient attributions, multiple feature attribution methods exhibit feature leakage. Furthermore, consistent with our findings on adversarial robustness vis-a-vis feature leakage, Figure 29 and Figure 30 show that feature attribution method evaluated on adversarially robust MLPs and Resnet18 model do not exhibit feature leakage on `BlockMNIST` data.

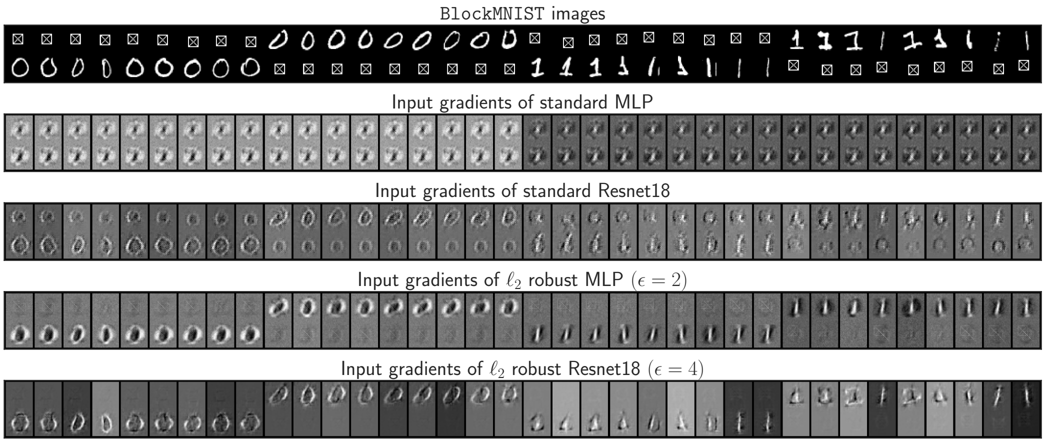
BlockMNIST images
Input gradients of standard MLP
Input gradients of standard Resnet18
Input gradients of ℓ_2 robust MLP $(\epsilon = 2)$
Input gradients of ℓ_2 robust Resnet18 $(\epsilon = 4)$

Figure 14: **BlockMNIST 0 vs. 1**. 40 BlockMNIST (MNIST 0 vs. 1) images and their corresponding input gradients. Recall that every image consists of a *signal* and *null* block, each randomly placed at the *top* or *bottom*. The *signal* block, containing the MNIST digit 0 or 1, determines the image class, 0 or 1. The *null* block, containing the square patch, does not encode any information of the image class. The second, third, and fourth rows show input gradients of standard Resnet18, standard MLP, ℓ_2 robust Resnet18 $(\epsilon = 2)$ and ℓ_2 robust MLP $(\epsilon = 4)$ respectively. The plots clearly show that input gradients of standard BlockMNIST models incorrectly highlight *the non-discriminative null block* as well, thereby violating (A). In contrast, input gradients of robust models highlight the signal block, suppress the null block, and satisfy (A). See Appendix D.1 for details.

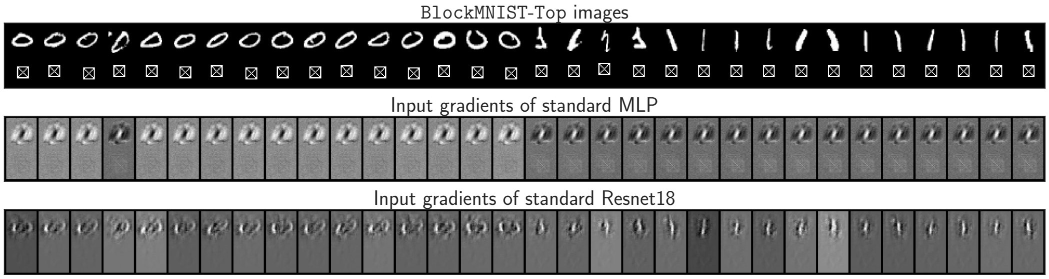
BlockMNIST-Top images
Input gradients of standard MLP
Input gradients of standard Resnet18

Figure 15: **BlockMNIST-Top 0 vs. 1**. 20 BlockMNIST-Top (MNIST 0 vs. 1) images and input gradients of standard MLP and Resnet18 models. As shown in the first row, the signal & null blocks are fixed at the top & bottom respectively in BlockMNIST-Top images. In contrast to results on BlockMNIST in fig. 1, input gradients of standard models trained on BlockMNIST-Top highlight the signal block, suppress the null block, and satisfy (A). Please see Appendix D.1 for details.

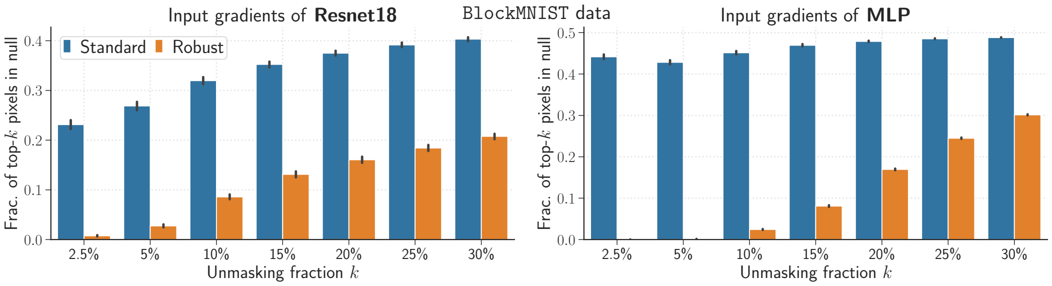

Figure 16: **Proxy metric to compare input gradients of standard and robust models trained on BlockMNIST (0 vs. 1) data**. The proxy metric measures the fraction of top-k attributions that are placed in the null block of images in the test dataset. The left and right subplots evaluate this metric on input gradient attributions of standard and robust Resnet18 models and MLPs respectively. Compared to input gradients of standard models, adversarially trained models place significantly fewer top-k attributions in the null block for multiple values of unmasking fraction k. Details in Appendix D.1.

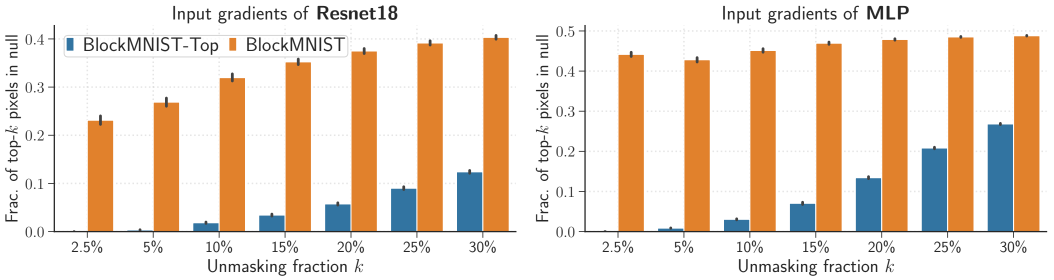

Figure 17: **Proxy metric to compare input gradients of standard models trained on** `BlockMNIST` **and** `BlockMNIST-Top` **(0 vs. 1) data**. The proxy metric measures the fraction of top-k attributions that are placed in the null block of images. The left and right subplots evaluate this metric on input gradient attributions of standard Resnet18 models and MLPs trained on `BlockMNIST` and `BlockMNIST-Top` data respectively. Compared to input gradients of models trained on `BlockMNIST`, standard models trained on `BlockMNIST-Top` place significantly fewer top-k attributions in the null block for multiple values of unmasking fraction k. Details in Appendix D.1.

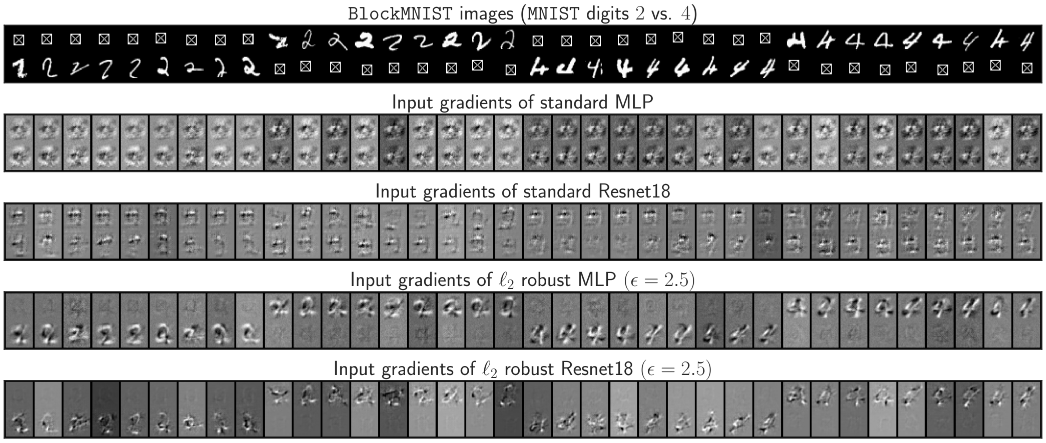
BlockMNIST images (MNIST digits 2 vs. 4)
Input gradients of standard MLP
Input gradients of standard Resnet18
Input gradients of ℓ_2 robust MLP ($\epsilon = 2.5$)
Input gradients of ℓ_2 robust Resnet18 ($\epsilon = 2.5$)

Figure 18: **BlockMNIST 2 vs. 4**. 40 `BlockMNIST` (`MNIST` 2 vs. 4) images and their corresponding input gradients. The *signal* block, containing the `MNIST` digit 2 or 4, determines the image class, 0 or 1. The second, third, and fourth rows show input gradients of standard Resnet18, standard MLP, ℓ_2 robust Resnet18 ($\epsilon = 2.5$) and ℓ_2 robust MLP ($\epsilon = 2.5$) respectively. The plots clearly show that input gradients of standard `BlockMNIST` models incorrectly highlight *the non-discriminative null block* as well, thereby violating (`A`). In contrast, input gradients of robust models highlight the signal block, suppress the null block, and satisfy (`A`). See Appendix D.1 for details.

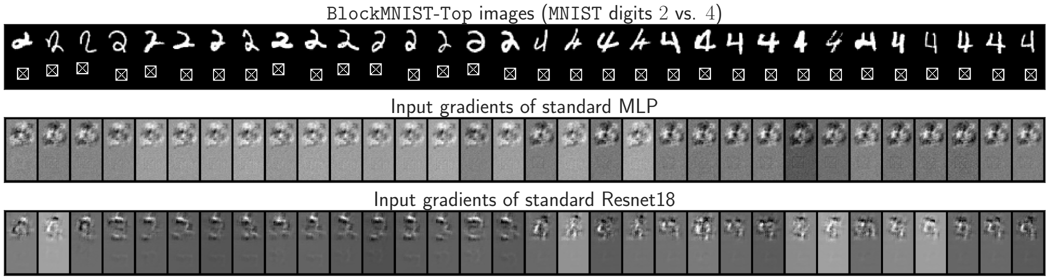
BlockMNIST-Top images (MNIST digits 2 vs. 4)
Input gradients of standard MLP
Input gradients of standard Resnet18

Figure 19: **BlockMNIST-Top 2 vs. 4**. 20 `BlockMNIST-Top` (`MNIST` 2 vs. 4) images and corresponding input gradients of standard MLP and Resnet18 models. As shown in the first row, the signal & null blocks are fixed at the top & bottom respectively in `BlockMNIST-Top` images. In contrast to results on `BlockMNIST` in fig. 1, input gradients of standard models trained on `BlockMNIST-Top` highlight the signal block, suppress the null block, and satisfy (`A`). Please see Appendix D.1 for details.

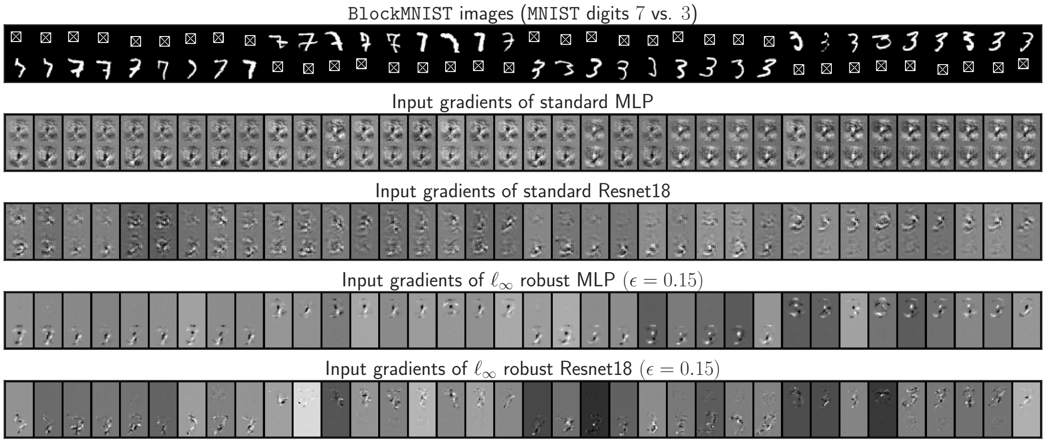
BlockMNIST images (MNIST digits 7 vs. 3)
Input gradients of standard MLP
Input gradients of standard Resnet18
Input gradients of ℓ_∞ robust MLP $(\epsilon = 0.15)$
Input gradients of ℓ_∞ robust Resnet18 $(\epsilon = 0.15)$

Figure 20: **BlockMNIST 3 vs. 7**. 40 BlockMNIST (MNIST 3 vs. 7) images and their corresponding input gradients. The *signal* block, containing the MNIST digit 3 or 7, determines the image class, 0 or 1. The second, third, and fourth rows show input gradients of standard Resnet18, standard MLP, ℓ_∞ robust Resnet18 $(\epsilon = 0.15)$ and ℓ_∞ robust MLP $(\epsilon = 0.15)$ respectively. The plots clearly show that input gradients of standard BlockMNIST models incorrectly highlight *the non-discriminative null block* as well, thereby violating (A). In contrast, input gradients of robust models highlight the signal block, suppress the null block, and satisfy (A). See Appendix D.1 for details.

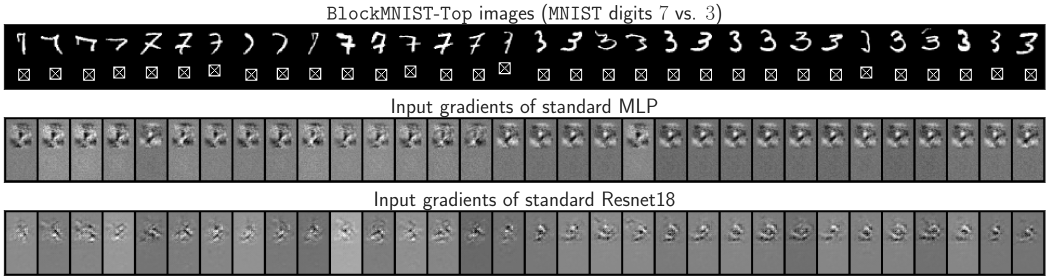
BlockMNIST-Top images (MNIST digits 7 vs. 3)
Input gradients of standard MLP
Input gradients of standard Resnet18

Figure 21: **BlockMNIST-Top 3 vs. 7**. 20 BlockMNIST-Top (MNIST 3 vs. 7) images and input gradients of standard MLP and Resnet18 models. As shown in the first row, the signal & null blocks are fixed at the top & bottom respectively in BlockMNIST-Top images. In contrast to results on BlockMNIST in fig. 1, input gradients of standard models trained on BlockMNIST-Top highlight the signal block, suppress the null block, and satisfy (A). Please see Appendix D.1 for details.

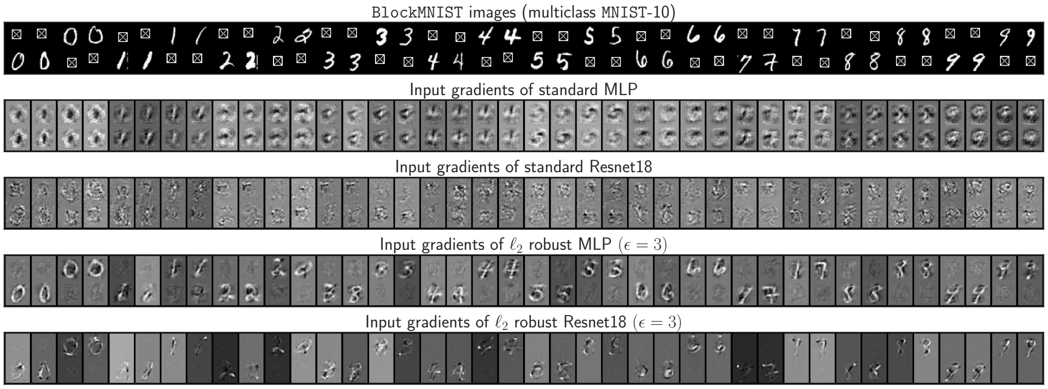
BlockMNIST images (multiclass MNIST-10)
Input gradients of standard MLP
Input gradients of standard Resnet18
Input gradients of ℓ_2 robust MLP ($\epsilon = 3$)
Input gradients of ℓ_2 robust Resnet18 ($\epsilon = 3$)

Figure 22: **Multiclass BlockMNIST**. 40 `BlockMNIST` (all `MNIST` classes) images and their corresponding input gradients. dataset. In this setting, the *signal* block, containing an `MNIST` digit sampled from a class chosen uniformly at random, determines the image class $y \in \{0, \dots, 9\}$. The second, third, and fourth rows show input gradients of standard Resnet18, standard MLP, ℓ_2 robust Resnet18 ($\epsilon = 3$) and ℓ_2 robust MLP ($\epsilon = 3$) respectively. The plots clearly show that input gradients of standard `BlockMNIST` models incorrectly highlight *the non-discriminative null block* as well, thereby violating (`A`). In contrast, input gradients of robust models highlight the signal block, suppress the null block, and satisfy (`A`). See Appendix D.1 for details.

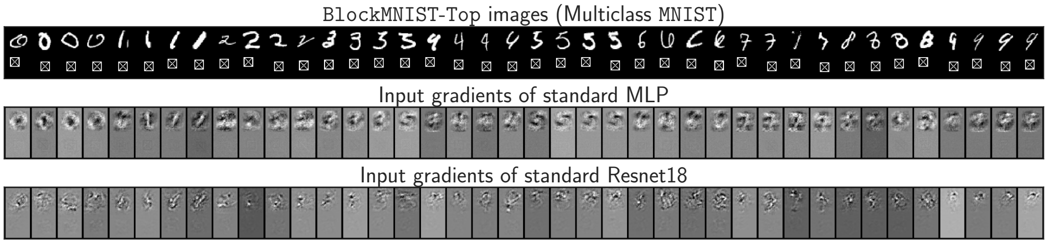
BlockMNIST-Top images (Multiclass MNIST)
Input gradients of standard MLP
Input gradients of standard Resnet18

Figure 23: **Multiclass BlockMNIST-Top**. 40 `BlockMNIST-Top` (all `MNIST` classes) images and corresponding input gradients of standard MLP and Resnet18 models. As shown in the first row, the signal & null blocks are fixed at the top & bottom respectively in `BlockMNIST-Top` images. In contrast to results on `BlockMNIST` in fig. 1, input gradients of standard models trained on `BlockMNIST-Top` highlight the signal block, suppress the null block, and satisfy (`A`). Please see Appendix D.1 for details.

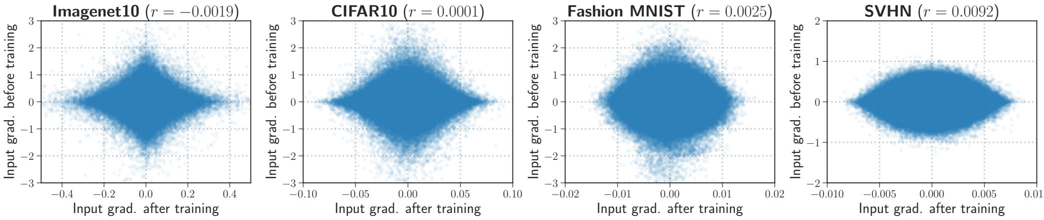

Figure 24: **Does random initialization affect input gradients after training?** The scatter plots above show the gradient values at the beginning of training and after end of training on y-axis and x-axis respectively. We can see that the scale of gradients is much larger at the end of training compared to that at the beginning of training and both of them are uncorrelated. This suggests that the poor quality of input gradients of standard trained models is a result of the training process, and not because of random initialization.

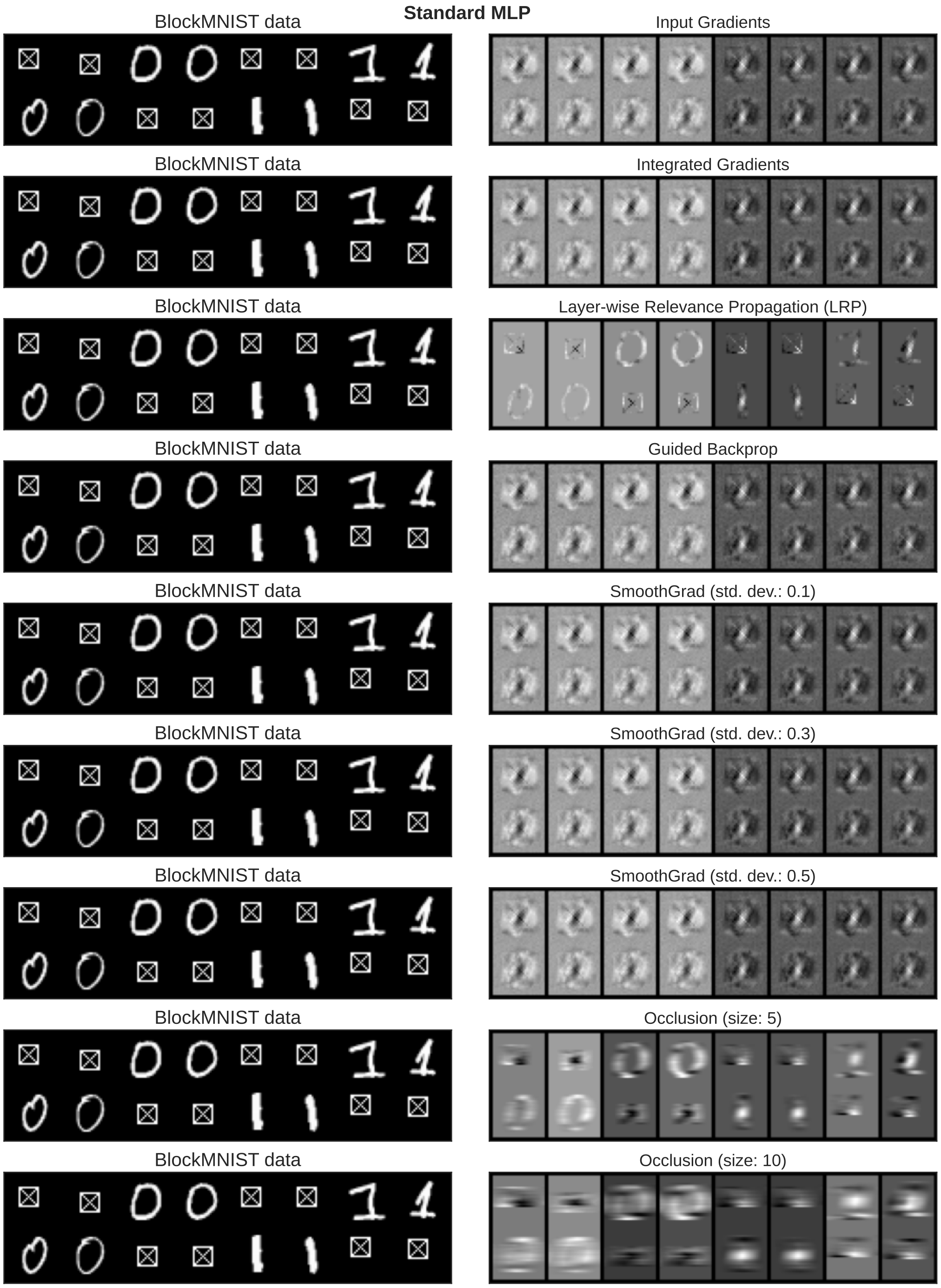

Figure 25: Multiple instance-specific feature attribution methods evaluated using a standard two-layer MLP trained on `BlockMNIST` data. All feature attribution methods exhibit feature leakage, as the attributions highlight the non-predictive null block in addition to the `MNIST` signal block.

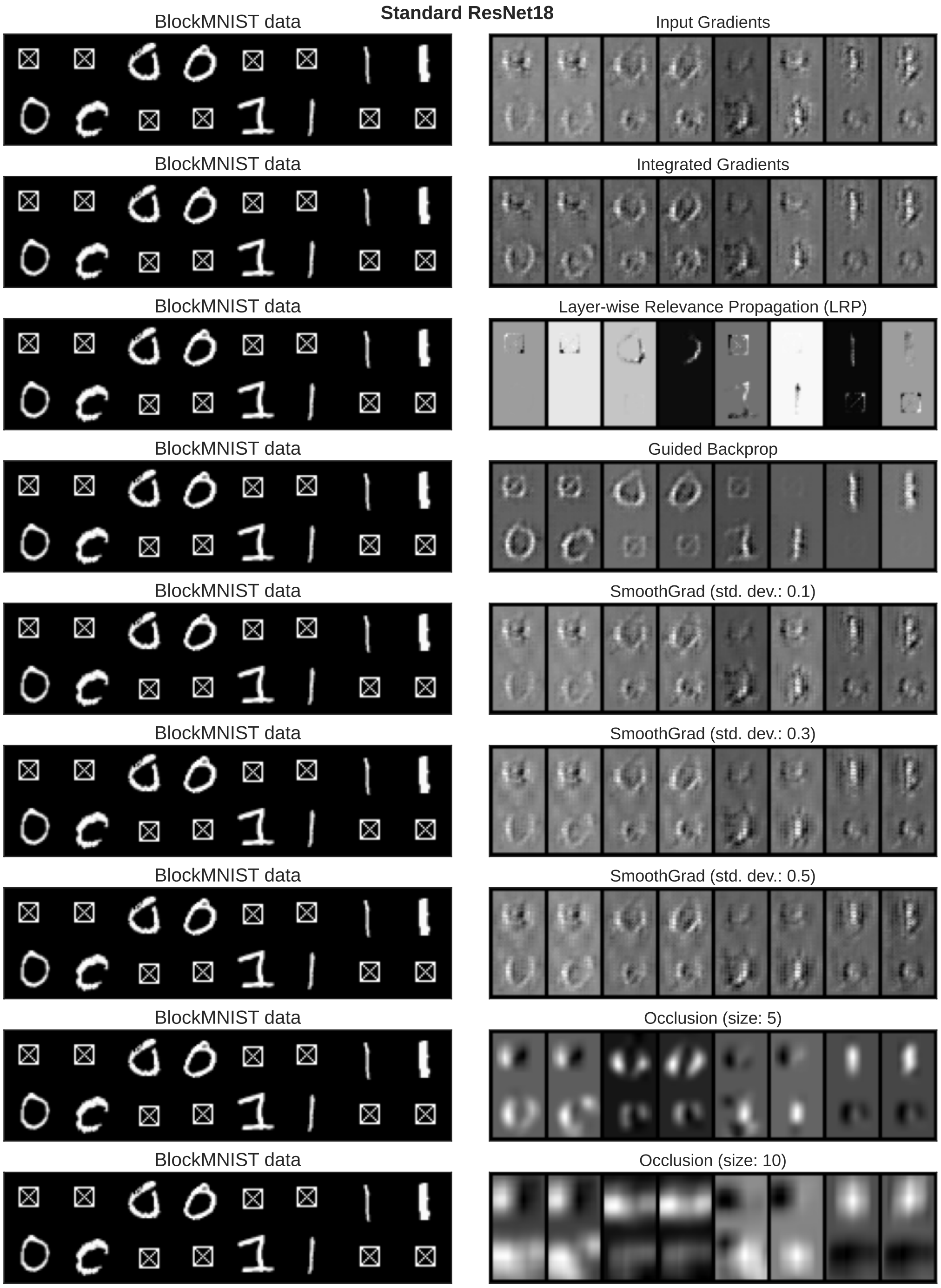

Figure 26: Multiple instance-specific feature attribution methods evaluated using a standard ResNet18 trained on `BlockMNIST` data. All feature attribution methods exhibit feature leakage, as the attributions highlight the non-predictive null block in addition to the `MNIST` signal block. Surprisingly, in some cases, `LRP` (third row) exclusively highlights the null block.

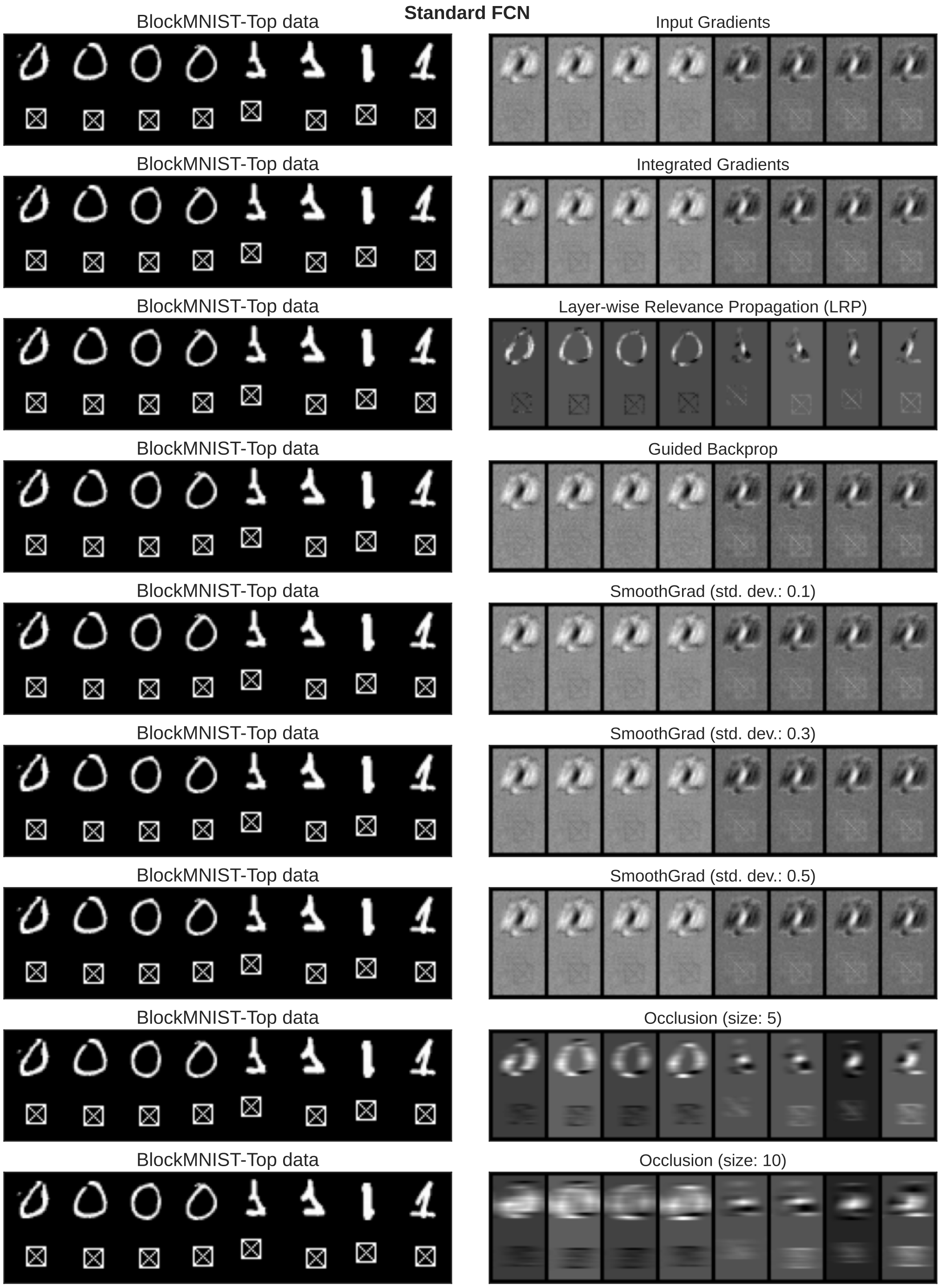

Figure 27: Multiple instance-specific feature attribution methods evaluated using a standard two-layer MLP trained on `BlockMNIST-Top` images, in which the `MNIST` signal block is fixed at the top. On this dataset, feature attributions of all five methods highlight discriminative features in the signal block, suppress the null block, and satisfy (`A`).

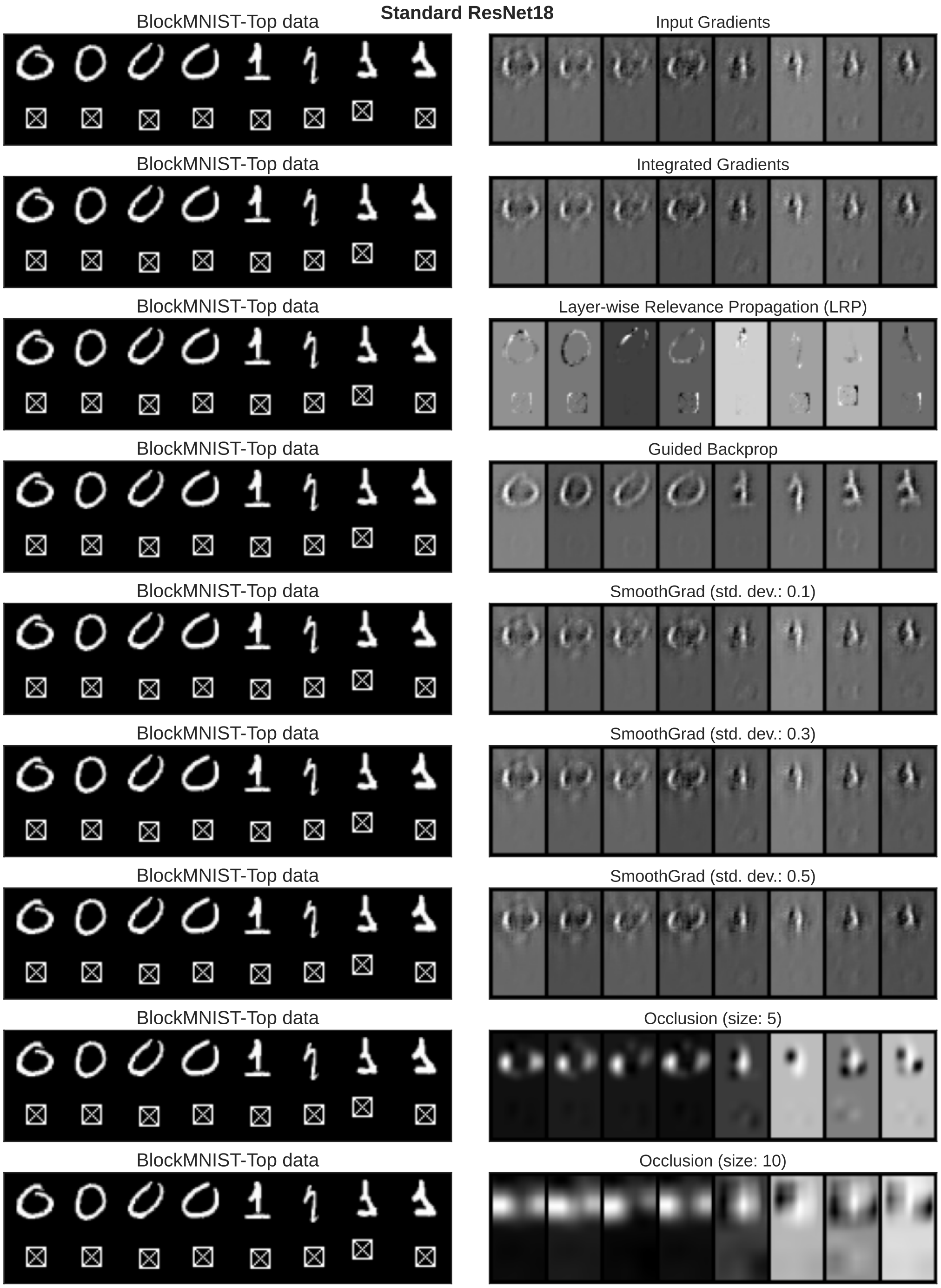

Figure 28: Multiple instance-specific feature attribution methods evaluated using a standard ResNet18 trained on `BlockMNIST-Top` data, in which the `MNIST` signal block is fixed at the top. On this dataset, feature attributions of all five methods highlight discriminative features in the signal block, suppress the null block, and satisfy (`A`). Surprisingly, `LRP` attributions (third row) highlight the null patch of `BlockMNIST-Top` images as well.

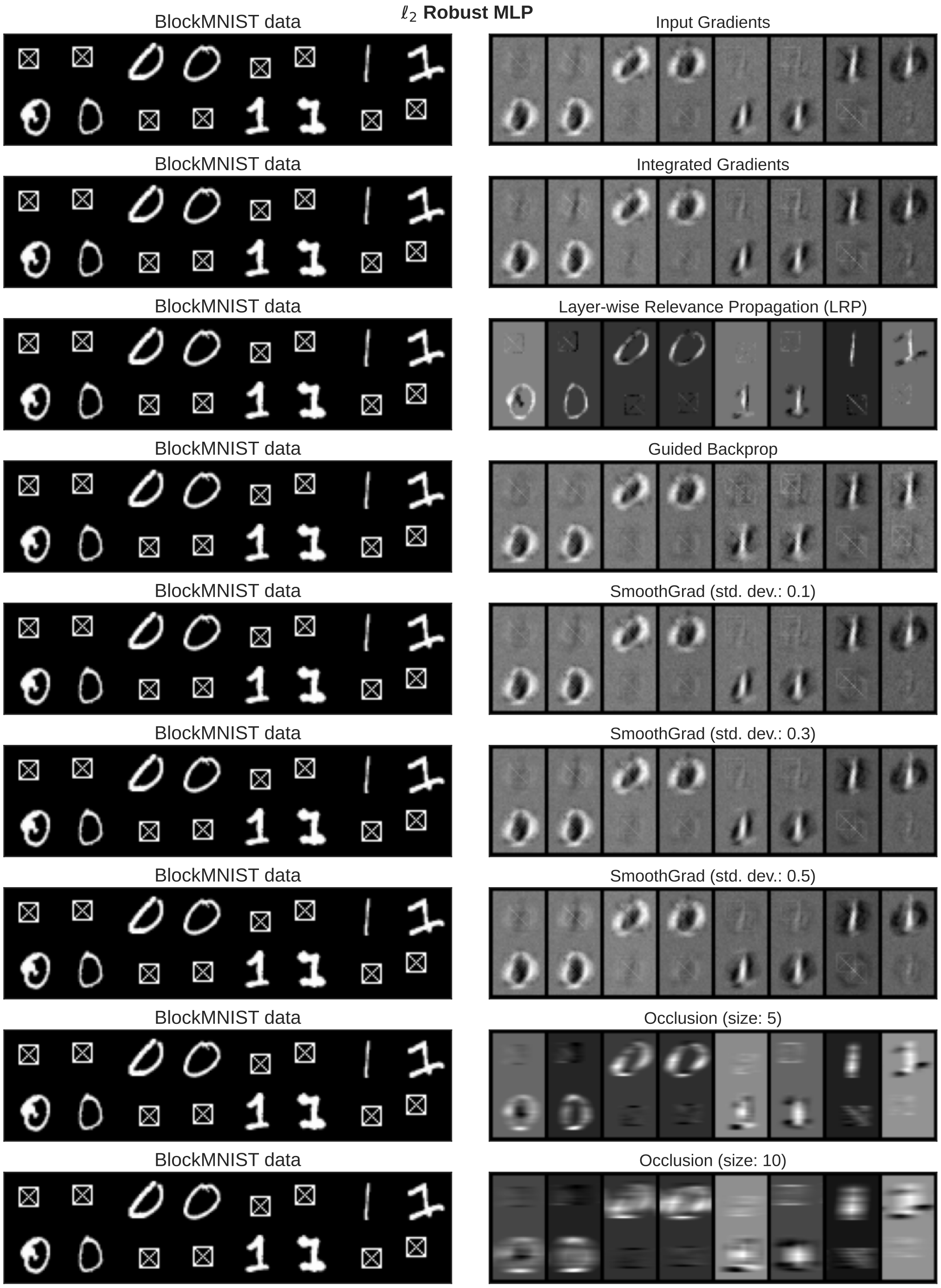

Figure 29: Multiple instance-specific feature attribution methods evaluated using a ℓ_2 robust two-layer MLP trained on `BlockMNIST` data. Consistent with our findings on adversarial robustness vis-a-vis feature leakage, feature attributions of all methods of robust models do not exhibit feature leakage.

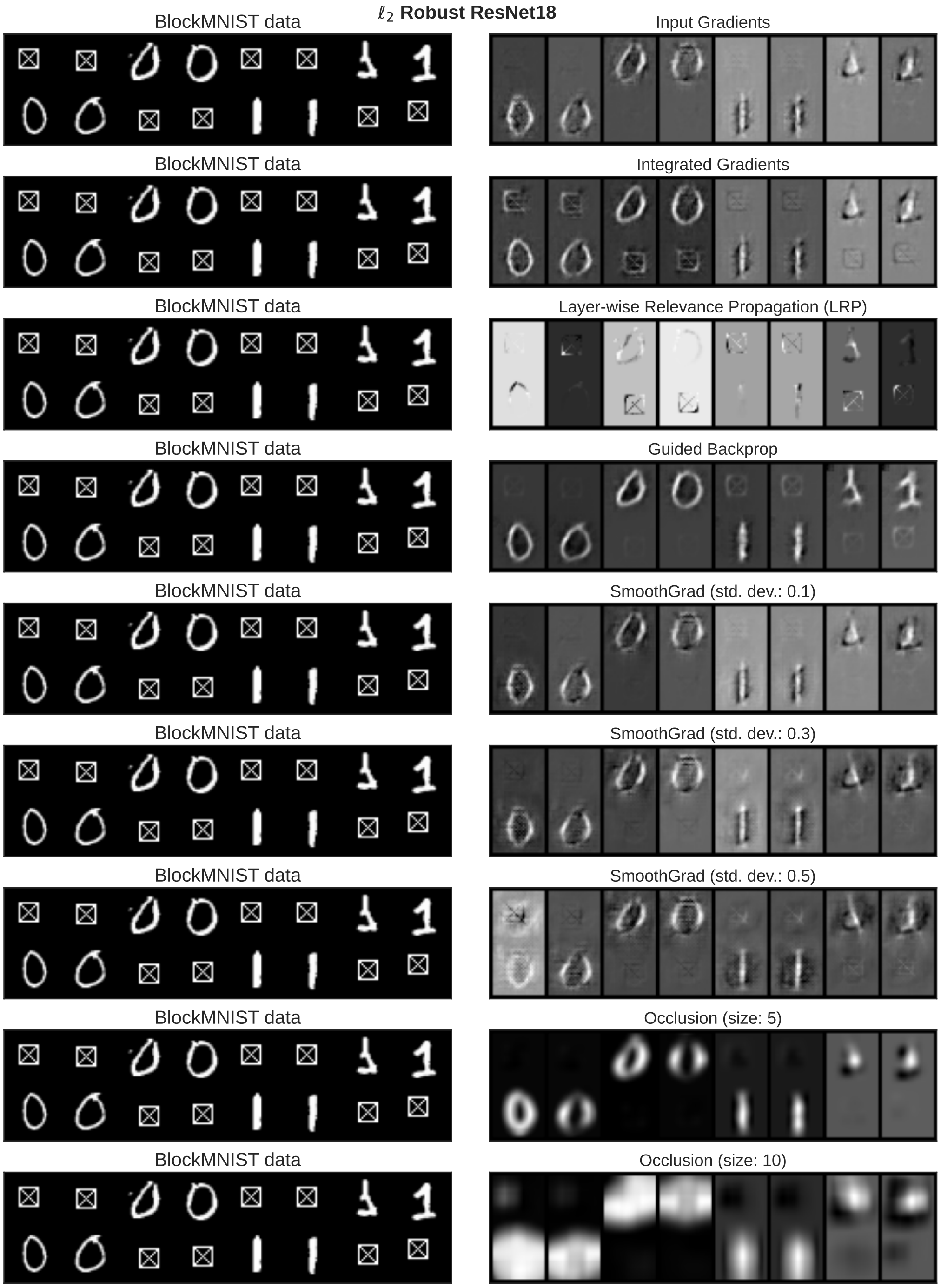

Figure 30: Multiple instance-specific feature attribution methods evaluated using a ℓ_2 robust ResNet18 trained on `BlockMNIST` data. Consistent with our findings on adversarial robustness vis-a-vis feature leakage, feature attributions of all methods (except `LRP`) of robust models do not exhibit feature leakage.

E Proof of Theorem 1

We first begin with the definition of a function, $\psi : \mathbb{R}^2 \to \mathbb{R}$ which will prove useful in the analysis:

$$\psi(a,b) := \phi(a+b) - \phi(-a+b) = \left\{ \begin{array}{cc} a-b & \text{if } a \leq -|b| \\ 0 & \text{if } b \leq 0,\ |a| \leq |b| \\ 2a & \text{if } b \geq 0,\ |a| \leq |b| \\ a+b & \text{if } a \geq |b| \end{array} \right. , \tag{5}$$

where we recall that $\phi(a) = \max(a,0)$ is the ReLU nonlinearity.

Proof of Theorem 1 in the rich regime. We first claim that the max-margin classifier (4) is given by $\nu^* = \frac{1}{2}\delta_{\theta_1^*} + \frac{1}{2}\delta_{\theta_2^*}$, where $\theta_1 := \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2((d/2)+(1-(\eta d/2)^2)}} z, \frac{(1-(\eta d/2))}{\sqrt{2((d/2)+(1-(\eta d/2))^2)}}\right)$ and $\theta_2 := \left(\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2((d/2)+(1-(\eta d/2))^2)}} z, \frac{(1-(\eta d/2))}{\sqrt{2((d/2)+(1-(\eta d/2))^2)}}\right)$ with $z \in \mathbb{R}^{\widetilde{d}\cdot d}$ denoting the concatenation of $d/2$ copies of $u^* \in \mathbb{R}^{\widetilde{d}}$ and $d/2$ copies of 0 vectors of dimension $\widetilde{d}$ each. To do so, we use [48, Proposition 12] which requires us to verify that there exists a probability distribution p^* over the training data points such that:

$$\text{Support}(\nu^*) \in \underset{(w,r,b)\in\mathbb{S}^{d\widetilde{d}+1}}{\arg\max}\ \mathbb{E}_{(x,y)\sim p^*}\left[y \cdot \left(w\phi\left(\langle r,x\rangle + b\right)\right)\right], \text{ and} \tag{6}$$

$$\text{Support}(p^*) \in \underset{(x,y)\in\mathcal{D}}{\arg\min}\, y \cdot \mathbb{E}_{(w,r,b)\sim\nu^*}\left[w\phi\left(\langle r,x\rangle + b\right)\right], \tag{7}$$

where $\mathbb{S}^{d\widetilde{d}+1}$ denotes the unit sphere in $\mathbb{R}^{d\widetilde{d}+2}$. In order to verify this condition, we use $p^* = \frac{1}{d}\sum_{\substack{j\in[d/2] \\ y\in\{\pm 1\}}} \delta_{(y\widetilde{u}_j, y)}$, where $\widetilde{u}_j \in \mathbb{R}^{\widetilde{d}\cdot d}$ is defined as the concatenation of d vectors, each in $\mathbb{R}^{\widetilde{d}}$, with the j^{th} one being u^*, the remaining $[d/2] \setminus \{j\}$ being $-\eta u^*$ and the last $[d/2]$ being all zero vectors.

We first prove (7). Consider the point $(\widehat{u}, y = 1)$ in the support of the training distribution with $\widehat{u} = (u^* + \eta g_1, \eta g_2, \cdots, \eta g_{d/2}, 0, \cdots, 0)$. We see that:

$$\begin{aligned} & y \cdot \mathbb{E}_{(w,r,b)\sim\nu^*}\left[w\phi\left(\langle r,\widehat{u}\rangle + b\right)\right] \\ & = \frac{1}{2}\cdot\frac{1}{\sqrt{2}}\left(\phi\left(\frac{\langle z,\widehat{u}\rangle}{\sqrt{2\left((d/2)+(1-(\eta d/2))^2\right)}} + \frac{1-(\eta d/2)}{\sqrt{2\left((d/2)+(1-(\eta d/2))^2\right)}}\right)\right. \\ & \qquad \left. -\phi\left(\frac{-\langle z,\widehat{u}\rangle}{\sqrt{2\left((d/2)+(1-(\eta d/2))^2\right)}} + \frac{1-(\eta d/2)}{\sqrt{2\left((d/2)+(1-(\eta d/2))^2\right)}}\right)\right) \end{aligned}$$

Since $\langle z,\widehat{u}\rangle = 1 + \eta\sum_{i\in[d/2]}\langle g_i, u^*\rangle \geq 1 - (\eta d/2) > 0$. Consequently, using (5), we have that:

$$y \cdot \mathbb{E}_{(w,r,b)\sim\nu^*}\left[w\phi\left(\langle r,\widehat{u}\rangle + b\right)\right] \geq \frac{1}{2\sqrt{2}} \cdot \frac{2\left(1-(\eta d/2)\right)}{\sqrt{2\left((d/2)+(1-(\eta d/2))^2\right)}} = y \cdot \mathbb{E}_{(w,r,b)\sim\nu^*}\left[w\phi\left(\langle r, y\widetilde{u}_j\rangle + b\right)\right].$$

This proves (7).

We now prove (6). For $\theta = (w,r,b)$, denote $\mathcal{L}(\theta) := \mathbb{E}_{\mathcal{D}}\left[y \cdot \left(w\phi\left(\langle r,x\rangle + b\right)\right)\right]$. We have $\mathcal{L}(\theta_1) = \mathcal{L}(\theta_2) = \frac{1}{d}\cdot\sum_{j\in[d/2]}\frac{1}{\sqrt{2}}\frac{2(1-(\eta d/2))}{\sqrt{2((d/2)+(1-(\eta d/2))^2)}} = \frac{1-(\eta d/2)}{2\sqrt{(d/2)+(1-(\eta d/2))^2}}$. We will now show that $\max_{\theta\in\mathbb{S}^{d+1}}\mathcal{L}(\theta) = \frac{1-(\eta d/2)}{2\sqrt{(d/2)+(1-(\eta d/2))^2}}$. For a given $\theta = (w,r,b)$, we first show that it is sufficient to consider the case where r is the concatenation of $\alpha_i u^*$ for some $\alpha_1, \cdots, \alpha_d$. In order to do this, given any $\theta = (w,r,b)$, let $\alpha_i := \langle r_i, u^*\rangle$ for $i \in [d/2]$ denote the inner product of the i^{th}-block vector of r with u^* and let $\bar{\alpha} := \frac{1}{d/2}\sum_{i\in[d/2]}\alpha_i$ be the mean of α_i. The function $\mathcal{L}(\theta)$ can be simplified as:

$$\mathcal{L}(\theta) = \frac{w}{d}\sum_{i\in[d/2]}\left(\phi\left(\alpha_i - (\eta d/2)\bar{\alpha} + b\right) - \phi\left(-\alpha_i + (\eta d/2)\bar{\alpha} + b\right)\right).$$

We can now consider r' to be the concatenation of $\langle r_i, u^*\rangle u^*$ for $i \in [d/2]$ and the remaining coordinates equal to zero, which ensures that $\mathcal{L}(\theta) = \mathcal{L}(\theta')$ for $\theta' = (w, r', b)$ and $\|\theta\| \geq \|\theta'\|$. We can then choose $|w'| \geq |w|$ such that $\|(w', r', b)\| = 1$ and $\mathcal{L}((w', r', b)) > \mathcal{L}(\theta)$. So it suffices to consider $\mathcal{L}(\theta)$ for $\theta = (w, r, b)$ where r is the concatenation of $\alpha_i u^*$ for some α_i for $i \in [d/2]$ and the remaining coordinates being set to zero. Let us consider two situations separately:

Case I, $b \geq 0$: Recall from (5) the definition $\psi(a, b) := \phi(a + b) - \phi(-a + b)$. First note from (5) that, $|\psi(a, b) - \psi(a', b)| \leq 2\,|a - a'|$ and $\psi(a, b) - \psi(a', b) \geq a - a'$ for every $a > a'$. If $\alpha_j < 0$ for some j, then choosing $r'_j = r_j - 2\alpha_i w^*$ with $r'_i = r_i$ for all $i \neq j$ gives us a corresponding θ' satisfying

$$\begin{aligned}\mathcal{L}(\theta') \geq \frac{w}{d} \sum_{i \neq j} \left(\phi\left(\alpha_i - (\eta d/2)\bar{\alpha} + b\right) - \phi\left(-\alpha_i + (\eta d/2)\bar{\alpha} + b\right)\right) - 2\eta(d/2 - 1)\,|\alpha_i| \\ + \frac{w}{d}\left(\phi\left(\alpha_i - (\eta d/2)\bar{\alpha} + b\right) - \phi\left(-\alpha_i + (\eta d/2)\bar{\alpha} + b\right)\right) + (1 - \eta)\,|\alpha_i| \\ \geq \mathcal{L}(\theta) + \frac{w}{d} \cdot (1 - \eta d) \cdot |\alpha_i|\,.\end{aligned}$$

So, it suffices to restrict our attention to θ such that $\alpha_i \geq 0$ for all i in order to prove (6). We will now show that making all α_i equal will further increase the value of $\mathcal{L}$. In order to see this, let $\alpha_1 = \min_{i \in [d/2]} \alpha_i$ and $\alpha_2 = \max_{i \in [d/2]} \alpha_i$. Then constructing r' from r by replacing α_1 and α_2 with $\alpha' := \frac{\alpha_1 + \alpha_2}{2}$ ensures that $\|r'\| \leq \|r\|$ while at the same time $\mathcal{L}((w, r', b)) - \mathcal{L}((w, r, b))$ since $\alpha_1 \geq 0$ implies $\alpha_1 - (\eta d/2)\bar{\alpha} > -\left(\alpha_2 - (\eta d/2)\bar{\alpha}\right)$. If $\alpha_i = \alpha_j$ for all $i, j \in [d]$, then from (5),

$$\mathcal{L}(\theta) = \frac{w}{d} \sum_{i \in [d]} \alpha_i - (\eta d/2)\bar{\alpha} + \min\left(\alpha_i - (\eta d/2)\bar{\alpha}, b\right) = (1 - (\eta d/2))w\left(\bar{\alpha} + \min\left(\bar{\alpha}, \frac{b}{1 - (\eta d/2)}\right)\right).$$

The maximizer of the above expression under the constraint $\|\theta\|^2 = w^2 + (d/2)\bar{\alpha}^2 + b^2 = 1$ can be seen to be when $w = \pm\frac{1}{\sqrt{2}}$, $\bar{\alpha} = \frac{2}{\sqrt{(d/2) + (1 - (\eta d/2))^2}}$ and $b = \frac{1 - (\eta d/2)}{\sqrt{2((d/2) + (1 - (\eta d/2))^2)}}$ achieving value $\frac{1 - (\eta d/2)}{2\sqrt{(d/2) + (1 - (\eta d/2))^2}}$.

Case II, $b < 0$: In this case, we have from (5) that

$$\begin{aligned}\mathcal{L}(\theta) \leq \frac{|w|}{d} \sum_{i \in [d]} \alpha_i - (\eta d/2)\bar{\alpha} = (1 - (\eta d/2))\,|w|\,\bar{\alpha} \leq \frac{(1 - (\eta d/2))\,|w|\,\|r\|}{\sqrt{d/2}} \leq \frac{1 - (\eta d/2)}{2\sqrt{d/2}} \\ < \frac{1 - (\eta d/2)}{\sqrt{(d/2) + (1 - (\eta d/2))^2}},\end{aligned}$$

where we used $\eta < \frac{1}{10d}$ in the last step. This shows that $\nu^* = \frac{1}{2}\delta_{\theta_1^*} + \frac{1}{2}\delta_{\theta_2^*}$ is a max-margin classifier satisfying (4).

Gradient magnitude: For any input (x, y), we note that the input gradient is of the form $\nabla_x \mathcal{L}(\nu^*, (x, y)) = \alpha z$ for some $\alpha \neq 0$. Consequently, the claim about the gradient magnitudes in different coordinates follows from the structure of z proved above.

□

F Effect of adversarial training

Consider training a model that is adversarially robust in an ℓ_p ball of radius ϵ. Assuming that the inner iterations of adversarial training find the optimal perturbations, it can be shown that if adversarial training converges asymptotically (i.e., in the rich regime), it does so to an appropriate max-margin classifier [68]:

$$\nu^{*} := \underset{\nu \in \mathcal{P}\left(\mathbb{S}^{d\widetilde{d}+1}\right)}{\arg\max} \; \min_{(x,y)\in B_p(\mathcal{D},\epsilon)} y \cdot f(\nu, x), \tag{8}$$

where $B_p(\mathcal{D},\epsilon) := \left\{(x,y) : (\tilde{x},y) \sim \mathcal{D}, \|x-\tilde{x}\|_p \leq \epsilon\right\}$. This implies that using the techniques of previous section, we should be able to analyze the input gradient. However, such analysis requires an explicit form of the max-margin classifier defined above. In contrast to the standard training studied above, computing explicit form of the max-margin classifier is significantly non-trivial in the adversarially training case even for the simple special case of $\widetilde{d} = 1, \eta = 0$ and $u^* = 1$. While we are unable to explicitly compute the max-margin classifier even for this case, we make the following conjecture about the max-margin classifier.

Conjecture 1. Let data distribution $\mathcal{D}$ follow (2) with $\widetilde{d} = 1, \eta = 0$ and $u^* = 1$. Then, the classifier $\tilde{\nu}$ defined below is a max-margin classifier for adversarial training (8) for $p = \infty$ and ϵ close to 0.5:

$$\tilde{\nu} := \frac{1}{d} \sum_{i \in [d/2]} \delta_{\theta_i} + \delta_{\theta_i'}, \tag{9}$$

with $\theta_i := (\frac{1}{\sqrt{2}}, \frac{3}{\sqrt{20}} e_i, \frac{-1}{\sqrt{20}}), \theta_i' := (\frac{-1}{\sqrt{2}}, \frac{-3}{\sqrt{20}} e_i, \frac{-1}{\sqrt{20}})$ where $e_i \in \mathbb{R}^d$ denotes i^{th} standard basis vector.

Figure 31 empirically verifies two consequences of this conjecture. In Figure 31(a), we show that first-layer weights with large alignment with standard basis vectors also have large second-layer weights, indicating that axis-aligned first-layer weights are highly influential in the final model's prediction. Figure 31(b) shows that the biases in first-layer ReLU units are predominantly negative.

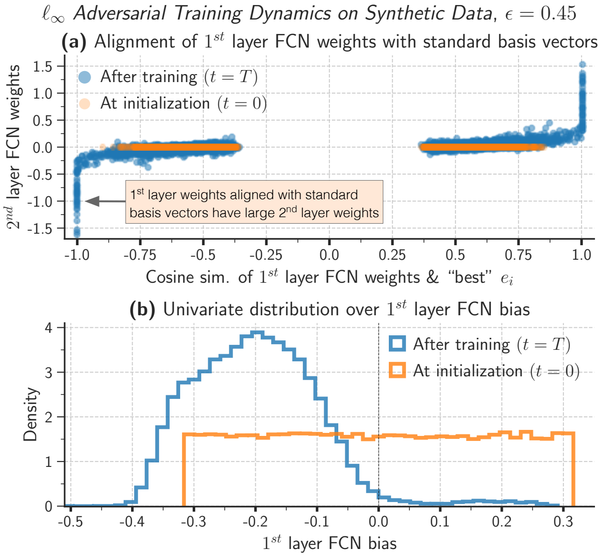

Figure 31: Adversarial training dynamics for training one-hidden-layer FCNs with width $50,000$ on 10-dimensional synthetic data. Subplot (a) shows that first-layer neurons aligned with standard basis vectors have large second-layer weights. Given a normalized 1^{st} layer weight vector (i.e., rescaled so that it has unit ℓ_2 norm), the x-axis in (a) plots the coordinate with largest magnitude in this normalized vector. Note that the gap around origin is due to that fact that the largest magnitude coordinate in a unit ℓ_2 norm vector in $d = 10$ dimensions is at least $\frac{1}{\sqrt{d}} \approx 0.32$. Subplot (b) shows that ℓ_∞ adversarial training results in first-layer bias terms that are predominantly negative. These observations support Conjecture 1 and indicate that adversarial training quickly enters the rich regime.

The following lemma shows that the input gradients of $\tilde{\nu}$ in (9) indeed highlight $j^*(x)$.

Lemma 1. *Let data distribution $\mathcal{D}$ follow* (2) *with $\widetilde{d} = 1, \eta = 0$ and $u^* = 1$ and let $\tilde{\nu}$ be as defined in* (9)*. Then, for any $(x, y) \sim \mathcal{D}$, we have: $\nabla_x \mathcal{L}(\tilde{\nu}, (x, y)) = c \cdot e_{j^*(x)}$, where $c \neq 0$ is a constant.*

Assuming Conjecture 1, this lemma shows that for the special case $\widetilde{d} = 1$ and $\eta = 0$, adversarially trained models have input gradients that reveal instance-specific features important for classification. Conjecture 1 and Lemma 1 also explain several other empirically observed properties of adversarial training such as visually perceptible input gradients and adversarial examples [29]. In this section, we prove Lemma 1.

Proof of Lemma 1. Given the classifier $\tilde{\nu}$ and a data point (x, y), the input gradient is given by

$$\begin{aligned} \nabla_x \mathcal{L}(\tilde{\nu}, (x, y)) &= c \cdot \nabla_x f(\tilde{\nu}, x) \\ &= c \cdot \mathbb{E}_{(w,r,b)\sim\tilde{\nu}} \left[w\phi' \left(\langle r, x \rangle + b \right) r \right], \end{aligned}$$

where $c = \frac{-y \exp(-y \cdot f(\tilde{\nu}, x))}{1+\exp(-y \cdot f(\tilde{\nu}, x))}$. Recall from (9) that

$$\tilde{\nu} = \frac{1}{d} \sum_{i \in [d/2]} \delta_{\theta_i} + \delta_{\theta_i'},$$

where $\theta_i{:=}(\frac{1}{\sqrt{2}}, \frac{3}{\sqrt{20}} e_i, \frac{-1}{\sqrt{20}})$ and $\theta_i'{:=}(\frac{-1}{\sqrt{2}}, \frac{-3}{\sqrt{20}} e_i, \frac{-1}{\sqrt{20}})$, e_i denotes the i^{th} standard basis vector in $\mathbb{R}^d$. If $(x, y) = (ye_{j^*(x)}, y)$ and $(w, r, b) \sim \delta_{\theta_i}$ or $(w, r, b) \sim \delta_{\theta_i'}$, then $\Pr\left[\phi'\left(\langle r, x \rangle + b\right) \neq 0\right] > 0$ if and only if $i = j^*(x)$. Consequently, the only contribution the input gradient comes from $\delta_{\theta_{j^*(x)}}$ and $\delta_{\theta'_{j^*(x)}}$. So,

$$\begin{aligned} \nabla_x \mathcal{L}(\tilde{\nu}, (x, y)) &= c \cdot \mathbb{E}_{(w,r,b)\sim\tilde{\nu}} \left[w\phi' \left(\langle r, x \rangle + b \right) r \right] \\ &= c' \cdot \left(\mathbb{E}_{(w,r,b)\sim\delta_{\theta_{j^*(x)}}} \left[w\phi' \left(\langle r, x \rangle + b \right) r \right] + \mathbb{E}_{(w,r,b)\sim\delta_{\theta'_{j^*(x)}}} \left[w\phi' \left(\langle r, x \rangle + b \right) r \right] \right) \\ &= c'' \cdot e_{j^*(x)}. \end{aligned}$$

This proves the result. □