Defensive Strategy for Explainability in Deep Neural Networks Under Adversarial Attacks

Tuan Trung Mac[1,2][0009−0003−0007−3998], Tan Loc Nguyen[1,2][0009−0001−7488−1643], and Bac Le[1,2(✉)][0000−0002−4306−6945]

[1] Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam
Email: mttrung21@clc.fitus.edu.vn, ntloc21@clc.fitus.edu.vn, lhbac@fit.hcmus.edu.vn

Abstract. Deep Neural Networks (DNNs) perform excellently on most problems, although the issue with many complex models is their vulnerability to adversarial attacks, which can mislead the model's explanation. Recent work indicates that explanation mechanisms can be compromised by an attacker who alters the explanation while the output remains accurate. This lowers the reliability and robustness of the explanation. Our work considers the threat of such an attack to Explainable Artificial Intelligence (xAI) and introduces a mechanism for defending the explanation against such attacks. Our research proposes NODA (Normalization Defense Against Adversaries), based on a Hessian regularizer and data normalization, to enhance the reliability of the explanation. Our investigation validates that NODA is effective in defending against the attack while not damaging the model's performance.

Keywords: Artificial Intelligence, Explainable Artificial Intelligence, Adversarial Attacks, Defense Against Adversarial Attacks.

1 Introduction

Deep learning models have performed very well in multiple domains but have raised questions about trust, fairness, and accountability in some cases [16]. A challenge is the lack of transparency in understanding how these models make decisions. One significant method for understanding AI decision-making is feature attribution based on gradients, which explains decisions in terms of model gradients. Gradient-based techniques are sensitive to noise, which prevents them from performing efficiently [8] [7]. Research has shown a strong connection between the interpretation of the model and its defense against attacks, suggesting that improving attribution methods can enhance the robustness of models against manipulations designed to deceive the model.

A range of explainable methods based on gradients have been utilized to overcome these limitations, such as Integrated Gradients [13], SmoothGrad [17], Grad-CAM [10], and Simple Gradients. The goal of these methods is to identify the important features that affect model outputs, thus making them more

interpretable and accountable. However, recent work has shown that these explanation methods are vulnerable to adversarial attacks [1].

This suggests that understanding the hidden mechanisms behind such attacks is crucial to realize that several mechanisms for explanations draw their reasons behind their explanations through the gradient of the black-box function f against the input x. As such, several scenarios may develop where someone might exploit this by performing small alterations to x (in its local neighborhood), having strong effects on the explanations given. Thus, defensive mechanisms must ensure stable explanation values in the local neighborhood of x to prevent such exploits.

Several ways have been advocated to make explanations robust: to keep consistency in explanations in the neighborhood around x, to simplify second derivatives to make them immune to infinitesimal changes, to regularize first derivatives to avoid abrupt changes, and to apply smoothening in gradients to avoid stability issues. Within this work in development, an efficient novel method is presented to make explanations robust.

The remainder of the paper is organized as follows. Section 2 reviews Adversarial Attacks on xAI, defense strategies, and data normalization. Section 3 details our methodology, including attack models targeting explanations and our defense techniques. Section 4 describes the experimental setup to evaluate defense effectiveness. Section 5 presents the evaluation results comparing various defense methods. Finally, Section 6 discusses future work.

2 Related Works

2.1 Adversarial Attacks on Explainable Artificial Intelligence

Today, neural network models are being used more and more in different fields. However, we do not fully understand how these models learn and make decisions based on the input data. We also can not know when AI models might be biased against certain groups or make unexpected mistakes. That's why we need a way to understand their decisions and ensure we can trust their outputs. There are various methods, known as interpreters, that aim to explain a model's output. Some focus on local explanations, while others provide global insights. Some are designed for transparent (white-box) models, while others work with complex (black-box) models [1].

In this part, we concentrate on interpreters that are based on gradients for black-box deep learning models. Interpretation methods by gradients, including SmoothGrad [17], Integrated Gradients [13], and Simple Gradients [17], have been used extensively to clarify deep learning models by relating predictions to input features.

We define the basic notations used throughout this work:

- x : Original input data
- f : A model to feed x
- y : The output or prediction given by the model

– $\mathcal{I}(f, x)$: Interpreter method to explain the output of $f(x)$

Simple Gradients - also called saliency map by Z. Wang et al. (2020) [17], is an easy way to explain based on output y via input x:

$$g(x) = \nabla_x f(x)$$

Integrated Gradients - proposed by Mukund Sundararajan et al. (2017) [13] identified that previous explanation methods like LRP and DeepLift could potentially be sensitive to unimportant factors. To overcome this weakness, Integrated Gradients was introduced, using an internal path from a predefined baseline to provide more reliable attributions:

$$g(x) = (x - x') \int_0^1 \frac{\partial f(x' + \alpha(x - x'))}{\partial x} d\alpha$$

Smooth Gradients - Proposed by Daniel Smilkov et al. (2017) [11], this method overcomes the limitations of simple gradients by averaging gradient values over a neighborhood around the input x, using a normal distribution. The research of Z. Wang et al. (2020) [17] proved that a higher variance σ enhances the robustness of Smooth Gradients, guided by a global Lipschitz function.

$$g(x) = \mathbb{E}_{z \sim \mathcal{N}(x, \sigma^2)} \nabla_z f(z)$$

Szegedy et al. (2014) [14] soon introduced a method to make the model's output incorrect by making a very small change to the input data - a change that is hard for humans to see. This also motivated A. Ghorbani et al. (2019) [4] to research ways to perturb the input slightly, leading to significant changes in the model's explanation. In this section, we focus on attackers who use gradients to determine the perturbation method for attacking the explainability of a black-box model by poisoning the input data. According to A. Ghorbani et al. (2019) [4], data poisoning can be categorized into several main types:

Targeted Attacks. These attacks modify the explanations toward a predefined target explanation t, by minimizing the distance:

$$D(t, I(f, x + \delta))$$

While ensuring that the model's prediction remains unchanged:

$$f(x) = f(x + \delta)$$

where $I(f, x)$ represents the explanation of model f on input x.

Top-k Attacks. These attacks alter the explanations to significantly change the top-k contributors, reducing their importance in the original saliency map.

Mass Center Attacks. These attacks manipulate the explanations to shift the "center of mass" of the saliency map. The mass center of a saliency map is defined as:

$$C_x = \frac{\sum_{x,y} x \cdot I(x,y)}{\sum_{x,y} I(x,y)}, \quad C_y = \frac{\sum_{x,y} y \cdot I(x,y)}{\sum_{x,y} I(x,y)}$$

where (C_x, C_y) are the coordinates of the mass center. This results in a saliency map with significantly different weightings.

2.2 Defense Strategies Against Adversarial Attacks

Our methodology is inspired by the work of A. K. Dombrowski et al. (2022) [3], which introduced important techniques to ensure that the model's gradients remain consistent within a neighborhood of an input x:

Frobenius Norm of the Hessian. The Hessian matrix is a square matrix of second-order partial derivatives of the model output with respect to the input. It is defined as:

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}, \quad \|H\|_F^2 = \sum_i \sum_j \left(\frac{\partial^2 f}{\partial x_i \partial x_j} \right)^2$$

Attackers tend to make small changes to the input, which can significantly disrupt explanations based on gradients. The key idea is that if a small perturbation in x causes a large change in the gradient, then the second derivative (i.e., the Hessian norm) is large. To improve robustness, we aim to keep this norm small.

Since computing the exact Hessian norm is expensive, we approximate the ℓ_2 Frobenius norm using the Taylor series [9]

To further enhance this approach, Sunghwan Joo et al. (2023) [5] introduced a method that not only accelerates the reduction of the Hessian norm backpropagation but also ensures that gradients in the neighboring region remain aligned by maintaining a high cosine similarity. The methods add two more regularization terms to the loss function, which are ℓ_2 loss in Eq. (ℓ_2) and Cosine loss in Eq. (cosine):

$$\Gamma^{\ell_2}_{\nabla g}(x, \boldsymbol{\delta_x}) = \alpha \|\nabla g_y(x + \boldsymbol{\delta_x}) - \nabla g_y(x)\|_2 \qquad (\ell_2)$$

$$\Gamma^{\cos}_{\nabla g_y}(x, \boldsymbol{\delta_x}) \coloneqq \frac{1}{2} \left(1 - \operatorname{cossim}(\nabla g_y(x + \boldsymbol{\delta_x}), \nabla g_y(x))\right), \qquad (\text{cosine})$$

Softplus [3]. The Softplus function, defined as $\text{Softplus}(x) = \frac{1}{\beta}\ln(1+e^{\beta x})$, has been shown to have bounded first and second derivatives:

$$|\ \text{Softplus}'(x)\ | \leq 1, \quad |\ \text{Softplus}''(x)\ | \leq \frac{1}{4}\beta.$$

Weight Decay [3]. It has been proven that the Frobenius norm of the Hessian is proportional to the norm of the model's weights and the second derivative of the activation function. Since the second derivative of the Softplus function is bounded, we only need a regularization term to ensure that the ℓ_2 norm of the model parameters remains small.

2.3 Data Normalization on Complex Machine Learning Model Performance

Data normalization of data is a very important process that enhances both effectiveness and performance of both simple and complex machine learning models. It is done by bringing down the value in the data to maintain consistency while emphasizing key aspects to analyze and eliminate data noises and outliers that can infect results or lead to inaccurate predictions.

One of the techniques that is used most is Z-score normalization, which standardizes data by centering it around a zero mean with unit variance: $x'_{i,n} = \frac{x_{i,n}-\mu_i}{\sigma_i}$, where μ_i and σ_i denote respectively the mean and standard deviation of the i-th feature and this method is effective for Gaussian-distributed data but can be sensitive to outliers. An alternative solution is Mean Centering normalization also treats the problem of offset by removing the mean of each of the features: $x'_{i,n} = x_{i,n} - \mu_i$.

3 Methodology

3.1 Adversarial Attack Model

We model adversarial perturbations targeting explanations by modifying input data x to $x + \delta$ such that the explanation $I(x)$ changes significantly, yet the prediction of the model remains the same. We examine both white-box and black-box attacks against xAI methods.

White-box attacks assume complete access to the attacker's environment, knowledge of the model, and gradients. This knowledge aids the attacker in making modifications specifically targeting saliency explanations. Black-box attacks oppose this and never get direct gradients' knowledge. They employ methods that query with modifications in explanations and constant predictions.

3.2 Defense Strategies

To mitigate adversarial threats to xAI, we explore multiple strategies aimed at improving robustness:

NODA: Normalization Defense Against Adversaries

This segment will further study the effect of normalization on the stability of explanations of models. Most models use normalization methods on the preprocessed data before the training process. The normalization methods optimize the training effectiveness of the models. One of the widely used methods includes Z-score standardization:

Considering an individual data point x from the sample X, we define the mean μ and standard deviation σ as:

$$\mu = \frac{1}{|X|}\sum_{i=1}^{|X|} x_i, \quad \sigma = \sqrt{\frac{1}{|X|}\sum_{i=1}^{|X|}(x_i - \mu)^2}$$

Then, the Z-score standardization is given by:

$$z = \frac{x - \mu}{\sigma}$$

This procedure typically comes before the training of the model and forms a part of the preprocessed data. Let the original training set be X, with $X \sim \mathcal{N}(\mu, \sigma^2)$. Let the normalized set Z be the set of the standard variables of X, with $Z = \{\text{normalize}(x) \mid x \in X\}$, where normalize$(x)$ denotes the Z-score standardization of each item x of the set X, based on normalize$(x) = \frac{x-\mu}{\sigma}$, where μ and σ are the mean and standard deviation of the variable X, respectively.

The methodology proposed by A. K. Dombrowski et al. (2022) [3] comes under analysis with a specific focus placed on the application of the Hessian matrix-associated squared Frobenius norm.

Mathematically, the Hessian matrix, H, is defined as $H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$. Incorporating such normalization of the input data before training results in the following transformation on the Hessian - $H'_{ij} = \frac{\partial^2 f}{\partial z_i \partial z_j}$, where the standard input z has been defined such that $z = \frac{x-\mu}{\sigma}$.

The expression that results from the derivative of z with respect to x

$$\frac{dz}{dx} = \frac{1}{\sigma}$$

Given that this forms a linear transformation, the second derivative becomes nonexistent:

$$\frac{d^2 z}{dx^2} = 0$$

Applying the chain rule, we express the Hessian transformation:

$$
\begin{aligned}
\frac{\partial^2 f}{\partial x_i \partial x_j} &= \frac{\partial}{\partial x_j}\left(\frac{\partial f}{\partial z_i}\frac{\partial z_i}{\partial x_i}\right) \\
&= \frac{\partial}{\partial x_j}\left(\frac{\partial f}{\partial z_i}\right) \times \frac{\partial z_i}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\frac{\partial z_i}{\partial x_i}\right) \times \frac{\partial f}{\partial z_i} \\
&= \frac{\partial}{\partial z_j}\left(\frac{\partial f}{\partial z_i}\right) \times \frac{\partial z_j}{\partial x_j} \times \frac{\partial z_i}{\partial x_i} + 0 \\
&= \frac{1}{\sigma^2} \times \frac{\partial^2 f}{\partial z_i \partial z_j}
\end{aligned}
$$

According to a paper of A. K. Dombrowski et al. (2022) [3], the Hessian square of the expression of the Frobenius norm is

$$
\|H\|_F^2 = \sum_i \sum_j \left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right)^2
$$

Integrating our transformation, we have:

$$
\|H\|_F^2 = \sum_i \sum_j \left(\frac{1}{\sigma^2}\frac{\partial^2 f}{\partial z_i \partial z_j}\right)^2 = \frac{1}{\sigma^4}\sum_i \sum_j \left(\frac{\partial^2 f}{\partial z_i \partial z_j}\right)^2 = \frac{1}{\sigma^4}\|H'\|_F^2
$$

Practical Implication

For datasets such as CIFAR-10 [6] and ImageNet [2], the variant in each dataset is that $\sigma \approx 0.25$. Therefore, the Frobenius norm $\|H\|_F^2$ will be approximated $\frac{1}{(0.25)^4} = 256$ times bigger compared with the original input.

Unfortunately, attackers typically manipulate the original input data rather than the normalized one. As a result, the gradient explosion after the normalization process creates a vulnerability, providing attackers with additional opportunities to manipulate the model's explanations [15].

We propose a simple method to integrate normalization directly into the model as a layer before the feedforward layers. This approach enables full backpropagation through the normalization stage and ensures that gradient descent follows a complete Hessian-decreasing direction.

Figure 1 illustrates the training process for NODA (Normalization Defense Against Adversaries). Training begins with raw input data X, which is immediately processed by the model. Unlike previous approaches that apply normalization as a preprocessing step, NODA incorporates normalization within the model itself. This design ensures that backpropagation is performed fully with respect to the original inputs rather than the normalized ones, allowing the model to be trained effectively to minimize the Hessian norm.

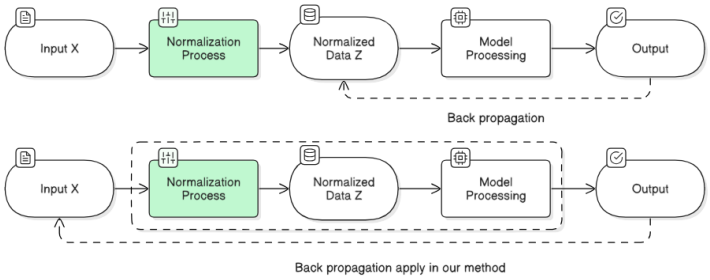

Fig. 1: Model Training Pipeline with NODA: Raw input x undergoes normalization with NODA to enhance robustness before being used for model training.

3.3 Regularization Techniques

The methods proposed by A. K. Dombrowski et al. (2022) [3] and Sunghwan Joo et al. (2023) [5] demonstrate strong performance in defending models against adversarial attacks on explanations. However, a key limitation remains: these studies do not provide a robust mechanism for protecting gradients through the normalization layer. As a result, small input perturbations can still be amplified after normalization, compromising the effectiveness of these defenses. By combining their approaches with NODA - which defends the normalized input and enables full backpropagation through the normalization layer - we can enhance robustness against such attacks.

We explore several techniques to enhance robustness. Our proposed approach can be effectively combined with existing defense strategies to improve the model's robustness:

- **Softplus**: Used to bound the Hessian norm of the activation function.
- **Weight decay**: Reduces the weight of model parameters, which also helps reduce Frobenius norm of the Hessian, as demonstrated in A. K. Dombrowski et al. (2022) [3].
- **Cosine and ℓ_2 loss**: Adds an additional regularization term to keep the gradients small in the local neighborhood and maintain similarity. We preserve the functionality of the loss in Eq. (L_2^*) while making the gradient smoother for training by using squared ℓ_2.

$$\Gamma^{\ell_2}_{\nabla g}(x, \delta_x) = \alpha \|\nabla g_y(x + \delta_x) - \nabla g_y(x)\|_2^2 \qquad (L_2^*)$$

Assuming the model uses cross-entropy loss $\mathcal{L}_{CE}$ as the primary loss function, the final loss is defined as:

$$\mathcal{L}(x, y) = \mathcal{L}_{CE}(x, y) + \lambda_{\ell_2} \mathbb{E}_{\delta_x} \left[\Gamma^{\ell_2}_{\nabla f}(x, \delta_x) \right] + \lambda_{\cos} \mathbb{E}_{\delta_x} \left[\Gamma^{\cos}_{\nabla f}(x, \delta_x) \right]$$

4 Experiments

We evaluate our method on CIFAR-10 [6] and ImageNet [2]. ImageNet10 and ImageNet100 are subsets of the original ImageNet1000 and are used to compare our approach with the techniques proposed in the backbone paper. We use the ResNet18 model in two versions: the standard ResNet18 and our modified version, which includes a normalization layer before the usual feedforward process.

Table 1: Training Parameters

Parameter	**Value**
Softplus β	3.0
Weight decay	5×10^{-4}
λ_{ℓ_2}	0.15
λ_{cos}	1.0
Learning rate	0.001
Optimizer	SGD with momentum

We compare the effectiveness of defense against perturbations on explanations using the following models:

- **CE only**: Trained using only the original cross-entropy loss.
- **Hessian**: Trained with a Hessian norm regularization term (proposed by A. K. Dombrowski et al. (2022) [3]).
- $\boldsymbol{\ell_2}$ **+ cos**: Trained with ℓ_2 and cosine-based robustness criteria (proposed by Joo et al. (2023) [5]).
- **Our**: Trained with NODA combined with $\boldsymbol{\ell_2}$ **+ cos**.

We evaluate the efficiency of our technique using the following metrics:

- **Accuracy.** Ensures that the technique enhances model explanation robustness without decreasing the model's accuracy.
- **The Random Perturbation Similarity (RPS).** RPS is defined in A. K. Dombrowski et al. (2022) [3] and Joo et al. (2023) [5]. RPS measures the similarity of the explanation at the given point x and the randomly perturbed point $x + \boldsymbol{\delta}_x$. To simplify, the explainer used is Simple Gradients, and we measure similarity using the cosine function:

$$RPS_g(\epsilon, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} \mathbb{E}_{\delta_x \sim \mathcal{U}_d([-\epsilon,\epsilon]^d)} \cos\big(h_{f_y}(x+\delta_x), h_{f_y}(x)\big).$$

With the test dataset $\mathcal{D}$, and $\mathcal{U}_d$ denoting the uniform distribution, we also sample 10 times, as in Joo et al. (2023) [5], for each data tuple to approximate the expectation over δ_x. The noise level ϵ is set to 16.

– **Insertion Score (INS).** In the original Insertion Score proposed in Joo et al. (2023) [5], we begin with a black image from the test set and reconstruct it using a ratio γ. From the explanation saliency map, we select the top γ contributors to the output and evaluate the model's confidence based on the constructed input. We denote the reconstructed input as $x^{\mathcal{I}_\gamma}$:

$$x^{\mathcal{I}_\gamma} = x \odot m^{\mathcal{I}_\gamma} + x_0 \odot (1 - m^{\mathcal{I}_\gamma}),$$

where x_0 is a zero-valued image, $m^{\mathcal{I}_\gamma} \in \{0,1\}^d$, and $\frac{\sum_i m_i^{\mathcal{I}_\gamma}}{d} = \gamma$. Here, $m_i^{\mathcal{I}_\gamma} = 1$ if the pixel at position i is among the top γ most contributing pixels to the output as determined by the interpreter $\mathcal{I}$.
A high result indicates that the explainer correctly identified the most relevant aspects of the input, leading to a strong explanation for the model decision. However, deleting unimportant parts by masking them of the input with black pixels cannot reliably measure the quality of an explanation, as black pixels (e.g., zeros) are not always a neutral or non-informative baseline. Stassin et al. (2024) [12] also suggested using randomly sampled pixels from a Gaussian distribution instead, as such pixels are less likely to contribute significantly to the model's prediction compared to the black-pixel method.

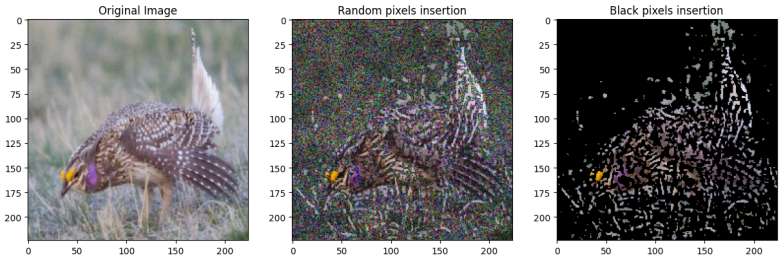

Fig. 2: Reconstructed images used in the INS with $\gamma = 0.15$. Masking with random pixels produces more precise results than masking with black pixels.

– **Hessian Norm:** $\|H\|_F^2$ is used to see whether the method efficiently keeps gradients small in the local neighborhood, as proposed by A. K. Dombrowski et al. (2022) [3]. We evaluate this in two variations: $\|H_x\|_F^2$, which measures the Hessian norm with respect to the original input x; and $\|H_z\|_F^2$, where $z = \text{normalize}(x)$, to analyze the effect of the normalization layer.

5 Results and Discussion

We compare the robustness of the explanation between different defense strategies. Our findings indicate that adversarial training combined with reg-

ularization significantly enhances xAI robustness, reducing attribution manipulation effects. We present qualitative and quantitative results demonstrating the effectiveness of our methods.

5.1 Results on evaluation metrics

We introduce some metrics to evaluate our method and compare it with other proposals. First, we consider accuracy: we require that it does not decrease significantly while enhancing the robustness of the explanation. To evaluate robustness, we use the RPS score and INS score over the normalized input (RPS_z and INS_z) and over the original input (RPS_x and INS_x).

With RPS, the reports use $\epsilon = 16$. For the insertion score, the values of γ used in $\{0.1, 0.2, \ldots, 0.9\}$ and the average values are calculated in all cases.

The results from Table 2, Table 3 and Table 4 demonstrate the performance comparison of different models on the CIFAR-10, ImageNet-10, and ImageNet-100 datasets. Methods such as the Hessian-based approach and ℓ_2+cos, proposed by Dombrowski et al. (2022) [3] and Joo et al. (2023) [5], perform well under the RPS_z metric, which measures the similarity between explanations on normalized inputs and their locally randomized neighbors. However, when evaluated using the RPS_x metric, these methods reveal a weakness in the robustness of the explanation: small changes in the original input can result in significant perturbations after the normalization step.

Our model outperforms the models proposed by A. K. Dombrowski et al. (2022) [3] and Sunghwan Joo et al. (2023) [5], achieving the highest RPS_o and INS_o. This indicates that our method effectively defends against perturbations on the original image over the normalization layer, which causes an explosion in the Hessian norm values.

In contrast, Tables 5 and 6 focus on the Frobenius norm of the Hessian matrix. Let $z = \text{normalize}(x) = \frac{x-\mu}{\sigma}$, where $\|H_z\|_F^2$ represents the Hessian matrix norm of the output with respect to the normalized input, while $\|H_x\|_F^2$ represents the Hessian matrix norm of the output with respect to the original input.

There is a significant difference between training with NODA and the ℓ_2 + cos method [5]. Although ℓ_2 + cos [5] is highly effective in keeping the Hessian norm small, it still leads to a significant explosion of gradients after the normalization layer. In contrast, our NODA method maintains a small Hessian norm, ensuring greater robustness in the local neighborhood around x.

Table 2: CIFAR-10

	ACC	RPS_x	RPS_z	INS_x	INS_z
CE only	**95.1**	0.37	0.69	0.39	0.65
Hessian [3]	94.2	0.94	0.95	0.53	0.81
ℓ_2 + cos [5]	94.4	0.94	0.98	0.64	0.85
Our	94.4	**0.97**	**0.99**	**0.64**	**0.86**

Table 3: IMAGENET-10

	ACC	RPS_x	RPS_z	INS_x	INS_z
CE only	**97.9**	0.80	0.89	0.43	0.69
Hessian [3]	97.4	0.88	0.97	0.48	0.72
ℓ_2 + cos [5]	97.3	0.92	0.97	0.49	0.71
Our	97.6	**0.95**	**0.99**	**0.56**	**0.85**

Table 4: IMAGENET-100

	ACC	RPS_x	RPS_z	INS_x	INS_z
CE only	**79.0**	0.41	0.67	0.31	0.55
Hessian [3]	78.7	0.79	0.83	0.37	0.61
ℓ_2 + cos [5]	78.0	0.86	0.94	0.39	0.63
Our	78.2	**0.94**	**0.97**	**0.42**	**0.66**

5.2 Results on adversarial attacks

We evaluated our proposal with NODA by allowing the input to be manipulated by the attacker, following the approach of A. Ghorbani, A. Abid, and J. Zou (2019) [4], and attempted to explain the output using both the raw ResNet18 model and the ResNet18 model trained with NODA + ℓ_2 + cosine similarity. The explainers we used were Integrated Gradients (IG), Smooth Gradients (SG), and Simple Gradients (G). Figure 3 shows the results of attacking the raw ResNet18 model trained without any defense regularization. The model is completely manipulated, producing significantly different saliency maps. In contrast, the saliency maps provided by the model trained with NODA still highlight the important contributions to the output in Figure 4.

Table 7, Table 8 and Table 9 show the result of metrics:

- **Cosine Similarity**: Measures the similarity between the explanation for the original input and that of the perturbed input.
- **Top-K Contributions**: We select $k = 1000$ to analyze how the most important k pixels in the explanation change after perturbing the input.
- **Center Dislocation**: Computes how the center of mass of the explanation map shifts after the attack.

After applying NODA and the defensive strategy, the model's explanation becomes more robust and focuses precisely on the object (the dog). Moreover, the explanation does not change significantly and continues to highlight the correct objective of the model's output.

Table 5: Frobenius Norm of Hessian on CIFAR10

	$\|H_x\|_F^2$	$\|H_z\|_F^2$
ℓ_2 + cos	2.5	647.5
our	**0.5**	**107.2**

Table 6: Frobenius Norm of Hessian on IMAGENET10

	$\|H_x\|_F^2$	$\|H_z\|_F^2$
ℓ_2 + cos	1.37	544
our	**0.03**	**11**

Table 7: Integrated Gradients - Defense Against Adversarial Attacks

	Cosine Similarity	Top K Contributions	Center Dislocation
CE only	0.26	0.07	67.53
Our	**0.83**	**0.65**	**22.17**

Table 8: Smooth Gradients - Defense Against Adversarial Attacks

	Cosine Similarity	Top K Contributions	Center Dislocation
CE only	0.82	0.59	13.41
Our	**0.94**	**0.82**	**5.01**

Table 9: Simple Gradients - Defense Against Adversarial Attacks

	Cosine Similarity	Top K Contributions	Center Dislocation
CE only	0.20	0.11	63.71
Our	**0.73**	**0.45**	**29.21**

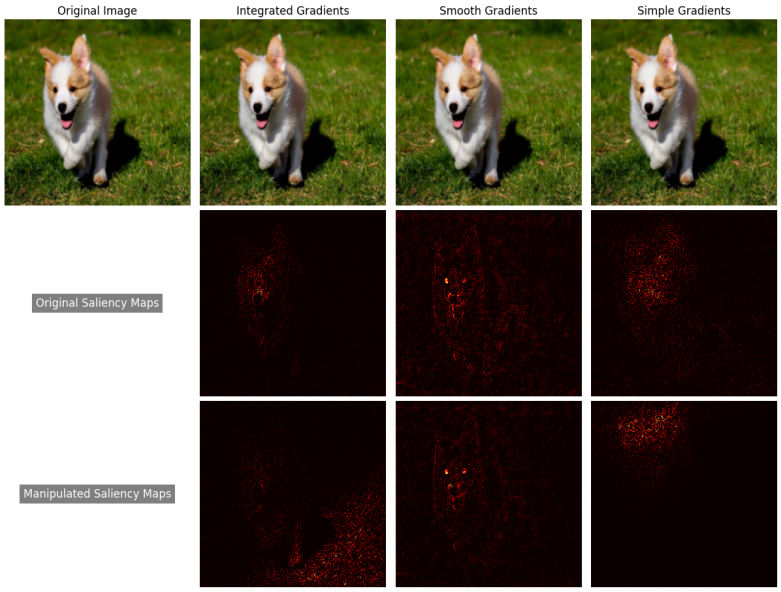

Fig. 3: Attack on CE only model

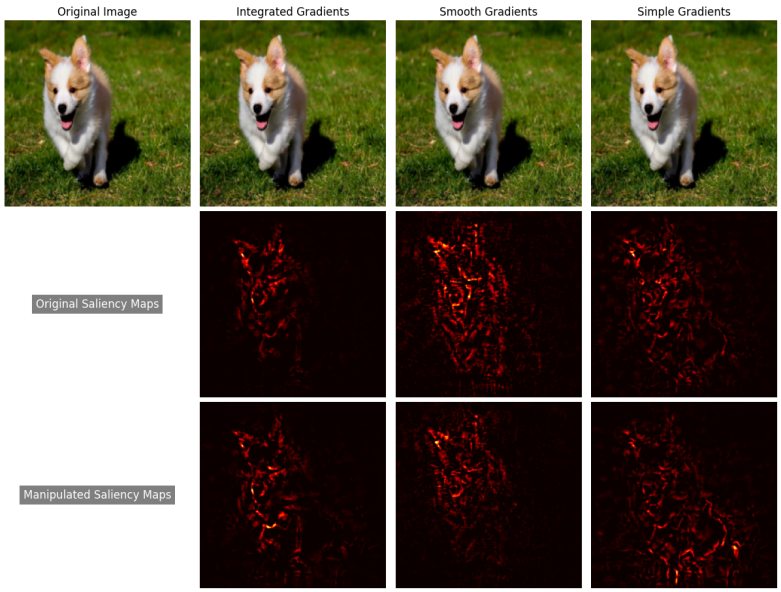

Fig. 4: Attack on our model

6 Closing Remarks and Future Research

We conducted an investigation into the vulnerability of Explainable AI (xAI) methods to adversarial attacks that manipulate explanations while preserving model predictions. To address this, we proposed NODA, a defense strategy that solves the challenge and also keeps the performance of model like accuracy well. Our approach improves the robustness of xAI methods, ensuring more reliable interpretations. In future work, we plan to extend NODA and apply NODA to complex models to further strengthen explanation security.

Acknowledgements

This research is partially funded by the Vingroup Innovation Foundation (VINIF) under the grant number VINIF.2021.JM01.N2.

References

1. Baniecki, H., Biecek, P.: Adversarial attacks and defenses in explainable artificial intelligence: A survey. Information Fusion **107**, 102303 (Jul 2024). https://doi.org/10.1016/j.inffus.2024.102303, http://dx.doi.org/10.1016/j.inffus.2024.102303
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848

3. Dombrowski, A.K., Anders, C.J., Müller, K.R., Kessel, P.: Towards robust explanations for deep neural networks (2020), https://arxiv.org/abs/2012.10425
4. Ghorbani, A., Abid, A., Zou, J.: Interpretation of neural networks is fragile (2018), https://arxiv.org/abs/1710.10547
5. Joo, S., Jeong, S., Heo, J., Weller, A., Moon, T.: Towards more robust interpretation via local gradient alignment (2022), https://arxiv.org/abs/2211.15900
6. Krizhevsky, A., Nair, V., Hinton, G.: CIFAR-10 and CIFAR-100 datasets. https://www.cs.toronto.edu/~kriz/cifar.html (2009), accessed: 2025-07-16
7. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks (2019), https://arxiv.org/abs/1706.06083
8. Nielsen, I.E., Dera, D., Rasool, G., Ramachandran, R.P., Bouaynaya, N.C.: Robust explainability: A tutorial on gradient-based attribution methods for deep neural networks. IEEE Signal Processing Magazine **39**(4), 73–84 (Jul 2022). https://doi.org/10.1109/msp.2022.3142719, http://dx.doi.org/10.1109/MSP.2022.3142719
9. Pearlmutter, B.A.: Fast exact multiplication by the hessian. Neural Computation **6**(1), 147–160 (01 1994). https://doi.org/10.1162/neco.1994.6.1.147, https://doi.org/10.1162/neco.1994.6.1.147
10. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. International Journal of Computer Vision **128**(2), 336–359 (Oct 2019). https://doi.org/10.1007/s11263-019-01228-7, http://dx.doi.org/10.1007/s11263-019-01228-7
11. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise (2017), https://arxiv.org/abs/1706.03825
12. Stassin, S., Corduant, V., Mahmoudi, S.A., Siebert, X.: Explainability and evaluation of vision transformers: An in-depth experimental study. Electronics **13**(1) (2024). https://doi.org/10.3390/electronics13010175, https://www.mdpi.com/2079-9292/13/1/175
13. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks (2017), https://arxiv.org/abs/1703.01365
14. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks (2014), https://arxiv.org/abs/1312.6199
15. Tamam, S.V., Lapid, R., Sipper, M.: Foiling explanations in deep neural networks (2023), https://arxiv.org/abs/2211.14860
16. Wang, Y., Zhang, T., Guo, X., Shen, Z.: Gradient based feature attribution in explainable ai: A technical review (2024), https://arxiv.org/abs/2403.10415
17. Wang, Z., Wang, H., Ramkumar, S., Fredrikson, M., Mardziel, P., Datta, A.: Smoothed geometry for robust attribution (2020), https://arxiv.org/abs/2006.06643