Towards stable and sparse saliency maps via feature map smoothing

Anonymous authors
Paper under double-blind review

Abstract

Input-gradient-based attribution methods, such as Vanilla Gradient, Integrated Gradients, and SmoothGrad, are widely used to explain image classifiers via saliency maps. However, these methods often produce explanations that are noisy, or unstable. While prior work primarily focuses on refining the explanation techniques themselves, we explore a complementary model-centered perspective grounded in explainability-by-design. Specifically, we examine how adversarial training affects saliency map quality and propose a lightweight feature-map smoothing mechanism that can be integrated during training. Evaluating across FMNIST, CIFAR-10, and ImageNette, we find that local smoothing (e.g., mean, median filters) improves stability and perceived clarity of explanations while preserving sparsity gains from adversarial training. However, gains in faithfulness are method and dataset dependent, highlighting that interpretability improvements may not generalize uniformly. A user study with 65 participants further confirms that explanations from smoothed adversarial models are perceived as more comprehensible and trustworthy. Our work highlights the value of model-level interventions for improving post-hoc explanations. Our code is available at `https://anonymous.4open.science/r/ImprovingVG-2BFA/README.md`.

1 Introduction

Input-gradient-based explanation methods highlight input features that most influence a model's decision, often visualized as saliency maps in the context of image classification. Classic method like Vanilla Gradient (VG) (Simonyan et al., 2014), computes gradients across input pixels, ranking features by their gradient magnitude. While prior studies have shown that input-gradients can capture relevant information regarding a model output (Samek et al., 2016), VG suffers from noisy saliency map. Hence, various methods like Integrated Gradients (IG) (Sundararajan et al., 2017), and SmoothGrad (SG) (Smilkov et al., 2017) have been proposed that modifies the input-gradient approach to reduce saliency map noise and improve the visual quality of the explanations. While widely adopted, these methods frequently produce saliency maps that are either excessively noisy, overly sparse, or lacking quantitative robustness, which can hinder human comprehension (Adebayo et al., 2018; Kindermans et al., 2019; Nie et al., 2018).

To be useful, saliency maps should satisfy several desiderata like sparsity, stability and faithfulness. Sparsity of saliency maps measures if explanations focus on the most relevant features by discarding irrelevant ones (Chalasani et al., 2020). Stability evaluates if saliency maps are consistent across small input perturbations (Alvarez-Melis & Jaakkola, 2018); and faithfulness measures if explanations accurately reflect the model's actual decision-making process (Rong et al., 2022). These attributes are essential for explanations to be trustworthy and actionable in real-world applications.

Most efforts to improve these properties focus on modifying the explanation methods themselves. In contrast, we pursue a complementary approach by modifying the model training procedure so that interpretability emerges from the learned representations of the model. Specifically, we investigate how training strategies influence the quality of saliency maps in input-gradient-based methods. While adversarial training (Goodfellow et al., 2015) is primarily designed to enhance model robustness, it has also been observed to yield sparser and sometimes more interpretable saliency maps (Etmann et al., 2019; Chalasani et al., 2020). However,

our analysis shows that adversarial training often degrades stability—producing explanations that are inconsistent across small input variations. To address this sparsity–stability trade-off, we propose a lightweight training-time intervention: the insertion of a feature-map smoothing block during adversarial training. This block applies local smoothing filters (e.g., mean, median, Gaussian) to intermediate feature maps, regularizing the learned representations and encouraging locally consistent gradients. We find that this technique can preserve the sparsity benefits of adversarial training while significantly enhancing the stability and structural coherence of saliency maps.

We evaluate our approach across three datasets (FMNIST, CIFAR-10, and ImageNette) using three widely adopted gradient-based explanation methods, Vanilla Gradient (VG) (Simonyan et al., 2014), Integrated Gradient (IG) (Sundararajan et al., 2017), and SmoothGrad (SG) (Smilkov et al., 2017), along three key desiderata, sparsity, stability, and faithfulness. Our findings show that local smoothing filters integrated into adversarial training consistently enhance explanation stability over adversarial training alone, largely preserving the sparsity gains. However, improvements in faithfulness are more dataset and method dependent. Additionally, a user study involving 65 participants demonstrates that explanations from feature-map smoothed models are perceived as more comprehensible and trustworthy than those from adversarial or naturally trained models.

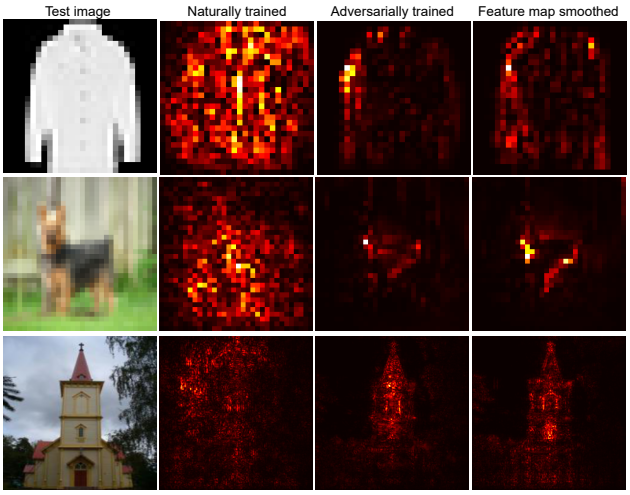


Figure 1: Saliency maps examples using Vanilla Gradient for different models that correctly classify the test images. Natural models produce noisy saliency maps (2^{nd} column), adversarial models produce sparser maps (3^{rd} column), and feature-map smoothed models smoothens the sparse maps (4^{th} column), improving comprehensibility.

Figure 1 illustrates this visually. Saliency maps from naturally trained models are noisy and diffuse *(second column)*. Adversarially trained models yield sparser maps, but overly sparse saliency maps can lead to incomplete model understanding *(third column)*. Our proposed approach, adversarial training with feature-map smoothing, yields explanations that strike a balance between sparsity and comprehensibility—highlighting key regions while maintaining structural coherence *(fourth column)*. Additional visualizations are provided in Appendix J.

2 Related Work

As highlighted by Ilyas et al. (2019), explanations that are meaningful and faithful to a model's decision-making process cannot be pursued independently from how the model is trained, a principle central to our approach. Below, we discuss two relevant background: improving saliency maps through training, and analyzing explanations in robust models.

Improving saliency maps through training modifications. Several prior works have proposed modifying training procedures to improve the quality of saliency maps. Kim et al. (2019) introduce layer-wise

thresholding during backpropagation, while Dombrowski et al. (2019) suggest replacing ReLU with soft-plus activations to yield smoother explanations. Wicker et al. (2023) propose a certified training framework for explanation robustness, and Chenyang & Chan (2023) train object detectors with explicit constraints on attribution consistency. While these approaches typically modify the network architecture or loss function, we instead introduce a simple feature-map smoothing block during adversarial training, focused on improving the interpretability of saliency maps via improved stability.

Saliency maps in adversarially trained models. Other works have analyzed saliency maps under adversarial training. Etmann et al. (2019) and Zhang & Zhu (2019) show that adversarially trained models yield sparser and shape-aligned explanations, attributing this to increased robustness. Chalasani et al. (2020) formally connect adversarial robustness and saliency sparsity in simple settings, but their results focus on 1-layer models. Shah et al. (2021) report that adversarial training improves explanation quality by suppressing spurious features, while Mangla et al. (2020) explore attribution robustness. In contrast, we focus on a previously underexplored trade-off: *while adversarial training improves sparsity, it often reduces explanation stability.* Our work proposes a simple local feature smoothing technique during adversarial training that improves stability without compromising sparsity, thus enhancing the reliability and interpretability of input-gradient explanations.

3 Method

Preliminaries: Consider a differentiable function $F(\mathbf{x})$, which represents a deep neural network. For simplicity, let us examine a single-layer model with the form $F(\mathbf{x}) = H(\langle \mathbf{w}, \mathbf{x} \rangle)$, where H is a differentiable scalar-valued activation function, $\langle \mathbf{w}, \mathbf{x} \rangle$ is the dot product between the weight vector $\mathbf{w}$ and input $\mathbf{x} \in \mathbb{R}^d$. The Vanilla Gradient (VG) method (Simonyan et al., 2014) measures the sensitivity of the model output $F(\mathbf{x})$ with respect to each feature of the input $\mathbf{x}$. This is given by computing the gradient of the output $F(\mathbf{x})$ with respect to the input $\mathbf{x}$. The Integrated Gradients (IG) method (Sundararajan et al., 2017) averages the gradients along a straight-line path from a baseline input $\mathbf{x}'$ (often a zero vector) to the actual input $\mathbf{x}$. SmoothGrad (SG) (Smilkov et al., 2017) improves on any gradient-based explanations like VG or IG by adding random noise to the input $\mathbf{x}$ multiple times, calculating the explanations for each noisy version, and then averaging the results. While these methods are widely used, their stability, how consistent the explanations remain under small perturbations, is crucial for reliability. We next establish a formal connection between model sensitivity and explanation stability.

3.1 Relationship between explanation stability and model sensitivity

We first compute explanation using Vanilla Gradient (VG) given by:

$$VG(\mathbf{x}) = \frac{\partial F(\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial H(\langle \mathbf{w}, \mathbf{x} \rangle)}{\partial \mathbf{x}} = H'(\langle \mathbf{w}, \mathbf{x} \rangle)\mathbf{w} \tag{1}$$

Here, $H'(\langle \mathbf{w}, \mathbf{x} \rangle)$ is the gradient of activation function H with respect to the $\langle \mathbf{w}, \mathbf{x} \rangle$. For example, for a sigmoid activation function, $H'(z) = H(z)(1 - H(z))$ where $z = \langle \mathbf{w}, \mathbf{x} \rangle$. This gives the VG attribution as:

$$VG^F(\mathbf{x}) = H(\langle \mathbf{w}, \mathbf{x} \rangle)(1 - H(\langle \mathbf{w}, \mathbf{x} \rangle))\mathbf{w} = F(\mathbf{x})(1 - F(\mathbf{x}))\mathbf{w} \tag{2}$$

Similarly, the Integrated Gradients (IG) feature attribution score for feature i of input image $\mathbf{x} \in R^d$ with baseline $\mathbf{u}$ for model F is given by Eqn. 3:

$$IG_i^F(\mathbf{x}, \mathbf{u}) = (x_i - u_i). \int_{\alpha=0}^{1} \partial_i F(\mathbf{u} + \alpha(\mathbf{x} - \mathbf{u}))\partial\alpha \tag{3}$$

Using a closed-form expression from Chalasani et al. (2020), IG can be rewritten as Eqn. 4:

$$IG^F(\mathbf{x}, \mathbf{u}) = [F(\mathbf{x}) - F(\mathbf{u})]\frac{(\mathbf{x} - \mathbf{u}) \odot \mathbf{w}}{\langle \mathbf{x} - \mathbf{u}, \mathbf{w} \rangle} \tag{4}$$

In SmoothGrad (SG), we add Gaussian noise $\mathbf{n} \sim \mathcal{N}(0, \sigma^2)$ to the input $\mathbf{x}$ and compute the input-gradient for multiple noisy samples $\mathbf{x}_k = \mathbf{x} + \mathbf{n}_k$ for $k = 1, \ldots, N$, where N is the number of noise samples. SG explanation, when aggregating VG, is given by:

$$SG(\mathbf{x}) = \frac{1}{N}\sum_{k=1}^{N}\frac{\partial F(\mathbf{x}_k)}{\partial \mathbf{x}_k} = \frac{1}{N}\sum_{k=1}^{N}\frac{\partial H(\langle \mathbf{w}, \mathbf{x}_k \rangle)}{\partial \mathbf{x}_k} = \frac{1}{N}\sum_{k=1}^{N} H'(\langle \mathbf{w}, \mathbf{x}_k \rangle).\mathbf{w} = \frac{1}{N}\sum_{k=1}^{N} F(\mathbf{x_k})(1 - F(\mathbf{x_k}))\mathbf{w} \tag{5}$$

Now consider $\mathbf{x}' \in \mathcal{N}_\mathbf{x}$ is a noisy version of input image $\mathbf{x}$ where $\mathcal{N}_\mathbf{x}$ indicates a neighborhood of inputs $\mathbf{x}$ where the model prediction is locally consistent. The stability of explanations-VG, IG and SG-can be computed by measuring the norm of the difference between the original explanation and explanation for the noisy image. Using Eqns. 1, 4 and 5, we obtain,

$$\Delta_{VG} = ||VG^F(\mathbf{x}') - VG^F(\mathbf{x})||_1 \leq (F(\mathbf{x}') - F(\mathbf{x}))).\mathbf{w} \tag{6}$$

$$\Delta_{IG} = ||IG^F(\mathbf{x}', \mathbf{u}) - IG^F(\mathbf{x}, \mathbf{u})||_1 \approx ||IG^F(\mathbf{x}', \mathbf{x})||_1 \approx \left|\left|[F(\mathbf{x}') - F(\mathbf{x})]\frac{(\mathbf{x}' - \mathbf{x}) \odot \mathbf{w}}{\langle \mathbf{x}' - \mathbf{x}, \mathbf{w} \rangle}\right|\right|_1 \tag{7}$$

$$\Delta_{SG} = \sum_{k=1}^{N} ||SG^F(\mathbf{x}') - SG^F(\mathbf{x})||_1 \leq \frac{1}{N}(F(\mathbf{x}') - F(\mathbf{x}))).\mathbf{w} \tag{8}$$

Since $\mathbf{w}$ is fixed for a given model, the bounds in Eqns 6, 7 and 8 indicate that the stability of explanations is influenced by the model sensitivity $F(\mathbf{x}') - F(\mathbf{x})$, setting up a basis for using methods that enhance explanation stability by reducing model sensitivity. However, these bounds do not imply a strict functional relationship, but rather offer an intuitive approximation of how explanation variance may behave in practice. For a detailed derivation, see Appendix H, and for conditions affecting the tightness of these bounds, refer to Appendix G.

3.2 Adversarial training and impact on saliency map stability

Building on the observations from Section 3.1, various regularization strategies can enhance the quality and stability of saliency maps. One such approach is natural training-based regularization, which involves incorporating explicit smoothness constraints on the model's gradients. A fundamental technique in this category is input noise regularization, where Gaussian noise is injected into training samples during optimization (Bishop, 1995). This method has been shown to produce smoother and more reliable saliency maps (Smilkov et al., 2017).

To explicitly address the issue of model sensitivity to worst-case perturbations, we consider adversarial training (Goodfellow et al., 2015) as a method for improving robustness. Adversarial training modifies the standard loss function by solving the following min-max objective:

$$\mathbb{E}_{(\mathbf{x},y)\sim D}\left[\max_{\|\delta\|_\infty \leq \epsilon} \mathbb{L}(\mathbf{x} + \delta, y; \mathbf{w})\right],$$

where $\mathbf{x} \in \mathbb{R}^d$ is an input sample, y is its ground-truth label, δ is a worst-case perturbation bounded by ϵ under the ℓ_∞ norm, $\mathbb{L}(\cdot)$ is the loss function (e.g., cross-entropy), and $\mathbf{w}$ denotes model parameters. The inner maximization computes an adversarial input $\mathbf{x}+\delta$ that maximizes the loss, while the outer minimization trains the model to minimize this adversarial risk, encouraging robustness to input perturbations.

Adversarial training modifies the loss function to minimize sensitivity to input perturbations by solving $\mathbb{E}_{(\mathbf{x},y)\sim D}\left[\max_{\|\delta\|_\infty \leq \epsilon} \mathbb{L}(\mathbf{x} + \delta, y; \mathbf{w})\right]$ where δ is a small, worst-case perturbation and ϵ is the perturbation

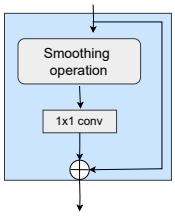


Figure 2: Feature-map smoothing block

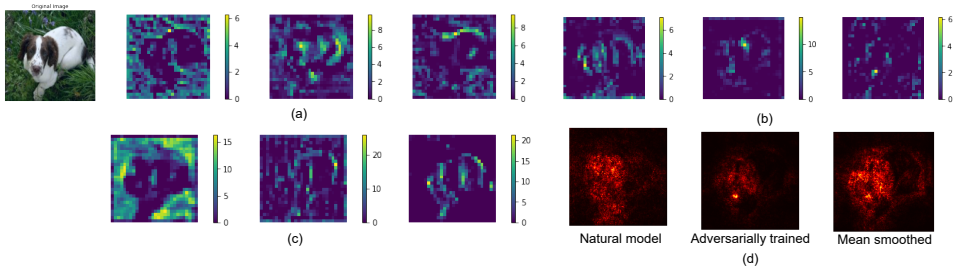


Figure 3: Plot of feature maps (channel=7, 21, 127) after first residual block for a test image on different ResNet18 ImageNette models: (a) a naturally trained model, (b) an adversarially-trained model, (c) an adversarially-trained model with feature-map smoothing (mean filter) (d) corresponding saliency maps using Vanilla Gradient.

bound. The inner maximization finds the most adversarial perturbation within the constraint $||\delta||_\infty \leq \epsilon$, while the outer minimization ensures that the model learns to be invariant to such perturbations.

In Figure 3, given a test image from the ImageNette dataset, we visualize feature maps derived from (a) a naturally trained model and (b) an adversarially trained model. All models use the identical ResNet18 architecture (He et al., 2016) and training settings (discussed in Appendix A). Feature maps are extracted from the first residual block (which consists of 128 channels), and three representative channels are shown for comparison. A key observation is that adversarial training shrinks many feature activations, leading to more selective attention in learned representations. This behavior directly affects input-gradient-based saliency maps, making them sparser in adversarially trained models compared to naturally trained ones (see Figure 3(d)).

However, adversarial training does not necessarily improve explanation stability in deep networks. As we demonstrate in Sections 4.1 and 6, while adversarial training enforces sparsity in saliency maps, it does not guarantee stability and comprehensibility of saliency maps. This leads to a trade-off: *sparser explanations may enhance readability but can also reduce attribution stability.* Our findings suggest that while adversarial training enhances sparsity of saliency maps, additional mechanisms (such as feature-map smoothing) may be required to preserve the stability of gradient-based explanations.

3.3 Feature map smoothing for stable explanations

To address the limitations of adversarial training on saliency map stability, we propose a lightweight regularization strategy: feature map smoothing (Xie et al., 2019). This technique integrates simple spatial smoothing filters (e.g., mean, median, Gaussian) into intermediate feature representations during adversarial training. By regularizing local activations, we aim to reduce sharp activation changes that amplify gradient instability, thereby producing saliency maps that are both sparse and stable. Unlike input-level noise methods such as SmoothGrad (Smilkov et al., 2017), which average gradient outputs post hoc, our method enforces local smoothness directly at the representation level.

As illustrated in Figure 2, we implement a smoothing block composed of a spatial filter followed by a 1×1 convolution and a residual connection. This module can be inserted into any convolutional layer with minimal impact on model accuracy. As shown in Appendix C, smoothing filters alone do not alter benign or robust accuracy substantially ($\pm 3\%$ range on FMNIST and CIFAR-10), and when combined with adversarial training, they can improve robustness in some settings. However, accuracy trade-offs can be more pronounced on larger datasets like ImageNette.

In our study, we focus on three simple local filters: mean, median and Gaussian filtering, which are easy to implement and computationally efficient. A mean filter replaces each feature with the average of nearby features within a defined kernel. A median filter computes the median value within a small sliding window over the feature map. A Gaussian filter applies a smoothing effect to feature maps by convolving them with a Gaussian kernel, effectively reducing Gaussian noise. See Appendix B for more discussion on each filter.

As shown in Figure 3(c), applying feature map smoothing to an adversarially trained model introduces a noticeable smoothing effect, which varies depending on the type of filter used. For example, mean filtering reduces rapid fluctuations in feature map values by averaging neighboring activations. While adversarial training alone (Figure 3(b)) shrinks feature activations leading to discontinuities in the learned representations, the addition of smoothing alleviates this issue by preserving key feature activations while eliminating high-frequency artifacts typically seen in naturally trained models. This results in smoother and more interpretable saliency maps, as illustrated in Figure 3(d). Furthermore, feature map smoothing aligns with the stability bounds derived in Section 3.1. Since smoothing reduces the norm of feature variations specifically, $\|F(\mathbf{x}') - F(\mathbf{x})\|$, this leads to tighter stability bounds for input-gradient based explanation methods (further discussed in Appendix G).

4 Experiment and Analysis

4.1 Experiment Framework

Setup: We evaluate our approach on three datasets: FMNIST (Xiao et al., 2017), CIFAR-10 (Krizhevsky et al., 2009), and ImageNette (Howard, 2020), training several model variants for each: 1) naturally trained (N), 2) adversarially trained (A), 3) adversarial training with mean-filter smoothing (M1), 4) adversarial training with median-filter smoothing (M2), 5) adversarial training with Gaussian-filter smoothing (G). Following the setup from Chalasani et al. (2020), we use LeNet (LeCun et al., 1998) for FMNIST and Wide-ResNet (Zagoruyko & Komodakis, 2016) for CIFAR-10. We use ResNet-18 He et al. (2016) for ImageNette. For adversarial training, we apply perturbations under the l_∞ norm using the PGD attack (Madry et al., 2018). The models are trained with $\epsilon = 0.1$ for FMNIST and CIFAR-10, and $\epsilon = 1/255$ for ImageNette, as these values yielded the best performance across our evaluations. We achieved optimal results by adding the smoothing block after the first convolutional or residual block. We discuss the impact of altering the smoothing block's position in Appendix E. Full details of our datasets and training methodology are provided in Appendix A. We also discuss the effect of feature map smoothing on saliency map quality for VGG (Simonyan & Zisserman, 2015) network in Appendix F.

Evaluation Metrics: Given a saliency map from Vanilla Gradient (VG), Integrated Gradients (IG) and SmoothGrad (SG) for each model and dataset, we compute its sparseness using Gini index (G) (Chalasani et al., 2020), and its stability using relative input stability (RIS), and relative output stability (ROS) (Agarwal et al., 2022). We analyze faithfulness using ROAD plot analysis (Rong et al., 2022), quantifying the plot using area over the perturbation curve, computed as $ROAD_{AOPC} = \frac{1}{L+1}\sum_{k=1}^{L}\left\langle f(x^{(0)}) - f(x^{(k)})\right\rangle$ where, L represents the number of feature removal steps, and $f(x)$ is the classifier's output probability for the originally predicted class given the input x. The term $x^{(0)}$ corresponds to the unperturbed input image, while $x^{(k)}$ represents the image after k perturbation steps. We evaluate saliency map similarity using structural similarity index (SSIM) (Adebayo et al., 2018). All results are aggregated for 1000 randomly selected test images that the model accurately classifies across all datasets. See Appendix I for detail discussion on metrics.

4.2 Results and discussion

Similar to Chalasani et al. (2020), we compare the sparsity, stability and faithfulness improvement of saliency maps with respect to the naturally trained model (N). Specifically, for a given training method M $\in$ {A, M1, M2, G}, we compute the following metrics that quantify the improvement in sparseness (dG), relative input stability (dRIS), relative output stability (dROS) and faithfulness (dROAD) of the explanation method $\phi(.) \in$ {VG, IG, SG}

$$
\begin{aligned}
dG[\phi(\mathbf{x})] &= G^M[\phi(\mathbf{x})] - G^N[\phi(\mathbf{x})] \\
dRIS[\phi(\mathbf{x})] &= RIS^M[\phi(\mathbf{x})] - RIS^N[\phi(\mathbf{x})] \\
dROS[\phi(\mathbf{x})] &= ROS^M[\phi(\mathbf{x})] - ROS^N[\phi(\mathbf{x})] \\
dROAD[\phi(\mathbf{x})] &= ROAD^M_{AUC}[\phi(\mathbf{x})] - ROAD^N_{AUC}[\phi(\mathbf{x})]
\end{aligned}
$$

4.2.1 On the sparsity, stability and faithfulness of saliency maps

Table 1: Sparsity–Stability–Faithfulness evaluation of Vanilla Gradient (VG), Integrated Gradients (IG), and SmoothGrad (SG) on FMNIST, CIFAR-10, and ImageNette on adversarially trained (A), and adversarially trained models with local smoothing (M1: mean, M2: median, G: Gaussian). Models with local smoothing show improved stability while retaining sparsity gains. Arrows indicate whether higher or lower values are better.

		FMNIST				**CIFAR-10**				**ImageNette**			
		A	**M1**	**M2**	**G**	**A**	**M1**	**M2**	**G**	**A**	**M1**	**M2**	**G**
VG	**dG ↑**	**0.198**	**0.198**	0.171	0.183	**0.188**	0.185	0.181	0.185	0.050	0.018	0.036	**0.063**
	dRIS ↓	2.193	1.396	**-1.025**	1.168	-0.458	-0.621	**-0.676**	-0.465	-0.056	**-0.121**	-0.016	-0.098
	dROS ↓	2.084	1.121	**-1.222**	0.739	0.217	0.260	**0.214**	0.226	-0.362	**-0.470**	-0.297	-0.456
	dROAD ↑	-0.005	-0.045	-0.050	**0.006**	0.029	0.036	0.035	**0.039**	-0.008	**0.151**	0.014	0.053
IG	**dG ↑**	0.067	**0.075**	0.047	0.050	0.091	0.091	0.092	**0.094**	0.034	0.033	**0.062**	0.041
	dRIS ↓	2.016	2.679	**-0.843**	4.564	-1.056	-1.504	**-1.862**	-1.662	0.143	**-0.0071**	0.135	0.276
	dROS ↓	1.931	2.917	**-0.698**	4.681	0.228	0.350	**-0.123**	-0.090	-0.230	**-0.532**	-0.451	-0.376
	dROAD ↑	**-0.010**	-0.070	-0.029	-0.035	0.071	0.076	**0.092**	0.069	**0.064**	0.045	0.017	0.051
SG	**dG ↑**	**0.198**	**0.198**	0.171	0.183	0.681	0.684	0.684	**0.678**	0.036	0.028	**0.064**	0.035
	dRIS ↓	0.945	0.799	**-0.466**	0.994	-0.040	-0.034	**-0.191**	0.885	0.017	**-0.148**	0.719	0.045
	dROS ↓	5.593	3.418	**-0.194**	2.034	4.619	5.087	**4.393**	4.540	-0.576	**-0.728**	-0.589	-0.657
	dROAD ↑	**0.006**	0.000	-0.018	-0.009	0.003	-0.004	0.012	**0.030**	**0.021**	0.000	0.019	-0.004

Table 1 presents the evaluation of saliency maps produced by Vanilla Gradients (VG), Integrated Gradients (IG), and SmoothGrad (SG) across three datasets. Consistent with prior findings Chalasani et al. (2020), adversarial training (A) generally improves the sparsity of saliency maps ($dG > 0$). Notably, these sparsity gains are largely preserved when local feature-map smoothing is introduced (M1, M2, G), with only minor variations across filter types and datasets.

However, the benefits of adversarial training come with a notable downside in stability: $dRIS$ and $dROS$ tend to increase, indicating greater sensitivity to small input perturbations. Introducing smoothing partially offsets this effect. Median filtering (M2) yields improved stability on FMNIST and CIFAR-10, while mean filtering (M1) is most effective on ImageNette. These results indicate that local smoothing can help regularize intermediate representations during adversarial training, restoring some robustness in saliency outputs.

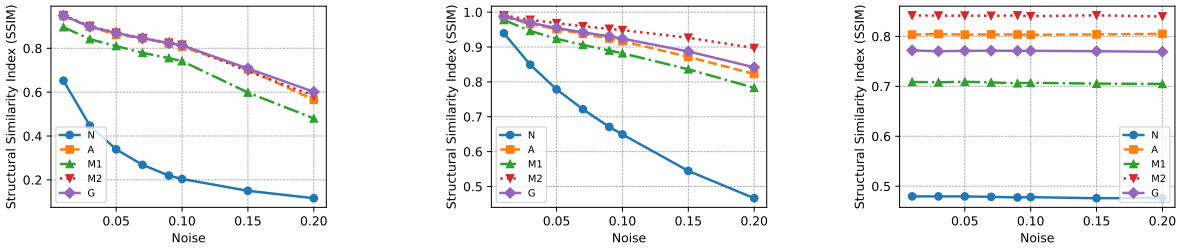
Faithfulness, on the other hand, does not show a consistent pattern of improvement. On FMNIST, most models—including those with smoothing—show minimal or even negative changes in $dROAD$. CIFAR-10 offers stronger gains: e.g., models smoothed with Gaussian filtering (G) show improved faithfulness for VG and SG, while median filtering (M2) performs best for IG. ImageNette results are mixed—VG sees the greatest improvement with mean filter (M1), whereas IG and SG show only marginal changes. These results suggest that enhancements in faithfulness are highly dependent on the dataset and explanation method, and cannot be assumed as a guaranteed outcome of adversarial training and smoothing-based interventions.

Takeaway: Local smoothing filters help recover stability lost to adversarial training while maintaining sparsity. However, there is no universal faithfulness gain. The resulting benefits are method and dataset dependent. We provide the ROAD curves in Appendix D.

4.2.2 On the structural similarity of saliency maps

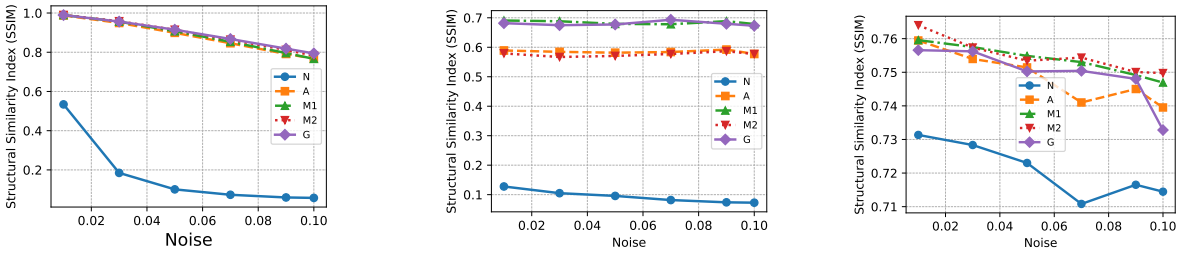
In this section, we evaluate how different training strategies affect the structural stability of saliency maps under input noise. Following Adebayo et al. (2018), we perturb each input image $\mathbf{x}$ with Gaussian noise to generate $\mathbf{x}'$, ensuring the model's prediction remains unchanged. We then compute saliency maps for both inputs and measure their similarity using the Structural Similarity Index (SSIM).

FMNIST: As shown in Figure 4, saliency maps from naturally trained models degrade rapidly in SSIM as noise increases, especially for Vanilla Gradient and Integrated Gradients. In contrast, adversarially trained model, particularly with median (M2) or mean (M1) filter smoothing, exhibit much higher structural consistency. For SmoothGrad, the SSIM curve is flatter across all models due to its inherent noise averaging, but smoothing still yields modest gains.



(a) Vanilla Gradient (b) Integrated Gradients (c) SmoothGrad

Figure 4: Structural similarity evaluation of saliency maps on various FMNIST models: naturally trained (N), adversarially trained (A), and adversarial trained with smoothing filters (M1: mean filter, M2: median filter and G: Gaussian filter). Results demonstrate that naturally trained models consistently yield lower SSIM values, and adversarial training combined with smoothing methods significantly enhances structural consistency.



(a) Vanilla Gradient (b) Integrated Gradients (c) SmoothGrad

Figure 5: Structural similarity evaluation of saliency maps on various CIFAR-10 models: naturally trained (N), adversarially trained (A), and adversarial trained with smoothing filters (M1: mean filter, M2: median filter and G: Gaussian filter). Results demonstrate that naturally trained models consistently yield lower SSIM values, and adversarial training combined with smoothing methods significantly enhances structural consistency.

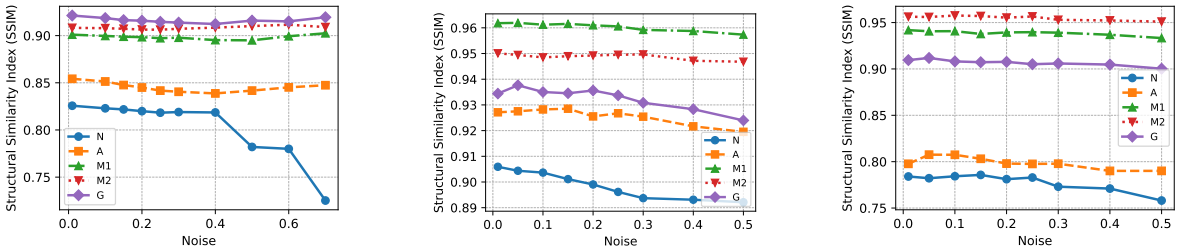
CIFAR-10: The results in Figure 5 show similar trends. Naturally trained models again yield the lowest SSIM, while adversarially trained models with local smoothing maintain higher structural similarity. Median filtering (M2) consistently stabilizes saliency maps across explanation methods. For SmoothGrad, although SSIM values remain high overall, models trained with smoothing still outperform the baseline, suggesting that local feature-map smoothing adds value even for inherently robust methods.

ImageNette: In Figure 6, the drop in SSIM for naturally trained models is more pronounced under Vanilla and Integrated Gradients. Models trained with Gaussian (G) or mean (M1) filtering achieve the most consistent performance across all noise levels. As in other datasets, SmoothGrad yields relatively stable SSIM, but smoothing further boosts its structural alignment.

Takeaway: Across all datasets and explanation methods, local feature-map smoothing improves the structural consistency of saliency maps when used with adversarial training, especially in CIFAR-10 and ImageNette datasets.

4.2.3 Trade-off between model performance & saliency map quality

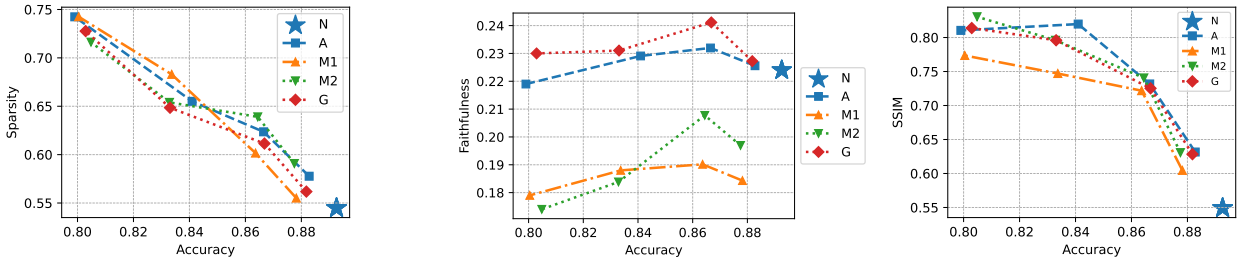
Our findings in Section 4.2.1 and Section 4.2.2 reveal that: a) input-gradient based attribution methods produce sparse saliency maps in adversarially trained models, b) gain in sparsity with adversarially trained models result in compromise of stability, c) adversarially trained models, with local-feature-map smoothing, enhances the stability of saliency maps without significantly compromising on sparsity, d) saliency maps in adversarially trained models with feature map smoothing consistently demonstrate invariance to noise, and



(a) Vanilla Gradient (b) Integrated Gradients (c) SmoothGrad

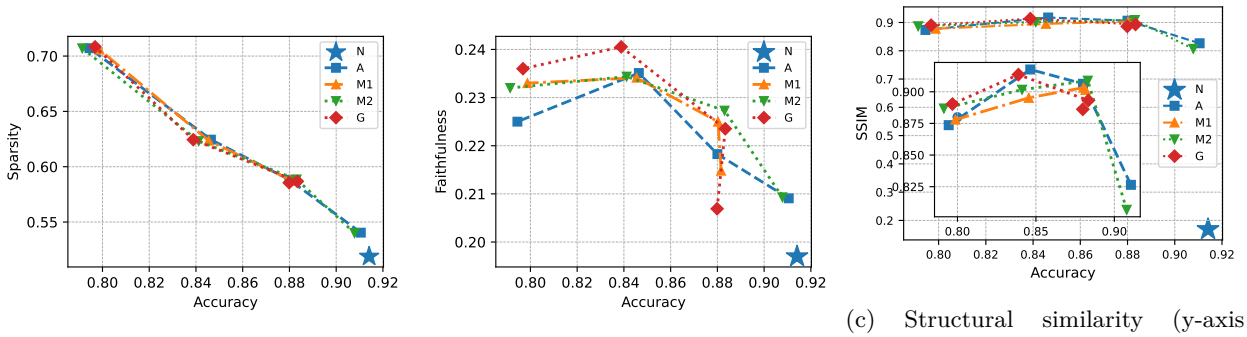
Figure 6: Structural similarity evaluation of saliency maps on various ImageNette models: naturally trained (N), adversarially trained (A), and adversarial trained with smoothing filters (M1: mean filter, M2: median filter and G: Gaussian filter). Results demonstrate that naturally trained models consistently yield lower SSIM values, and adversarial training combined with smoothing methods significantly enhances structural consistency.

e) faithfulness gains are dataset-dependent as we observed that on FMNIST, natural models often remain competitive, while on CIFAR-10, adversarially trained models with smoothing consistently outperform (See Table 4.2.1)).



(a) Sparsity (b) Faithfulness (c) Structural similarity

Figure 7: Tradeoff between saliency map quality and model performance on FMNIST models: naturally trained (N), adversarially trained (A), and adversarially trained with local filter smoothing (M1: mean filter, M2: median-filter, G: Gaussian-filter. Results show that adversarially trained models (with smoothing filters) improve saliency map quality but at the expense of benign accuracy.



(a) Sparsity (b) Faithfulness (c) Structural similarity (y-axis zoomed)

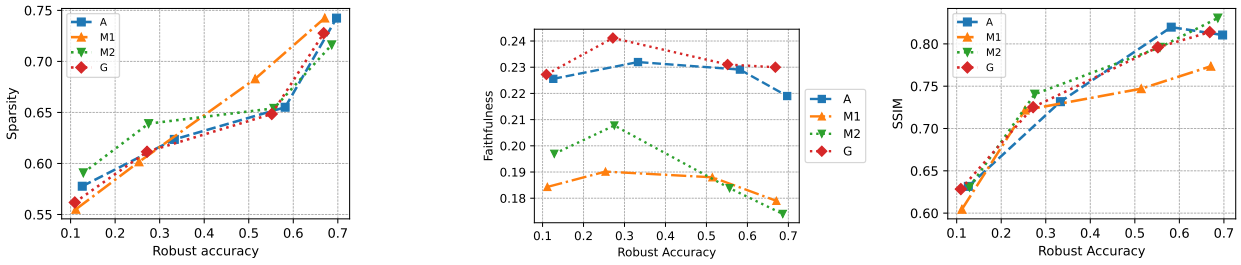
Figure 8: Tradeoff between saliency map quality and model performance on CIFAR-10 models: naturally trained (N), adversarially trained (A), and adversarially trained with local filter smoothing (M1: mean filter, M2: median-filter, G: Gaussian-filter. Results show that adversarially trained models (with smoothing filters) improve saliency map quality but at the expense of benign accuracy.

However, it's important to note a caveat of adversarially trained models: *they come at the expense of benign accuracy.* We illustrate this tradeoff in Figure 7 and Figure 8. We train $L_\infty(\epsilon)$ robust FMNIST and CIFAR-10 models with perturbation strength $\epsilon \in [0.01, 0.03, 0.06, 0.1]$ for adversarial training (A), adversarial training with smoothing filters of mean (M1), median (M2), and, Gaussian (G). For each model, we compute its benign accuracy, and three saliency map characteristics using Vanilla Gradient: sparsity (Chalasani et al., 2020), area over perturbation curve of ROAD (Rong et al., 2022) for faithfulness, and structural similarity (Adebayo et al., 2018). Then, we plot the saliency map characteristics against the benign accuracy of the model.

In FMNIST (Figure 7a), stronger adversarial training increases sparsity, but also decreases benign accuracy. However, the faithfulness trend (Figure 7b) is less consistent—some robust models (e.g., with Gaussian smoothing) perform better, but others do not improve over the naturally trained baseline. On CIFAR-10 (Figure 8b), the improvement is clearer. Models with local smoothing, especially Gaussian and median filters, achieve higher faithfulness than natural models, particularly at intermediate accuracy levels. This supports the claim that robust models can yield more faithful explanations when carefully regularized. Structural similarity (Figures 7c and 8c) increases with adversarial robustness, indicating more noise-tolerant explanations. Smoothing amplifies this effect, yielding stable explanations even under input perturbations.

4.2.4 Relationship between model robustness & saliency map quality

In Section 4.2.3, we observed that increasing adversarial training strength (ϵ) reduces benign accuracy but improves various saliency map properties. Here, we explore how these improvements correlate with robust accuracy—i.e., accuracy on adversarially perturbed inputs.



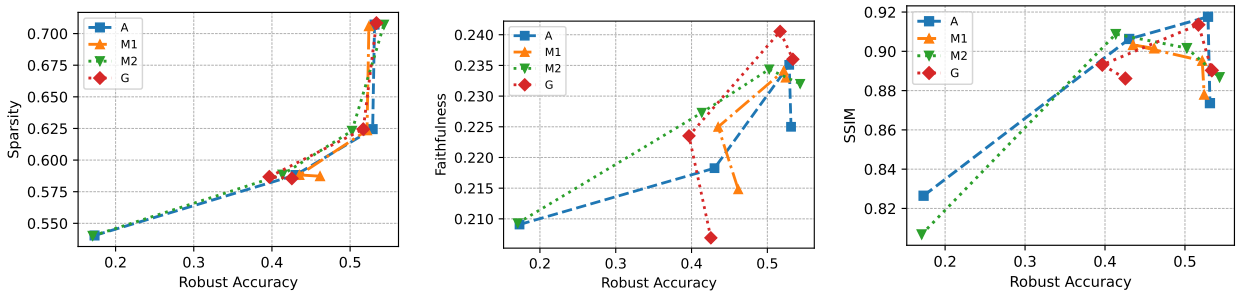
(a) Sparsity (b) Faithfulness (c) Structural similarity

Figure 9: Relationship between model robustness and saliency map quality on FMNIST models: naturally trained (N), adversarially trained (A), and adversarially trained with local filter smoothing (M1: mean filter, M2: median-filter, G: Gaussian-filterResults show that increasing robustness of adversarially trained models (with smoothing filters) improves saliency map quality.

For each model trained with L_∞ adversarial training at $\epsilon \in \{0.01, 0.03, 0.06, 0.1\}$, we compute robust accuracy using PGD attacks ($\epsilon = 0.1$, 100 steps) (Madry et al., 2018). We then measure three saliency map metrics: sparsity (Chalasani et al., 2020), area over perturbation curve of ROAD (Rong et al., 2022) for faithfulness, and structural similarity (Adebayo et al., 2018), and plot them against robust accuracy in Figures 9 and 10.

On FMNIST, we observe that sparsity increases consistently with robust accuracy (Figure 9a), supporting the notion that robust models emphasize more localized and discriminative features. Faithfulness (Figure 9b) peaks at moderate robustness levels (e.g., with Gaussian smoothing), but then slightly declines. This suggests that some robustness enhances attribution alignment, but excessive robustness may lead to misaligned saliency. Structural similarity (Figure 9c) improves steadily with robustness, indicating that robust models yield explanations that are more invariant to input noise.

Figure 10a shows a strong positive correlation between robustness and sparsity in CIFAR-10, mirroring FMNIST. Faithfulness (Figure 10b) increases with robustness, especially when local smoothing (M2) is applied. While some smoothing configurations (e.g., M1) show irregular fluctuations, the general trend



(a) Sparsity (b) Faithfulness (c) Structural similarity

Figure 10: Relationship between model robustness and saliency map quality on CIFAR-10 models: naturally trained (N), adversarially trained (A), and adversarially trained with local filter smoothing (M1: mean filter, M2: median-filter, G: Gaussian-filter

suggests that faithfulness improves with robustness. As shown in Figure 10c, structural similarity improves with robustness across all model variants, with smoothing further amplifying this effect.

5 Ablation study: Receptive field expansion

To isolate the role of receptive field expansion from smoothing, we perform an ablation study by modifying the feature-map smoothing block to include only a convolution operation without any smoothing filter. Specifically, we evaluate two variants: initialized with an identity and random convolution. These configurations expand the receptive field without reducing local activation noise, providing a clean baseline for assessing the effect of receptive field growth in adversarially trained models.

Table 2: Effect of receptive field expansion on Vanilla Gradient (VG) explanations in CIFAR-10. M2 means adversarial training with median filter; Identity and Random include only 1×1 convolutions without filtering operations. ↑ and ↓ means higher and lower is better.

Models	**M2**	**Identity**	**Random**
Sparsity (dG) (↑)	0.18	0.16	0.15
Relative input stability (dRIS) (↓)	-0.68	-0.41	-0.36
Relative output stability (dROS) (↓)	0.21	0.07	0.06

We evaluate Vanilla Gradient (VG) saliency maps from CIFAR-10 models trained under three settings: a) adversarial training with median filter (M2), b) adversarial training with a smoothing block consisting of identity initialized convolution but no filter (identity), and c) adversarial training with a smoothing block consisting of randomly initialized convolution but no filter (random). Table 2 summarizes the results, indicating that all models achieve comparable sparsity gains highlighting that adversarial training primarily contributes to the sparsity gain. The M2 model achieves the best input stability (lowest dRIS), highlighting the role of smoothing in mitigating attribution noise from small input perturbations. However, M2 model performs worse than the convolution-only baselines in output stability (dROS), suggesting that while smoothing helps in stabilizing attributions with respect to inputs, it might not directly translate to stability at the model's output layer.

These results suggest that smoothing primarily improves saliency map stability at the input level, whereas receptive field expansion aids output-level stability. Combining both with the adversarial training improves explanation quality holistically.

6 Qualitative Analysis

Our quantitative analyses showed that adversarially trained models produce sparser saliency maps, though often at the cost of stability. Incorporating local feature-map smoothing helps restore stability while preserving sparsity. To complement these findings, we now investigate how these differences manifest in practice—i.e., how end-users perceive and understand the resulting saliency maps.

Motivation. While metrics such as sparsity and faithfulness provide useful proxies, saliency maps are ultimately intended for human interpretation. A faithful yet overly noisy or, sparse explanation may fail to assist users in real-world decision-making Gilpin et al. (2018). While prior works (Nguyen et al., 2021; Kim et al., 2022; Adebayo et al., 2020) focus on qualitative evaluation for utility of explanations, we conduct a survey to measure comprehensibility of saliency maps.

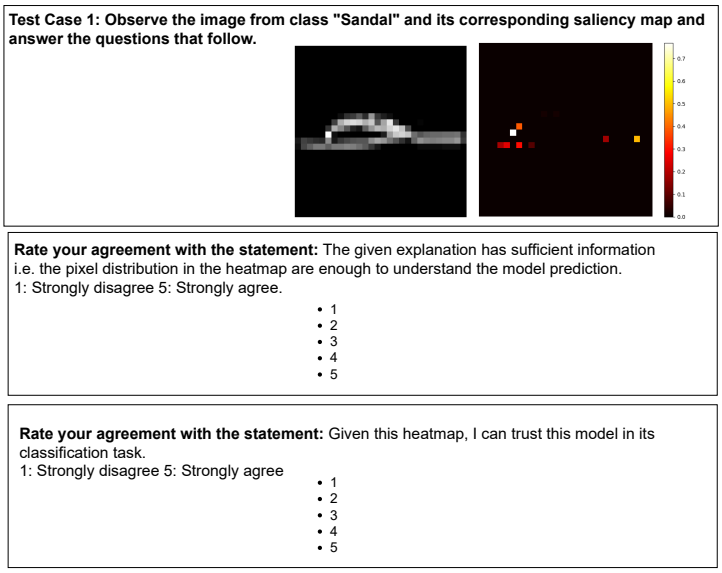


Figure 11: A sample of question from the survey.

Survey Methodology. We conducted a survey with 65 graduate students (Ph.D. and M.S.), each with at least one year of experience in computer vision.[1] The objective was to determine whether the information conveyed by saliency maps was sufficient for understanding the underlying model behavior. Participants were presented with saliency maps using Vanilla Gradient (Simonyan et al., 2014) for 10 randomly selected FMNIST and CIFAR-10 images, each generated by three models: a naturally trained (N), an adversarially trained (A), and an adversarially trained with median-filter smoothing (M2). The 60 resulting image-saliency pairs were shown in randomized order, without disclosing model identity.

Each participant rated the explanations based on two questions from the Hoffman satisfaction scale (Hoffman et al., 2023): "Does the explanation provide sufficient information?" (Sufficiency). "Do you trust the model's classification based on this explanation?" (Trust). Responses were on a 5-point Likert scale (1 = strongly disagree, 5 = strongly agree) (see Figure 11). Finally, participants compared saliency maps from all three models side-by-side (Figure 12) and selected the most comprehensible one, providing free-text justifications.

Results and Analysis. The feature-map smoothed model (M2) consistently outperformed other models in both sufficiency (3.33 ± 1.03) and trust (3.14 ± 1.01). The adversarial model (A) also showed improvement (sufficiency: 2.99 ± 0.93, trust: 3.08 ± 0.90) over the naturally trained model (N), which scored lowest (sufficiency: 2.08 ± 0.75, trust: 2.02 ± 0.82). These results suggest that while adversarial training improves explanation quality, smoothing enhances perceived clarity and reliability even further. Participants appre-

[1] An Institutional Review Board (IRB) approval was granted by our institution prior to interviewing human subjects for our qualitative study.

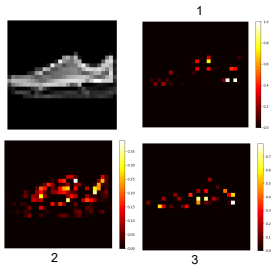


Figure 12: Sample images in the survey.

Table 3: Wilcoxon and ANOVA test results on the survey for naturally trained model (N), adversarially trained model (A) and adversarially trained with median filter (M2).

	Wilcoxon (p-value)			one-way ANOVA	
	N vs A	**A vs M2**	**N vs M2**	**F-stat**	**p value**
Sufficiency	9.79E-41	4.26E-14	3.71E-27	200.38	7.82E-72
Trust	5.56E-39	3.24E-11	3.89E-24	193.86	6.58E-70

ciated the reduction in noise and highlighted the clarity and relevance of the explanations. When asked to choose the most comprehensible map, 56% of participants preferred M2, citing reduced noise and better alignment with object features (e.g., "highlights important features without excessive detail"). 29% preferred A, and only 15% chose N, often noting its maps were "too noisy" or "distracting."

To validate these findings, we performed Wilcoxon signed-rank tests and one-way ANOVA on the scores. As reported in Table 3, the results are statistically significant ($p < 10^{-10}$), confirming that differences in comprehensibility across model types are robust and not due to chance.

7 Limitations

While our work provides actionable insights into improving saliency map comprehensibility through adversarial training and local feature-map smoothing, several limitations must be acknowledged. First, this study focuses exclusively on input-gradient-based explanation techniques. While we employ widely used and representative gradient-based methods, they do not cover other methods such as perturbation-based methods (e.g., LIME (Ribeiro et al., 2016)). Our experiments are also limited to mid-scale datasets: FMNIST, CIFAR-10, and ImageNette. While these settings are common in interpretability studies, they may not reflect the behavior of larger-scale datasets (e.g., full ImageNet). Prior work (Zhang et al., 2019) has shown that adversarial training at scale is challenging, and our methods may encounter similar issues with robustness-accuracy trade-offs in such settings. Lastly, although we explore a range of local feature-map smoothing filters (mean, median, Gaussian), the choice of filter type is currently empirical and dataset-method specific.

8 Conclusion

We explored how adversarial training affects the interpretability of saliency maps produced by input-gradient-based explanation methods. While adversarial training improves explanation sparsity, it often harms stability. To address this, we introduced a lightweight intervention: applying local feature-map smoothing during training. Our experiments demonstrate that this approach improves stability across multiple datasets and explanation methods, without significantly compromising sparsity. However, gains in faithfulness are inconsistent and depend on both the dataset and attribution method used. These findings suggest that while model-level changes can enhance some aspects of interpretability, their effects are not universally beneficial across all explanation dimensions. Overall, our work promotes a complementary paradigm to explanation method design showing that training-time and architectural adjustments can help shape model representations toward better post-hoc interpretability.

References

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, 2018.

Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. *arXiv preprint arXiv:2011.05429*, 2020.

Chirag Agarwal, Nari Johnson, Martin Pawelczyk, Satyapriya Krishna, Eshika Saxena, Marinka Zitnik, and Himabindu Lakkaraju. Rethinking stability for attribution-based explanations. In *ICLR 2022 Workshop on PAIR {\textasciicircum} 2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2022.

David Alvarez-Melis and Tommi S Jaakkola. On the robustness of interpretability methods. *ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*, 2018.

Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1): 108–116, 1995.

Prasad Chalasani, Jiefeng Chen, Amrita Roy Chowdhury, Xi Wu, and Somesh Jha. Concise explanations of neural networks using adversarial training. In *International Conference on Machine Learning*, pp. 1383–1391. PMLR, 2020.

ZHAO Chenyang and Antoni B Chan. Odam: Gradient-based instance-specific visual explanations for object detection. In *The Eleventh International Conference on Learning Representations*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. Ieee, 2009.

Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. *Advances in Neural Information Processing Systems*, 32, 2019.

Christian Etmann, Sebastian Lunz, Peter Maass, and Carola Schoenlieb. On the connection between adversarial robustness and saliency map interpretability. In *International Conference on Machine Learning*, pp. 1823–1832. PMLR, 2019.

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

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6572`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Anna Hedström, Leander Weber, Daniel Krakowczyk, Dilyara Bareeva, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina Marina M.-C. Höhne. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *Journal of Machine Learning Research*, 24(34): 1–11, 2023. URL `http://jmlr.org/papers/v24/22-0142.html`.

Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. Measures for explainable ai: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-ai performance. *Frontiers in Computer Science*, 5:1096257, 2023.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Jeremy Howard. A smaller subset of 10 easily classified classes from imagenet, and a little more french, 2020. *URL https://github. com/fastai/imagenette*, 2020.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in Neural Information Processing Systems*, 32, 2019.

Beomsu Kim, Junghoon Seo, Seunghyeon Jeon, Jamyoung Koo, Jeongyeol Choe, and Taegyun Jeon. Why are saliency maps noisy? cause of and solution to noisy saliency maps. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 4149–4157. IEEE, 2019.

Sunnie SY Kim, Nicole Meister, Vikram V Ramaswamy, Ruth Fong, and Olga Russakovsky. Hive: Evaluating the human interpretability of visual explanations. In *European Conference on Computer Vision*, pp. 280–298. Springer, 2022.

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. *Explainable AI: Interpreting, explaining and visualizing deep learning*, pp. 267–280, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Puneet Mangla, Vedant Singh, and Vineeth N Balasubramanian. On saliency maps and adversarial robustness. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 272–288, 2020.

Giang Nguyen, Daeyoung Kim, and Anh Nguyen. The effectiveness of feature attribution methods and its correlation with automatic evaluation scores. *Advances in Neural Information Processing Systems*, 34: 26422–26436, 2021.

Weili Nie, Yang Zhang, and Ankit Patel. A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. In *International Conference on Machine Learning*, pp. 3809–3818. PMLR, 2018.

Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: randomized input sampling for explanation of black-box models. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, pp. 151. BMVA Press, 2018. URL http://bmvc2018.org/contents/papers/1064.pdf.

Edgar Riba, Dmytro Mishkin, Jian Shi, Daniel Ponsa, Francesc Moreno-Noguer, and Gary Bradski. A survey on kornia: an open source differentiable computer vision library for pytorch. *arXiv preprint arXiv:2009.10521*, 2020.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *22nd ACM SIGKDD*, 2016.

Yao Rong, Tobias Leemann, Vadim Borisov, Gjergji Kasneci, and Enkelejda Kasneci. A consistent and efficient evaluation strategy for attribution methods. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 18770–18795. PMLR, 2022.

Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673, 2016.

Harshay Shah, Prateek Jain, and Praneeth Netrapalli. Do input gradients highlight discriminative features? *Advances in Neural Information Processing Systems*, 34:2046–2059, 2021.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.1556.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6034.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017. URL http://arxiv.org/abs/1706.03825.

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

Thomas Tanay and Lewis D. Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *CoRR*, abs/1608.07690, 2016. URL http://arxiv.org/abs/1608.07690.

Matthew Robert Wicker, Juyeon Heo, Luca Costabello, and Adrian Weller. Robust explanation constraints for neural networks. In *The Eleventh International Conference on Learning Representations*, 2023.

Less Wright. Ranger - a synergistic optimizer. https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer, 2019.

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

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 501–509, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Procedings of the British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, Inderjit Dhillon, and Cho Jui Hsieh. The limitations of adversarial training and the blind-spot attack. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

Tianyuan Zhang and Zhanxing Zhu. Interpreting adversarially trained convolutional neural networks. In *International Conference on Machine Learning*, pp. 7502–7511. PMLR, 2019.

Appendix

Table of Contents

A Dataset and training

FMNIST (Xiao et al., 2017): The Fashion MNIST dataset consists of 28x28 pixel grayscale images of different clothing items and accessories. It contains a total of 70,000 images, divided into a training set of 60,000 examples and a test set of 10,000 examples. Similar to Chalasani et al. (2020), we train a neural network consisting of two convolutional layers with 32 and 64 filters, respectively, each followed by 2x2 max-pooling and a fully connected layer of 1024. We use the Adam optimizer with a learning rate of 0.001, a batch size of 32 and 50 training epochs.

CIFAR-10 (Krizhevsky et al., 2009): CIFAR-10 consists of 60,000 32x32 pixel color images, with each image belonging to one of ten different classes: airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Similar to Chalasani et al. (2020), we use a wide Residual Network (Zagoruyko & Komodakis, 2016) for training CIFAR-10 with the following hyperparameter settings: batch size=128, momentum optimizer with momentum = 0.9, and weight decay = 5e-4, training steps = 70000. We use an adaptive learning rate where the learning rate is set to 0.1 for the first 40000 steps, 0.01 for 40000-50000 steps, and 0.001 for the remaining steps. The wide residual network is trained with 28 layers and widen factor of 10.

ImageNette (Howard, 2020): ImageNette is a 10-class subset of ImageNet (Deng et al., 2009) with 9469 training images and 3925 test images. We use the 320-pixel resolution images (for the shortest side) and randomly resize and crop them to 224x224 pixels during training. We use the standard ResNet-18 model architecture (He et al., 2016) and Ranger optimizer (Wright, 2019) with an initial learning rate of 8e-03 and epsilon 1e-6. We train the models from scratch for 200 epochs and employ the early stopping criterion to select the best-performing model for evaluation.

A.1 Adversarial training

Adversarial training (Goodfellow et al., 2015) involves training a model in the presence of adversarial examples. Adversarial examples are inputs specifically designed to mislead or deceive the model, causing the model to make incorrect predictions. The goal of adversarial training is to improve the robustness and generalization of a model against such perturbed examples. To perform adversarial training, we generate adversarial examples that are produced from natural samples $\mathbf{x} \in R^d$ by adding a perturbation vector $\delta \in R^d$. The perturbation vector differs based on the type of attack employed. We use the PGD (Madry et al., 2018) attack to obtain adversarial perturbations. PGD is an iterative attack where the perturbation is computed multiple times with small steps. The hyper-parameters of PGD attack in our adversarial training: for FMNIST and CIFAR-10, $\epsilon = 0.1$, attack step size = $\epsilon/10$, and number of iterations = 40; for ImageNette $\epsilon = 1/255$, step size = 0.00784 and number of iterations = 20. Other training hyperparameters are kept as explained in Appendix A.

B Smoothing filters

A generic convolutional neural network with a feature map smoothing block is presented in Figure 13. The smoothing block consists of local filtering operations. We use with the following filters in our study:

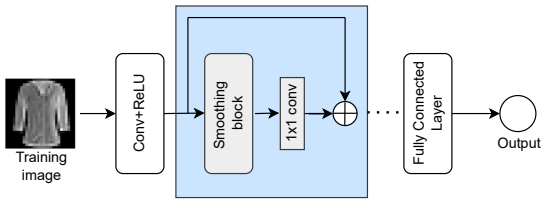


Figure 13: A generic convolutional neural network with a feature-map smoothing block.

- **Mean filter:** A mean filter replaces each feature with the average of nearby features within a defined kernel. For an input feature map (I) of size HxW and a K-sized kernel, the output feature map $O(u, v)$ is calculated using Eqn. 9:

$$O(u,v) = \frac{1}{K^2} \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} I(u+i, v+j) \tag{9}$$

  Here, u and v represent spatial coordinates in the output feature map, ranging from 0 to $H-K$ and 0 to $W-K$ respectively. $I(u+i, v+j)$ denotes the feature value at spatial location $(u+i, v+j)$ in the input feature map. This operation is applied independently to each channel of the input.

- **Median filter:** A median filter computes the median value within a small sliding window over the feature map, given by Eqn. 10. Given an input feature map I and a median filter window size K, the output feature map $O(u, v)$ is computed using Eqn. 10:

$$O(u,v) = median(I(u - \frac{K}{2} : u + \frac{K}{2}, v - \frac{K}{2} : v + \frac{K}{2}) \tag{10}$$

  Here, $I(u - \frac{K}{2} : u + \frac{K}{2}, v - \frac{K}{2} : v + \frac{K}{2})$ represents the subset of the input feature around (u, v) with a size of KxK. This operation is applied independently to each channel of the input feature map.

- **Gaussian filter:** A Gaussian filter applies a smoothing effect to feature maps by convolving them with a Gaussian kernel, effectively reducing Gaussian noise. The degree of smoothing can be adjusted by modifying the standard deviation (σ) of the Gaussian kernel. Given an input feature map I and a Gaussian filter kernel K, the output feature map $O(u, v)$ is calculated with Eqn. 11:

$$O(u,v) = (I * K)(u,v) \tag{11}$$

  Here, $*$ denotes 2D convolution. The Gaussian kernel K is generated using a Gaussian function with a specific standard deviation σ, defined in Eqn. 12:

$$K(u,v) = \frac{1}{2\pi\sigma^2} e^{(-\frac{u^2+v^2}{2\sigma^2})} \tag{12}$$

  This operation is independently applied to each channel of the input feature map.

Implementation: We utilize the differentiable local filters available in Kornia (Riba et al., 2020). We use a 3x3 Kernel for mean, median, and Gaussian filtering. The standard deviation of the kernel for Gaussian filtering was computed as $(0.3 * ((\mathbf{x}.shape[3] - 1) * 0.5 - 1) + 0.8, 0.3 * ((\mathbf{x}.shape[2] - 1) * 0.5 - 1) + 0.8)$ where $\mathbf{x}$ is the input image.

C Effect of smoothing filter on model performance

In Table 4, we present the results of various models on FMNIST, CIFAR-10 and ImageNette, with both natural (benign) and adversarial (robust) accuracy. Benign accuracy measures the model performance on benign (clean) test set, whereas robust accuracy evaluates how well the models detect adversarially perturbed samples. The robust models under evaluation are trained at $\epsilon = 0.1$ for FMNIST and CIFAR-10 and $\epsilon = 1/255$ for ImageNette. Robust accuracy evaluation is performed on a test-set consisting of adversarial samples created using PGD attack (Madry et al., 2018) at $\epsilon = 0.1$ L_∞ perturbation bound.

Table 4: Natural and Robust Accuracy of Various FMNIST, CIFAR-10, and ImageNette models: naturally trained (N), adversarially trained (A), natural training with mean-filter smoothing (NM1), adversarial training with mean-filter smoothing (M1), natural training with median-filter smoothing (NM2), adversarial training with median-filter smoothing (M2), natural training with Gaussian-filter smoothing (NG), adversarial training with Gaussian-filter smoothing (G).

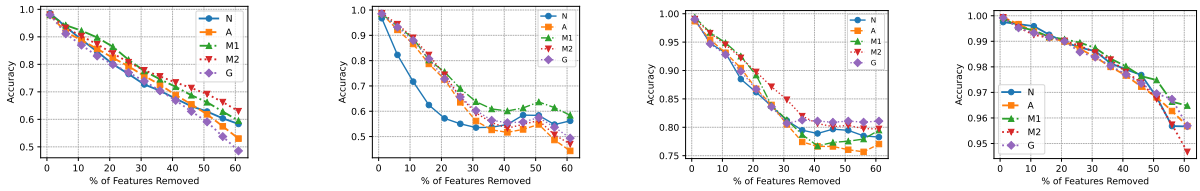
Dataset	**Models/Accuracy**	**N**	**A**	**NM1**	**M1**	**NM2**	**M2**	**NG**	**G**
FMNIST	**Benign Accuracy**	89.9	79.9	88.4	80.0	88.8	80.5	89.1	80.3
	Robust Accuracy	9.5	67.7	8.5	67.1	8.2	68.6	6.9	66.8
CIFAR-10	**Benign Accuracy**	90.9	80.5	89.7	79.6	88.6	80.1	90.2	80.8
	Robust Accuracy	4.8	54.3	4.5	51.2	4.7	56.3	6.8	53.9
ImageNette	**Benign Accuracy**	96.3	70.8	93.3	58.8	90.9	55.3	95.5	51.6
	Robust Accuracy	1.6	12.2	1.2	6.5	2.3	14.3	3.7	13.5

Across all datasets, applying smoothing filters alone did not result in significant changes in natural or robust accuracy ($\approx \pm 3\%$). The smoothing filters, when used without adversarial training, did not drastically improve robustness or reduce natural accuracy, indicating that their primary role may be in stabilizing feature maps without dramatically altering decision boundaries.

However, when smoothing filters were combined with adversarial training, robust accuracy improved for some filters, particularly in FMNIST and CIFAR-10, where models trained with adversarial samples and smoothing exhibited stronger defense against adversarial attacks. On the ImageNette dataset, we observed a notable drop in benign accuracy when smoothing filters were applied during adversarial training.

D ROAD plots

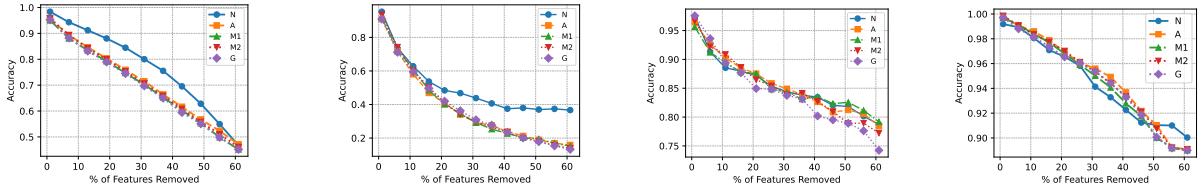
In this section, we plot the individual ROAD plots for each explanation method and discuss the faithfulness. We also include a random gradient (randomly sampled saliency map) for comparison of explanation faithfulness with Vanilla Gradient (VG), Integrated Gradients (IG) and SmoothGrad (SG).



(a) Vanilla Gradient (b) Integrated Gradients (c) SmoothGrad (d) Random Gradient

Figure 14: ROAD evaluation for measuring saliency map faithfulness where sharper drop in accuracy is better on various FMNIST models: naturally trained (N), adversarially trained (A), and adversarially trained models with local smoothing (M1: mean-filter, M2: median-filter, G: Gaussian-filter.

FMNIST: In Table 1, we observed that across all explanation methods, adversarial training, with or without smoothing, shows minimal or negative impact on faithfulness compared to naturally trained models. For VG and IG, we can observe in Figures 14a and 14b that naturally trained models show sharper drops in accuracy under feature removal, suggesting more faithful attributions. SmoothGrad (SG), in Figure 14c, exhibits similar behavior, with only model A yielding the sharpest drop. Figure 14d, presents the ROAD curve for the baseline Random Gradient, shows the lowest drop in accuracy, validating the reliability of the ROAD evaluation framework.

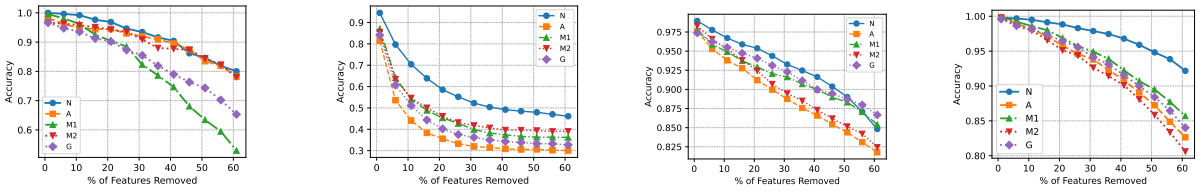


(a) Vanilla Gradient (b) Integrated Gradients (c) SmoothGrad (d) Random Gradient

Figure 15: ROAD evaluation for measuring saliency map faithfulness where sharper drop in accuracy is better on various CIFAR10 models: naturally trained (N), adversarially trained (A), and adversarially trained models with local smoothing (M1: mean-filter, M2: median-filter, G: Gaussian-filter

CIFAR-10: Unlike FMNIST, all adversarially trained models on CIFAR-10, especially those with smoothing, show improved attribution faithfulness (see Table 1). For VG(Figure 15a) and IG (Figure 15b), naturally trained model has the least steep accuracy drop. For SG, the adversarially trained with Gaussian filter (G) shows the steepest drop (Figure 15c). The Random Gradient baseline in Figure 15d shows minimal accuracy drop, reinforcing the validity of the ROAD evaluation metric.

ImageNette: Faithfulness improvements on ImageNette are most pronounced for VG and IG (see Table 1). For VG, the adversarially trained model with mean filter (M1) achieves the highest dROAD score and steepest accuracy drop under perturbation (Figure 16a). IG similarly shows better performance in adversarially trained model (A) and adversarially trained with Gaussian filter (G) (Figure 16b), surpassing the naturally trained baseline. In contrast, SG exhibits limited gains: only adversarially trained model (A) outperforms the baseline marginally (Figure 16c).



(a) Vanilla Gradient (b) Integrated Gradients (c) SmoothGrad (d) Random Gradient

Figure 16: ROAD evaluation for measuring saliency map faithfulness where sharper drop in accuracy is better on various ImageNette models: naturally trained (N), adversarially trained (A),and adversarially trained models with local smoothing (M1: mean-filter, M2: median-filter, G: Gaussian-filter.

E Ablation study: Position of smoothing block

We study the effect of smoothing block placement within the network on the sparsity and stability of saliency maps using Vanilla Gradient. Specifically, we compare three configurations for CIFAR-10, where the smoothing block is inserted after the (i) first, (ii) second, and (iii) third residual block. Results for insertion after the second and third blocks are presented in Tables 5 and 6, while results for insertion after the first block are already shown in Table 1.

Table 5: Sparsity and Stability evaluation, when smoothing block is placed after **second** residual block, on various adversarially trained CIFAR-10 models, *M1: Mean-filter, M2: Median-filter, G: Gaussian-filter.* ↑ and ↓ indicate higher and lower is better.

	CIFAR-10		
	M1	**M2**	**G**
dG (↑)	0.178	0.185	0.176
dRIS (↓)	-0.605	-0.663	-0.477
dROS (↓)	0.268	0.225	0.239
dRRS (↓)	0.464	0.445	0.462

Table 6: Sparsity and Stability evaluation, when smoothing block is placed after **third** residual block, on various adversarially trained CIFAR-10 models, *M1: Mean-filter, M2: Median-filter, G: Gaussian-filter.* ↑ and ↓ indicate higher and lower is better.

	CIFAR-10		
	M1	**M2**	**G**
dG (↑)	0.185	0.180	0.187
dRIS (↓)	-0.599	-0.670	-0.470
dROS (↓)	0.271	0.221	0.235
dRRS (↓)	0.470	0.429	0.468

Across all configurations, the sparsity gain (dG) remains consistent, ranging from 0.176 to 0.188. Notably, inserting the smoothing block after the third residual block yields slightly higher sparsity scores. However, the best stability, both in terms of input and output (dRIS and dROS), is achieved when the smoothing is applied earlier, particularly after the first residual block. These results suggest that applying smoothing earlier in the network leads to greater stability in gradient-based saliency maps, likely due to the regularization of lower-level feature representations. In contrast, placing the smoothing block deeper in the network modestly enhances sparsity but offers diminished gains in stability. Therefore, to balance both sparsity and stability, we adopt the design choice of placing the smoothing block after the first residual block in our main experiments.

F Additional network: VGG-16

To assess the generalizability of our findings, we extend our evaluation to a VGG-16 network (Simonyan & Zisserman, 2015) trained on CIFAR-10.

Training setup: We train VGG-16 using SGD with momentum, a learning rate of 0.1 (decayed by a factor of 0.1 every 30 epochs), weight decay of 5e-4, and batch normalization. For adversarial training, we apply PGD with $\epsilon = 0.1$. Feature-map smoothing filters (mean, median, and Gaussian) are integrated after the first convolutional block, following the setup in our main experiments (see Appendix B). Similar to previous sections, we train following models for this network: naturally-trained (N), adversarially-trained (A), adversarial training with mean-filter smoothing (M1), adversarial training with median-filter smoothing (M2), and adversarial training with Gaussian-filter smoothing (G).

Table 7: Sparsity and Stability Evaluations for Vanilla Gradient (VG), Integrated Gradient (IG), and SmoothGrad (SG) on various VGG-16 models: adversarially-trained (A), adversarial training with mean-filter smoothing (M1), adversarial training with median-filter smoothing (M2), adversarial training with Gaussian-filter smoothing (G). Here, ↑ and ↓ indicate higher and lower values are better.

	Vanilla Gradient (VG)				**Integrated Gradients (IG)**				**SmoothGrad (SG)**			
	A	**M1**	**M2**	**G**	**A**	**M1**	**M2**	**G**	**A**	**M1**	**M2**	**G**
dG ↑	0.10	0.10	0.10	0.10	0.02	0.03	0.02	0.01	0.08	0.08	0.08	0.08
dRIS ↓	-0.30	-0.40	-0.35	-0.39	-0.29	-0.62	-0.74	-0.60	-0.33	-0.36	-0.46	-0.10
dROS ↓	-0.24	-0.31	-0.26	-0.30	-0.13	-0.22	-0.52	-0.24	-0.42	-0.50	-0.49	-0.40

Evaluation: For each model, we compute saliency maps using Vanilla Gradient (VG), Integrated Gradients (IG), and SmoothGrad (SG). We then evaluate sparsity (dG) (Chalasani et al., 2020), relative input stability ($dRIS$) (Agarwal et al., 2022) and relative output stability ($dROS$) (Agarwal et al., 2022). All metrics are reported relative to the naturally trained model (N), following Chalasani et al. (2020).

$$dG[\phi(\mathbf{x})] = G^M[\phi(\mathbf{x})] - G^N[\phi(\mathbf{x})] \tag{13}$$

$$dRIS[\phi(\mathbf{x})] = RIS^M[\phi(\mathbf{x})] - RIS^N[\phi(\mathbf{x})] \tag{14}$$

$$dROS[\phi(\mathbf{x})] = ROS^M[\phi(\mathbf{x})] - ROS^N[\phi(\mathbf{x})] \tag{15}$$

$$\tag{16}$$

Results: Table 7 summarizes the results. All explanation methods consistently show positive dG values, confirming that adversarial training improves sparsity in VGG-based models as well. However, the sparsity gains are relatively stable across the A, M1, M2, and G models, suggesting that the smoothing filter type has limited impact on sparsity in VGG. In contrast, smoothing has a clear benefit for stability. Both $dRIS$ and $dROS$ improve significantly for models using smoothing filters, particularly for IG and SG. Median filtering (M2) provides the best input stability across methods, while all smoothing variants outperform the adversarial baseline (A).

G Conditions affecting the tightness of stability bounds

The stability bounds presented in Section 3.1 serve as indicators of the relationship between model sensitivity and attribution stability. However, these bounds are inherently approximate and depend on several factors. For example, the nonlinearity of the model, particularly the choice of activation function H, might influence the bounds' tightness. For activation functions with bounded gradients, such as sigmoid or tanh, the change in $H'(\langle \mathbf{w}, \mathbf{x} \rangle)$ is limited, leading to more consistent attributions across small perturbations and therefore tighter stability bounds. Specifically, for sigmoid, $H(z) = \frac{1}{1+e^{-z}}$ and $H'(z) = H(z)(1-H(z))$, both of which remain bounded as $H(z)$ approaches 0 or 1. Conversely, for ReLU activation, $H(z) = \max(0, z)$ with $H'(z) = 1$ when $z > 0$ and 0 otherwise, the gradient can change abruptly across input perturbations. Thus, for perturbations where $\mathbf{x}$ is shifted across the activation boundary, $H'(\langle \mathbf{w}, \mathbf{x} \rangle)$ may vary significantly, producing looser bounds. Similarly, the type and scale of input perturbations (Gaussian noise with $\mathbf{n} \sim \mathcal{N}(0, \sigma^2)$) can also impact bound tightness. For small perturbations with minimal output change, the stability bounds remain tight. However, larger perturbations can result in more significant output shifts $|F(\mathbf{x}') - F(\mathbf{x})|$, leading to looser bounds. This can be pronounced for high dimensional images which tend to lie close to decision boundaries, making them susceptible to small noise that can lead to misclassification (Tanay & Griffin, 2016).

H Relationship between attribution stability and model sensitivity

Consider a single-layer DNN with the form $F(\mathbf{x}) = H(\langle \mathbf{w}, \mathbf{x} \rangle)$, where H is a differentiable scalar-valued activation function (e.g., sigmoid), $\langle \mathbf{w}, \mathbf{x} \rangle$ is the dot product between the weight vector $\mathbf{w}$ and input $\mathbf{x} \in \mathbb{R}^d$.

H.1 Relationship for Vanilla Gradient (VG)(Simonyan et al., 2014)

Let $\mathbf{x} \in R^d$ denote an input image. The Vanilla Gradient (VG) explanation for a model F is computed as,

$$VG(\mathbf{x}) = \frac{\partial F_c(\mathbf{x})}{\partial \mathbf{x}} \tag{17}$$

For a single-layer DNN with the form $F(\mathbf{x}) = H(\langle \mathbf{w}, \mathbf{x} \rangle)$, where H is a differentiable scalar-valued activation function, $\langle \mathbf{w}, \mathbf{x} \rangle$ is the dot product between the weight vector $\mathbf{w}$ and input $\mathbf{x} \in \mathbb{R}^d$, the VG can be computed by applying the chain rule as follows:

$$VG(\mathbf{x}) = \frac{\partial H(\langle \mathbf{w}, \mathbf{x} \rangle)}{\partial \langle \mathbf{w}, \mathbf{x} \rangle} . \frac{\partial \langle \mathbf{w}, \mathbf{x} \rangle}{\partial \mathbf{x}} = H'(\langle \mathbf{w}, \mathbf{x} \rangle)\mathbf{w} \tag{18}$$

Here, $H'(\langle \mathbf{w}, \mathbf{x} \rangle)$ is the gradient of activation function H with respect to the $\langle \mathbf{w}, \mathbf{x} \rangle$. Let $z = \langle \mathbf{w}, \mathbf{x} \rangle$ and $H(z) = \frac{1}{1+exp(-z)}$ be a sigmoid activation function then,

$$\begin{aligned} H'(z) &= \frac{exp(-z)}{(1+exp(-z))^2} \\ &= \frac{1}{1+exp(-z)}(1 - \frac{1}{1+exp(-z)}) \\ &= H(z)(1 - H(z)) \end{aligned} \tag{19}$$

Then, the VG attribution for an input $\mathbf{x}$ is given by

$$VG^F(\mathbf{x}) = H(\langle \mathbf{w}, \mathbf{x} \rangle)(1 - H(\langle \mathbf{w}, \mathbf{x} \rangle))\mathbf{w} \tag{20}$$

Now consider $\mathbf{x}' \in \mathcal{N}_\mathbf{x}$ is a noisy version of input image $\mathbf{x}$ where $\mathcal{N}_\mathbf{x}$ indicates a neighborhood of inputs $\mathbf{x}$ where the model prediction is locally consistent. Then, the VG attribution for an input $\mathbf{x}'$ is given by

$$VG^F(\mathbf{x}') = H(\langle \mathbf{w}, \mathbf{x}' \rangle)(1 - H(\langle \mathbf{w}, \mathbf{x}' \rangle))\mathbf{w} \tag{21}$$

The stability of the VG attribution is computed as the norm of the difference between the attribution of the original image and its noisy counterpart and can be expressed as

$$\Delta = ||VG^F(\mathbf{x}') - VG^F(\mathbf{x})||_1 \tag{22}$$

Substituting the expressions for $VG^F(\mathbf{x})$ and $VG^F(\mathbf{x}')$, and simplifying, we obtain

$$\begin{aligned} \Delta &= ||VG^F(\mathbf{x}') - VG^F(\mathbf{x})||_1 \\ &= ||H(\langle \mathbf{w}, \mathbf{x}' \rangle)(1 - H(\langle \mathbf{w}, \mathbf{x}' \rangle))\mathbf{w} - H(< \mathbf{w}, \mathbf{x} >)(1 - H(\langle \mathbf{w}, \mathbf{x} \rangle)).\mathbf{w}||_1 \\ &= ||\Big(H(\langle \mathbf{w}, \mathbf{x}' \rangle)(1 - H(\langle \mathbf{w}, \mathbf{x}' \rangle)) - H(\langle \mathbf{w}, \mathbf{x} \rangle)(1 - H(\langle \mathbf{w}, \mathbf{x} \rangle))\Big)\mathbf{w}||_1 \\ &= ||\Big(F(\mathbf{x}')(1 - F(\mathbf{x}')) - F(\mathbf{x})(1 - F(\mathbf{x}))\Big)\mathbf{w}||_1 \\ &= ||\Big((F(\mathbf{x}') - F(\mathbf{x}))(1 - F(\mathbf{x}') - F(\mathbf{x}))\Big)\mathbf{w}||_1 \end{aligned} \tag{23}$$

Bounding this by the magnitude of the change in model prediction,

$$\begin{aligned}\Delta &\leq ||\Big(F(\mathbf{x}') - F(\mathbf{x})\Big)\mathbf{w}||_1 \\ \Delta &\leq ||F(\mathbf{x}') - F(\mathbf{x})||_1.||\mathbf{w}||_1\end{aligned} \tag{24}$$

Assuming $\mathbf{w}$ to be constant for a given model, the stability of the VG attribution is a direct result of the sensitivity of the model $||F(\mathbf{x}') - F(\mathbf{x})||$.

H.2 Relationship for Integrated Gradients (IG) (Sundararajan et al., 2017)

The feature attribution score computed by Integrated Gradients (IG) for feature i of input image $\mathbf{x} \in R^d$ with baseline $\mathbf{u}$, model F is given by:

$$IG_i^F(\mathbf{x}, \mathbf{u}) = (x_i - u_i).\int_{\alpha=0}^{1} \partial_i F(\mathbf{u} + \alpha(\mathbf{x} - \mathbf{u}))\partial\alpha \tag{25}$$

For an input image $\mathbf{x}$, IG returns a vector $IG^F(\mathbf{x}, \mathbf{u}) \in R^d$ with scores that quantify the contribution of x_i to the model prediction $F(\mathbf{x})$. For a single layer network $F(\mathbf{x}) = H(\langle \mathbf{w}, \mathbf{x} \rangle)$ where H is a differentiable scalar-valued function and $\langle \mathbf{w}, \mathbf{x} \rangle$ is the dot product between the weight vector $\mathbf{w}$ and input $\mathbf{x} \in R^d$, IG attribution has a closed form expression (Chalasani et al., 2020).

For given $\mathbf{x}$, $\mathbf{u}$ and α, let us consider $\mathbf{v} = \mathbf{u} + \alpha(\mathbf{x} - \mathbf{u})$. If the single-layer network is represented as $F(\mathbf{x}) = H(\langle \mathbf{w}, \mathbf{x} \rangle)$ where H is a differentiable scalar-valued function, $\partial_i F(\mathbf{v})$ can be computed as:

$$\begin{aligned}\partial_i F(\mathbf{v}) &= \frac{\partial F(\mathbf{v})}{v_i} \\ &= \frac{\partial H(\langle \mathbf{w}, \mathbf{v} \rangle)}{\partial v_i} \\ &= H'(z)\frac{\partial \langle \mathbf{w}, \mathbf{v} \rangle}{\partial v_i} \\ &= w_i H'(z)\end{aligned} \tag{26}$$

Here, $H'(z)$ is the gradient of the activation $H(z)$ where $z = \langle \mathbf{w}, \mathbf{v} \rangle$. To compute $\frac{\partial F(\mathbf{v})}{\partial \alpha}$:

$$\frac{\partial F(\mathbf{v})}{\partial \alpha} = \sum_{i=1}^{d}(\frac{\partial F(\mathbf{v})}{\partial v_i}\frac{\partial v_i}{\partial \alpha}) \tag{27}$$

We can substitute value of $\frac{\partial v_i}{\partial \alpha} = (x_i - u_i)$ and $\partial_i F(\mathbf{v})$ from Eq. 26 to Eq. 27.

$$\begin{aligned}\frac{\partial F(\mathbf{v})}{\partial \alpha} &= \sum_{i=1}^{d}[w_i H'(z)(x_i - u_i)] \\ &= \langle \mathbf{x} - \mathbf{u}, \mathbf{w} \rangle H'(z)\end{aligned} \tag{28}$$

This gives:

$$dF(\mathbf{v}) = \langle \mathbf{x} - \mathbf{u}, \mathbf{w} \rangle H'(z)\partial\alpha \tag{29}$$

Since $\langle \mathbf{x} - \mathbf{u}, \mathbf{w} \rangle$ is scalar,

$$H'(z)\partial\alpha = \frac{dF(\mathbf{v})}{\langle \mathbf{x} - \mathbf{u}, \mathbf{w} \rangle} \tag{30}$$

Eq. 30 can be used to rewrite the integral in the definition of $IG_i^F(\mathbf{x})$ in Eq. 25,

$$\begin{aligned}\int_{\alpha=0}^{1} \partial_i F(\mathbf{v}) \partial\alpha &= \int_{\alpha=0}^{1} w_i H'(z) \partial z \quad \text{[From Eqn. 26]} \\ &= \int_{\alpha=0}^{1} w_i \frac{dF(\mathbf{v})}{\langle \mathbf{x}-\mathbf{u}, \mathbf{w} \rangle} \\ &= \frac{w_i}{\langle \mathbf{x}-\mathbf{u}, \mathbf{w} \rangle} \int_{\alpha=0}^{1} dF(\mathbf{v}) \\ &= \frac{w_i}{\langle \mathbf{x}-\mathbf{u}, \mathbf{w} \rangle} [F(\mathbf{x}) - F(\mathbf{u})] \end{aligned} \tag{31}$$

Hence, we obtain the closed form for Integrated Gradients from its definition in Eqn. 25 as

$$\begin{aligned} IG_i^F(\mathbf{x}, \mathbf{u}) &= [F(\mathbf{x}) - F(\mathbf{u})] \frac{(x_i - u_i) w_i}{\langle \mathbf{x}-\mathbf{u}, \mathbf{w} \rangle} \\ IG^F(\mathbf{x}, \mathbf{u}) &= [F(\mathbf{x}) - F(\mathbf{u})] \frac{(\mathbf{x}-\mathbf{u}) \odot \mathbf{w}}{\langle \mathbf{x}-\mathbf{u}, \mathbf{w} \rangle} \end{aligned} \tag{32}$$

Here, $\odot$ is the entry-wise product of two vectors.

Now consider $\mathbf{x}' \in \mathcal{N}_{\mathbf{x}}$ is a noisy version of input image $\mathbf{x}$ where $\mathcal{N}_{\mathbf{x}}$ indicates a neighborhood of inputs $\mathbf{x}$ where the model prediction is locally consistent. The stability of the IG attribution can be computed using Eqn. 33.

$$\Delta = ||IG^F(\mathbf{x}', \mathbf{u}) - IG^F(\mathbf{x}, \mathbf{u})||_1 \tag{33}$$

This is equivalent to,

$$\begin{aligned} \Delta &\approx ||IG^F(\mathbf{x}', \mathbf{x})||_1 \\ &= \Big|\Big| [F(\mathbf{x}') - F(\mathbf{x})] \frac{(\mathbf{x}'-\mathbf{x}) \odot \mathbf{w}}{\langle \mathbf{x}'-\mathbf{x}, \mathbf{w} \rangle} \Big|\Big|_1 \\ &= \Big|\Big| [F(\mathbf{x}') - F(\mathbf{x})] \frac{\Delta_x \odot \mathbf{w}}{\langle \Delta_x, \mathbf{w} \rangle} \Big|\Big|_1 \end{aligned} \tag{34}$$

Assuming $\mathbf{w}$ to be constant for a given model, we can conclude from Eqn. 34 that the sensitivity of the IG attribution is a direct result of the sensitivity of the model $||F(\mathbf{x}') - F(\mathbf{x})||$.

H.3 Relationship for SmoothGrad (SG) (Smilkov et al., 2017)

To compute SmoothGrad (SG), we introduce Gaussian noise $\mathbf{n} \sim \mathcal{N}(0, \sigma^2)$ to the input $\mathbf{x}$ and compute the input-gradient for multiple noisy samples $\mathbf{x}_k = \mathbf{x} + \mathbf{n}_k$ for $k = 1, \ldots, N$, where N is the number of noise samples.

$$SG(\mathbf{x}) = \frac{1}{N} \sum_{k=1}^{N} \frac{\partial F(\mathbf{x}_k)}{\partial \mathbf{x}_k} \tag{35}$$

SG explanation is then obtained by averaging the explanations. Since SG is a simple averaging of Vanilla Gradient, the relationship for SG follows from relationship of VG, as shown in Section H.1.

I Evaluation metrics

Below, we discuss evaluation metrics used in our experiments.

I.1 Sparsity (Chalasani et al., 2020)

We measure the sparsity of the attribution vector $\phi(\mathbf{x})$ by computing its Gini index. Given a vector of attribution $\phi(\mathbf{x}) \in R^d$, the absolute of the vector is first sorted in non-decreasing order, and the Gini index is computed using Eqn. 36.

$$G(\phi(\mathbf{x})) = 1 - 2\sum_{k=1}^{d} \frac{\phi(\mathbf{x})_{(k)}}{||\phi(\mathbf{x})||_1} \frac{d-k+0.5}{d} \tag{36}$$

The formula calculates a weighted sum of fractions, where each fraction represents the contribution of the k-th largest element to the overall sparsity. The formula assigns greater weight to larger elements and smaller weight to smaller elements. The Gini Index values lie in between $[0, 1]$; A value of 1 indicates perfect sparsity, where only one element in the vector $\phi_i(\mathbf{x}) > 0$. The sparsity is zero if all the vectors are equal to some positive value.

I.2 Stability (Agarwal et al., 2022)

The stability metric measures how similar explanations are for similar inputs. Relative input stability (given by Eqn. 37) is measured as the difference between two attribution vectors $\phi(\mathbf{x})$ and $\phi(\mathbf{x}')$ with respect to the difference between the two inputs $\mathbf{x}$ and $\mathbf{x}'$. $\mathbf{x}'$ is computed by perturbing $\mathbf{x}$. A lower RIS value shows that explanations are similar for similar inputs.

$$\begin{aligned} RIS = max_{\mathbf{x}'} \frac{||\frac{\phi(\mathbf{x})-\phi(\mathbf{x}')}{\phi(\mathbf{x})}||}{max(||\frac{\mathbf{x}-\mathbf{x}'}{\mathbf{x}}||_p, \epsilon_{min})} \\ \forall \mathbf{x}' \; s.t. \; \mathbf{x}' \in \mathcal{N}_{\mathbf{x}}; \hat{y}_{\mathbf{x}} = \hat{y}_{\mathbf{x}'} \end{aligned} \tag{37}$$

Relative input stability only measures the difference in input space and does not measure whether there was a change in the logic path of a network for a perturbed input.

Relative output stability (given by Eqn. 38) measures the difference between two attribution vectors $\phi(\mathbf{x})$ and $\phi(\mathbf{x}')$ with respect to the difference between the model logits for two inputs $z(\mathbf{x})$ and $z(\mathbf{x}')$ when $\mathbf{x}$ is perturbed to produce $\mathbf{x}'$. A lower ROS value shows that explanations are similar for similar inputs.

$$\begin{aligned} ROS = max_{\mathbf{x}'} \frac{||\frac{\phi(\mathbf{x})-\phi(\mathbf{x}')}{\phi(\mathbf{x})}||}{max(||z(\mathbf{x}) - z(\mathbf{x}')||_p, \epsilon_{min})} \\ \forall \mathbf{x}' \; s.t. \; \mathbf{x}' \in \mathcal{N}_{\mathbf{x}}; \hat{y}_{\mathbf{x}} = \hat{y}_{\mathbf{x}'} \end{aligned} \tag{38}$$

$\mathcal{N}_{\mathbf{x}}$ in Eqn. 37, and Eqn. 38 indicates a neighborhood of inputs $\mathbf{x}'$ similar to $\mathbf{x}$. We use the implementation of the stability metrics available in Quantus (Hedström et al., 2023).

I.3 ROAD: Remove and Debias (Rong et al., 2022)

Faithfulness metrics that involve pixel removal and measuring model prediction changes (such as insertion/deletion (Petsiuk et al., 2018)) introduces artifacts and cause a distribution shift in the perturbed inputs. Retraining based approaches like ROAR (Hooker et al., 2019) addresses this problem but is computationally expensive. ROAD (Rong et al., 2022) addresses both concerns in faithfulness evaluation.

ROAD measures the accuracy of a model on the provided test set at each step of an iterative process of removing k most important pixels. Removal of pixels is done with a noisy linear imputation to avoid out-of-distribution samples. We set $k = 5$ in our experiments, and adopt the MoRF (Most Relevant First) removal strategy where a faster drop in accuracy with increase in removal of k most important features indicate that key discriminative features are being removed. ROAD demonstrates consistent results with both MoRF and LeRF (Least Removal First) removal strategy. For further details, see Rong et al. (2022).

I.4 Structural similarity (Adebayo et al., 2018)

Structural similarity measures the structural similarity between saliency maps of original and perturbed samples, given the same model prediction. We measure the similarity of saliency maps using the structural similarity index (SSIM). For each image, we add Gaussian noise and generate its noisy version such that the model prediction is consistent. We then compute the saliency map of the two images and measure the structural similarity between the maps.

J Additional visualization

We provide additional visualizations on Vanilla Gradient (VG) in Figures 17, 18 and 19 for various models: naturally-trained (N), adversarially-trained (A), adversarial training with mean-filter smoothing (M1), adversarial training with median-filter smoothing (M2), adversarial training with Gaussian-filter smoothing (G), adversarial training with embedded filter smoothing (E), and adversarial training with non-local gaussian smoothing (NG). We can observe that saliency maps from the adversarial models (A) are sparser than the naturally trained model (N). Adversarially trained models with local feature map smoothed models (M1, M2, G) reduce the sparsity to improve stability. The use of non-local smoothing filters (E and NG) increases the sparsity further.

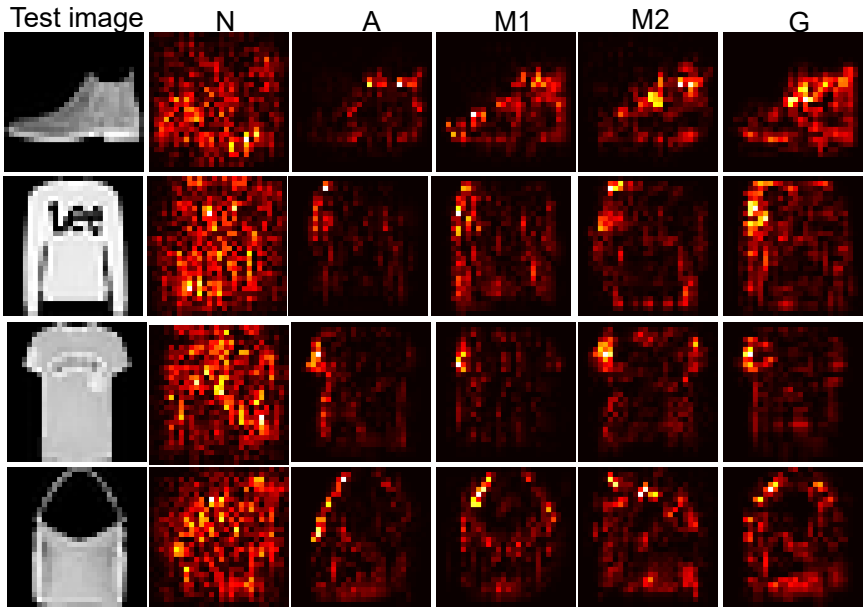


Figure 17: Additional visualization for VG (FMNIST) (N: naturally-trained, A: adversarially-trained, M1: adversarially-trained with mean-filter, M2: adversarially-trained with median-filter, G: adversarially-trained with Gaussian-filter)

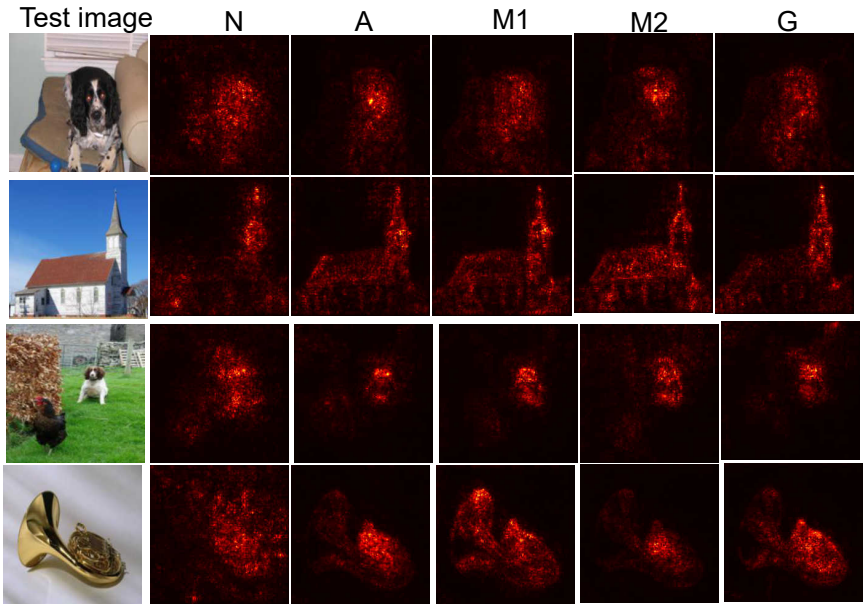


Figure 18: Additional visualization for VG (ImageNette) (N: naturally-trained, A: adversarially-trained, M1: adversarially-trained with mean-filter, M2: adversarially-trained with median-filter, G: adversarially-trained with Gaussian-filter, E: adversarially-trained with embedded filter, NG: adversarially-trained with non-local gaussian)

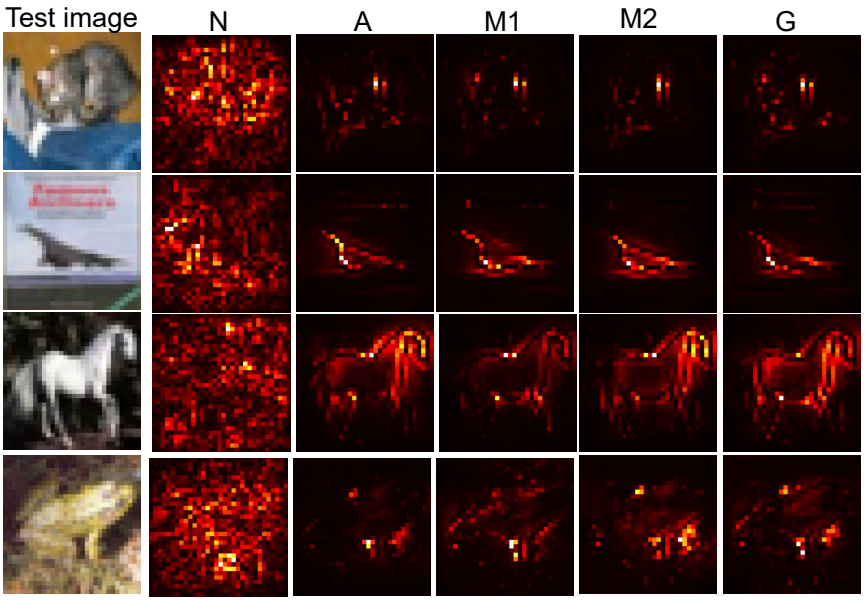


Figure 19: Additional visualization for VG (CIFAR-10) (N: naturally-trained, A: adversarially-trained, M1: adversarially-trained with mean-filter, M2: adversarially-trained with median-filter, G: adversarially-trained with Gaussian-filter, E: adversarially-trained with embedded filter, NG: adversarially-trained with non-local gaussian)