Xplain: Analyzing Invisible Correlations in Model Explanation

Kavita Kumari and Alessandro Pegoraro, *Technical University of Darmstadt;*
Hossein Fereidooni, *Kobil;* Ahmad-Reza Sadeghi, *Technical University of Darmstadt*

https://www.usenix.org/conference/usenixsecurity24/presentation/kumari

This paper is included in the Proceedings of the
33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

Open access to the Proceedings of the
33rd USENIX Security Symposium
is sponsored by USENIX.

Xplain: Analyzing Invisible Correlations in Model Explanation

Kavita Kumari
Technical University of Darmstadt
kavita.kumari@trust.tu-darmstadt.de

Alessandro Pegoraro
Technical University of Darmstadt
alessandro.pegoraro@trust.tu-darmstadt.de

Hossein Fereidooni
Kobil
hossein.fereidooni@kobil.com

Ahmad-Reza Sadeghi
Technical University of Darmstadt
ahmad.sadeghi@trust.tu-darmstadt.de

Abstract

Explanation methods analyze the features in backdoored input data that contribute to model misclassification. However, current methods like path techniques struggle to detect backdoor patterns in adversarial situations. They fail to grasp the hidden associations of backdoor features with other input features, leading to misclassification. Additionally, they suffer from irrelevant data attribution, imprecise feature connections, baseline dependence, and vulnerability to the "saturation effect".

To address these limitations, we propose *Xplain*. Our method aims to uncover hidden backdoor trigger patterns and the subtle relationships between backdoor features and other input objects, which are the main causes of model misclassification. Our algorithm improves existing path techniques by integrating an additional baseline into the Integrated Gradients (IG) formulation. This ensures that features selected in the baseline persist along the integration path, guaranteeing baseline independence. Additionally, we introduce quantitative noise to interpolate samples along the integration path, which reduces feature dependency and captures non-linear interactions. This approach effectively identifies the relevant features that significantly influence model predictions.

Furthermore, *Xplain* proposes sensitivity analysis to enhance AI system resilience against backdoor attacks. This uncovers clear connections between the backdoor and other input data features, thus shedding light on relevant interactions. We thoroughly test the effectiveness of *Xplain* on the Imagenet and the multimodal domain of the Visual Question Answering dataset, showing its superiority over current path methods such as Integrated Gradient (IG), left-IG, Guided IG, and Adversarial Gradient Integration (AGI) techniques.

1 Introduction

Black box neural networks are widely deployed in domains such as disease detection [5], image synthesis [34], protein folding [25], and backdoor analysis [20, 26]. However, the model's complexity and high-dimensional dynamics, driven by numerous parameters and non-linear interactions between input data features, make interpreting their decision-making processes challenging. Consequently, analyzing the backdoor features contributing to the model misclassification presents additional security concerns.

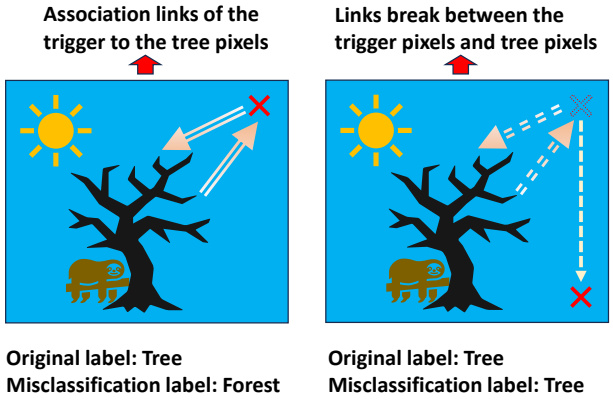


Figure 1: This example highlights the importance of capturing links between trigger features and their surrounding features in the input sample. Both trigger features and their relationships with other features contribute to misclassification. The trigger in the top-right corner affects the central tree (as shown by the arrows), causing misclassification from "Tree" to "Forest". However, moving the trigger to the bottom right disrupts this relationship, which does not cause misclassification.

To detail model predictions, attribution methods have emerged as a crucial tool, offering explanations by quantifying the contribution of each input feature to a decision [39]. Attribution methods broadly fall into perturbation-based methods [33, 45], backpropagation-based methods [36, 41], and gradient-based methods [39, 42].

Our paper focuses on path gradient-based methods, known for their axiomatic foundations and model-agnostic nature [1]. These methods evaluate feature importance by incrementally

increasing their "presence" from the baseline (initial feature values) to the input sample (final feature values) to measure their impact on model output. A limitation in these techniques is the introduction of noise from irrelevant features (features not contributing to model prediction) during gradient computation, which can stem from highly correlated features in the input [17,22,24,40], baseline selection [15], and the saturation problem [24].

The baseline issue arises from interpolating between a baseline with "missing" features and the feature's actual value in the input sample, neglecting the feature values used in the baseline. For example, a constant black image baseline may not emphasize significant black pixels in the final explanation attribution map. The saturation problem occurs when input feature gradients become negligible, so shifting features (from baseline to input sample) does not change the model prediction. Alternatively, the model output probability does not change for the incremental interpolation samples along the integration path.

Consequently, these limitations obstruct the interpretability of attribution methods in security. Analyzing backdoor triggers in the input, which prompt a trojan model to produce a specific misclassification, becomes challenging [21]. Figure 1 demonstrates the importance of feature interactions in the model misclassification. Both trigger pixels and their relationship (shown by the solid arrows) with the tree pixels contribute to the misclassification of the model prediction from "tree" to "forest". However, this relationship is detached when the trigger is moved to the bottom-right corner (shown by checked arrows). Consequently, the classification label remains the same, i.e., tree. Furthermore, a defense mechanism that cannot detect the trigger and its relationships will not effectively prevent adversaries from bypassing it. Triggers and their connections are crucial for misclassification, so a robust attribution method should identify them during inference.

Various solutions have been proposed to address the highly correlated features [40], select better baselines [8, 43], and limit the saturation effect impact [13, 15, 24, 28], however, these approaches, while addressing specific problems, do not necessarily enhance the overall quality of attributions, as we demonstrate in Section 6. Therefore, we encountered an open challenge of updating the path gradient methods, specifically Integrated Gradients (IG) [42], such that the problems mentioned above do not limit them. The rationale behind selecting IG is that it is the most prominent path method in the literature. Additionally, we select the other SoTA path methods in the literature: Left-IG (LIG) [24], Guided IG (GIG) [15], and Adversarial Gradient Integration (AGI) [28]. These methods are an update on IG but still suffer from the same limitations (demonstrated in Section 6).

Our Goals and Contributions. To tackle the challenges outlined above, we present the design and implementation of *Xplain*, designed to enhance the understanding of the relationship between the input sample features. *Xplain* incorporates two enhancements to Integrated Gradients (IG), a widely used gradient-based method in the field.

Firstly, our approach tackles the complexities in input data by introducing quantitative noise, disrupting complex connections between input features and predicted labels. The addition of noise proves beneficial in scenarios where input features are correlated. Secondly, we eliminate the reliance on a particular baseline in the baseline selection. We append an extra baseline to each interpolation sample along the gradient pathway. This simple yet novel technique removes the need for specific baseline choices (see Section 5.2).

The intuition behind adding quantitative noise is to integrate rapid small alterations in the model prediction (because of the addition of noise) along the gradient path. It automatically tackles the saturation issue because the gradients of the interpolated samples gradients keep on adjusting (i.e., never saturate). This also exposes more complex, subtle, and non-trivial correlations among input data features. Next, the intuition behind adding an extra baseline is to integrate the effect of baseline features (missing features) in the gradient computation along the integration path. It guarantees the effect of baseline features persists in the interpolated samples.

Additionally, these updates make *Xplain* remain unaffected by different decoy triggers and their positioning. More details are provided in Section 6. In contrast, other path techniques (IG, LIG, GIG, and AGI) cannot discover the hidden association links between the trigger features and other features in the input sample [20]. Hence, if the trigger is moved to a different location (Figure 1), these methods still inaccurately deem the moved trigger significant for the model, even without misclassification. Our approach, however, can accurately identify the true impact of these artifacts (as analyzed in Section 6.4). This enhancement in explainability offers a valuable tool for identifying and mitigating Trojaned image data samples. Finally, we propose a sensitivity analysis framework to strengthen AI system resilience against backdoor attacks. This analysis evaluates the effect of possible backdoor triggers on model predictions, examining interactions between relevant features for security scrutiny. The aim is to offer insight into how backdoor patches and their different associations collectively influence model predictions. Further analysis is detailed in Section 5.5.

Therefore, integrating noise insertion, baseline independence, and sensitivity analysis forms a robust XAI analysis framework. This is essential for critical applications where backdoor attacks and trigger recognition are major concerns.

In summary, our main contributions are as follows:

- We present *Xplain*, a novel attribution methodology to fully grasp the hidden associations or links of backdoor features with other features in the queried input data sample, which are the root causes for the model misclassification.

- *Xplain* methodically resolves the issues of noisy relevance scores, notably when data samples have correlated features. Additionally, *Xplain* deploys a simple update on the IG technique to mitigate the saturation effect and baseline selection. It outperforms the existing path-gradient techniques by determining the pertinent features that impact model predictions, as shown in Section 5.

- We propose a sensitivity analysis framework to fortify the robustness of AI systems against backdoor attacks, which helps to quantify the pertinent features and their relationships that can be leveraged in detecting backdoors, as detailed in Section 5.

- We extensively compare *Xplain* with popular gradient-based methods like Integrated Gradients [42], Left-IG (LIG) [24], Guided IG (GIG) [15], and Adversarial Gradient Integration (AGI) [28] on both Imagenet and text datasets. We aim to evaluate the interpretability of these methods in detecting various backdoor triggers in the input. Since triggers are crucial for misclassification, a robust XAI method should automatically identify their presence and association with other features during inference, as shown in Section 6.

2 Background

2.1 Neural Network

The training of a Neural Network (NN) F takes samples from a domain $\mathcal{D}$ as input and returns predictions from the set $\mathcal{L}$. Learning a function $F_\theta : \mathcal{D} \rightarrow \mathcal{L}$ is dependent on the parameters θ of the NN. The goal of the adversary $\mathcal{A}$ is to inject a backdoor into the aggregated model, making F predict an adversary-chosen label $l_\mathcal{A} \in \mathcal{L}$ for all samples containing one or more specific pattern (called *triggers*), establishing the *trigger set* $\mathcal{D}_\mathcal{A} \subset \mathcal{D}$.

2.2 Integrated Gradients

Given an input data point $\overrightarrow{x} \in \mathcal{D}$ and a classification model F_θ, an explanation method, denoted by $\mathcal{H}$, takes the form of an influence (or attribution) vector. This vector elucidates the model's decisions based on the contributions of each feature. The i^{th} element of this vector, $\mathcal{H}_i(\overrightarrow{x})$, signifies the extent to which the i^{th} feature impacts the predicted label y for the data point $\overrightarrow{x}$. Integrated Gradients [42] are derived by accumulating gradients calculated at all points along a linear path from a baseline $\overrightarrow{x}'$ (often chosen as $\overrightarrow{x}' = \overrightarrow{0}$) to the actual input $\overrightarrow{x}$ [42]. Essentially, integrated gradients are the path integral of the gradients along a straight-line path from the baseline $\overrightarrow{x}'$ to the input $\overrightarrow{x}$. The integrated gradient for an input $\overrightarrow{x}$ and baseline $\overrightarrow{x}'$, which we denote as $\mathcal{H}_{IGRAD}$, for the i-th feature, is defined as follows:

$$\mathcal{H}_{IGRAD}(\overrightarrow{x})_i = (x_i - x_i^{'}) \cdot \int_{\alpha=0}^{1} \frac{\partial F_\theta(\overrightarrow{x}' + \alpha(\overrightarrow{x} - \overrightarrow{x}'))}{\partial x_i} \mathrm{d}\alpha$$

Here, α is the interpolation factor that determines the precision to which interpolation samples ($\overrightarrow{x}' + \alpha(\overrightarrow{x} - \overrightarrow{x}')$) between the baseline $\overrightarrow{x}'$ and the input $\overrightarrow{x}$ will be computed. Hence, if the model output (logits) does not change for these interpolation samples, i.e. while moving α along the path, the corresponding gradients are not very important for the model's decision-making. Left-IG (LIG) [24] was designed to mitigate this issue (detailed below).

IG also suffers from spurious or noisy attribution maps as it does not consider the spatial relationships between features in an input. The reason is that IG does not account for the spatial arrangement or the context provided by neighboring features, which can be crucial for understanding the overall contribution of the features to the model prediction. Guided IG (GIG) [15] was proposed to mitigate this issue (detailed below).

Lastly, IG also suffers from the choice of baseline. If the baseline does not adequately capture the critical features in the input sample, the resulting explanations might not be reliable or easy to understand. Hence, different baselines can yield different attributions for the same input, interpreting results as ambiguous and less robust. Adversarial Gradient Integration (AGI) [28] was proposed to mitigate this issue (detailed below).

2.3 Left Integrated Gradients

To separate the contribution of saturated areas (i.e., areas of the integral path where the model output changes minimally) from unsaturated areas (i.e., areas of the integral path where the model output changes substantially), Left-IG [24] splits the approach described in Section 2.2 into two regions. Considering only the leftmost region, representing the substantial increases in the function characterized by F_θ, while the other saturated regions correspond to minimal changes in F_θ. For a given threshold φ, Left-IG wants to determine the minimum value of α such that the target output $F(\overrightarrow{x} + \alpha \cdot (x - x^{'}))$ exceeds $\mathcal{H}_{IGRAD}$ by φ along the linear path.

2.4 Guided Integrated Gradients

To obtain attributions features with the lowest absolute value of partial derivatives, Guided IG [15] integrates gradient along an adaptive path determined by the input $\overrightarrow{x}$, baseline $\overrightarrow{x}'$ and model F_θ. This adaptive path is defined from the baseline towards the input, moving in the direction of features with the lowest absolute value of partial derivatives. At each step of the integration, Guided IG selects pixels with partial derivatives lower than a specific threshold φ ($|\partial x_i| < \varphi$) and moves only that subset closer to the intensity in the input image, leaving

all others unchanged, until there are no longer candidates for selection.

2.5 Adversarial Gradient Integration

Adversarial Gradient Integration [28] replaces the baseline $\overrightarrow{x'}$ by calculating the gradient of adversarial examples. Given the input data point $\overrightarrow{x}$, the approach aims to explain the model's prediction by discriminating against the false classes instead of focusing only on the correct classification of the true class. With an inverse relationship between the gradient of the adversarial examples on the false class and the attribution of the true class. This methodology is based on the notion that while we commonly describe a model as classifying inputs, an alternative viewpoint is that the model excludes inputs from the other categories.

3 Threat Model

We consider a distinct threat model concerning adversaries seeking to stray from the normal behavior of the trained model during inference through the use of accurately crafted backdoor samples.

Adversary Capability: In our threat model, we assume an attacker $\mathcal{A}$ leverages access to the machine learning dataset. $\mathcal{A}$ inserts backdoors into the training dataset before training and does not necessarily need to be involved in the training procedure. For instance, they do not need to know how many epochs training will last, what the hyperparameters of the network will be, or what preprocessing steps may be applied, except for what they may deduce from the nature of the training set. Thus, $\mathcal{A}$ knows the existence and functionality of these backdoors. $\mathcal{A}$ can exploit these backdoors to alter the model's predictions on specific inputs without the need for direct access to the model architecture or parameters. Finally, we assume that $\mathcal{A}$ may acquire knowledge of the workings of the model through various means. However, the specifics of how they gain this knowledge are not within the scope of our investigation.

Attack Objectives: The primary objective of an adversary is to manipulate model predictions on targeted inputs without raising suspicion. Additionally, an adversary aims to undermine the integrity and reliability of the model by integrating backdoors of different types in different locations of the input sample, fulfilling the following criteria. First, the adversary seeks to evade detection by conventional security mechanisms, as the existing mechanisms may not be able to extract the complex non-linear relationship between the trigger features and other features of the input data sample. Second, to further compromise the reliability of any employed defense, the model adversaries divert the defenses into erroneously flagging safe samples.

4 Problem Description

This section thoroughly explores the constraints of gradient-based techniques, mainly focusing on the Integrated Gradients (IG) method. Additionally, we present an extensive analysis explaining how our approach successfully addresses and surpasses these limitations.

4.1 Limitations of IG

Lack of Spatial Information: The lack of spatial information in IG refers to its inability to consider the spatial relationships between features in an input. It cannot analyze a specific feature's spatial arrangement and local interactions among features. This limitation presents challenges in cases where a complex relationship exists between different input sample features. As a result, IG struggles to discern which features hold more meaningful information or are more relevant for the model prediction. Below, we detail key factors for this limitation.

First, IG treats each feature of the input sample independently when calculating attributions. Thus, it does not account for the spatial arrangement or the context provided by neighboring features, which can be crucial for understanding the overall contribution of the features to the model prediction. This independent feature assumption fails to capture how features interact, leading to an incomplete analysis of the model's decision-making process.

Second, IG's reliance on a linear interpolation path from a baseline to the input image further exaggerates this issue. It does not consider how spatially related features might change together along more complex paths, potentially missing critical spatial dynamics and interactions. For instance, a feature's contribution might be over- or under-estimated because the linear path does not reflect the changes in feature interactions.

Third, IG provides a localized attribution score for each feature, indicating its contribution to the model's prediction. However, it does not inherently capture how groups of features interact to form meaningful patterns that influence the model's decision. This localized attribution can misrepresent the significance of individual features when their importance is derived from their spatial context.

For example, consider a model distinguishing between cats and dogs based on features like ears (x_1) and fur color (x_2). Suppose a specific combination of feature values triggers a misclassification as a dog. Assuming feature independence, IG may highlight only the ear or fur color, even if both are manipulated. This results in distorted attributions, misrepresenting the true importance of combined individual features. A single feature might appear highly important, but its significance could be due to its relationship with surrounding features, which IG does not explicitly capture.

In summary, IG's lack of spatial information determines whether it can miss or misrepresent the importance of spatially

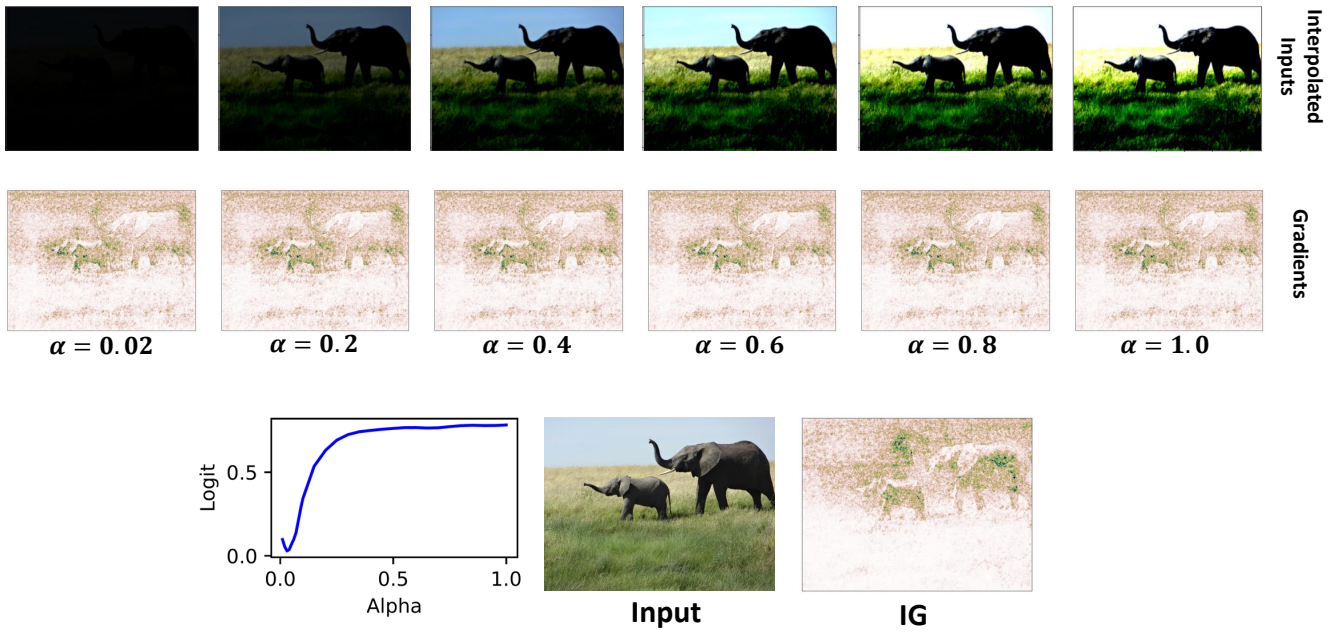


Figure 2: The diagram depicts the IG attribution technique, discrepancies in attribution scores, and saturation phenomena in path integrals. The upper row displays interpolated inputs, while the second row showcases the corresponding gradients. The IG attribution map (depicted in the bottom-right) represents the gradient average. The third row features the logit-α curve, delineating saturation areas.

correlated features and structures in the input data sample.

Baseline Selection: The initial features (baseline) values are crucial in the IG method. If the baseline does not adequately capture the critical features in the input sample, the resulting explanations might not be reliable or easy to understand.

IG draws inspiration from cooperative game theory, specifically the Aumann-Shapley value, which quantifies the contribution of a group in a game by measuring how the game's value increases with more group members. In machine learning, IG assesses a feature's importance or missingness, such as x_i, by determining how much the network's output grows as this feature is incremented from the baseline value to the input sample value.

An inappropriate baseline can lead to misleading attributions, especially when fixed feature values have unintended meanings. It gets worse when we consider the difference from the baseline term $(\overrightarrow{x} - \overrightarrow{x}')$. For example, suppose a constant black image is the baseline. In that case, IG will not emphasize black pixels, even if they are essential for the object. IG does not care about the chosen baseline feature values.

In summary, IG's attributions are sensitive to the choice of baseline. Different baselines can yield different attributions for the same input, interpreting results as ambiguous and less robust. This sensitivity complicates selecting a universally appropriate baseline, especially for diverse datasets or models with varied input characteristics.

Saturation Effect: The saturation problem in IG is when the gradients of interpolated sample features become negligible along the path from the baseline to the input, even though the network heavily depends on these features. Thus, it results in attributions that underestimate certain features' importance, failing to accurately reflect their contribution to the model's output. Consequently, shifting features in the input sample (from baseline to actual value) often fails to alter the model predictions.

To introduce and understand the saturation effect problem within path integrals, we examine the performance of the IG attribution method in 2. As observed in Figure 2, the output logits (model prediction) (logit-Alpha curve) of the interpolated samples $(\overrightarrow{x}' + \alpha(\overrightarrow{x} - \overrightarrow{x}')$ for $\alpha \in [0,1])$ starts to saturate after a certain threshold. α is the interpolation factor defined in Section 2.2. Thus, changes in these features do not significantly impact the model's output, leading to their importance being underestimated.

It can be observed that (i) gradients from the saturation regions are of low quality, and (ii) gradients from the decision region are of high quality. The conclusion is relatively straightforward. The corresponding gradients in the saturation regions are not crucial to the decision of the model. We study the quality of the computed gradients concerning the decision and saturated regions of the path integral.

It prompts a fundamental question: Can we design an integral path that mitigates the problems of path explanation methods presented above?

5 System Design

This section provides an in-depth examination of the structural configuration of our proposed XAI technique, *Xplain*. First, it presents the high-level idea of *Xplain*, then elaborates on the method's core components and their corresponding functionalities.

5.1 High-Level Idea

We introduce a simple yet novel technique for generating attributions that circumvent challenges associated with feature interconnections or lack of spatial information, baseline selection, and saturation effects. Specifically, our approach integrates the following two key strategies.

First, we systematically add the "speckle" or noise along the gradient trajectory of the input data, transitioning from the baseline input toward the original input's noisy variant. The intuition behind adding quantitative noise is integrating rapid small alterations in the model prediction (because of the addition of noise) along the gradient path. It automatically tackles the saturation issue because the gradients of the interpolated samples gradients keep on adjusting (i.e., never saturate). This also exposes more complex, subtle, and non-trivial correlations among input data features. This technique aims to improve the interpretability of the model's predictions by introducing controlled perturbations that aid in understanding how features contribute to the model's decision-making process. This establishes a straightforward link between the input data and the model's prediction and, at the same time, adjusts the output logit values as the $\Delta\alpha$ moves along the path to account for the important features of the model prediction.

Second, we incorporate an additional baseline to the "increasing missing feature" values in the interpolation samples along the gradient pathway. When generating interpolated images with varying alpha (α) values between 0 and 1, the initial feature values present in the baseline are not adequately represented in the final attribution. To address this, our method integrates an additional baseline to ensure the effect of baseline features persists in the interpolated samples or the final attribution map.

Our framework effectively unravels how features interact, mitigating the three constraints commonly seen in path explanation methods. Our method focuses on input points along the interpolation path where feature connections are minimized. Consequently, it automatically addresses the issues of the baseline selection and the saturation effects by consistently modifying the scaled image. As observed in Figure 3, the output logits (model prediction) (logit-Alpha curve) of the interpolated samples ($\overrightarrow{x}' + \alpha(\overrightarrow{x} - \overrightarrow{x}')$ for $\alpha \in [0,1]$) automatically starts to minimize the effect of saturation. Thus, the impact of features that do not significantly impact the model's output is optimally diminished. Note: as mentioned earlier, we are adding quantitative noise to the interpolation samples. Thus, as the α increases, the explanation visualization becomes a bit noisy. However, the focus is to highlight all the relevant pixels contributing to model predictions, as seen in Figure 3.

Finally, we conduct a sensitivity analysis framework to strengthen AI system resilience against backdoor attacks. It computes the strength of how much one feature impacts the other features or provides insights into the degree to which the behavior of one feature can be predicted from the behavior of another. Thus, it illustrates how the sensitivities (impact of $\overrightarrow{x}'$ on $\mathcal{H}_{IGRAD}$) of the features share variance with each other.

The efficacy of our approach is substantiated through comprehensive experiments, notably in pinpointing the relevant features underpinning model predictions. Further insights into these evaluations can be found in Section 6.

5.2 Interpolation Samples Computation

This section illustrates how the IG formulation (Section 2) is updated to integrate the additional baseline and the speckle noise to mitigate the three limitations of the existing path methods.

Noise Integration: To accomplish this, we introduce "speckle" noise at each interpolation step of IG, i.e., for each α. The goal is to ensure the gradients of the interpolated samples gradients keep on adjusting (i.e., never saturate). This noise is drawn from the standard normal distribution of $\overrightarrow{x}$ and is evenly distributed across the entire feature space of $\overrightarrow{x}$. This noisy input is then added to the specific input sample. These steps aim to reduce the direct dependency between input features, as shown in Equation 1 and illustrated in Figure 4 for different α. We use "speckle" noise instead of Gaussian noise (or other noises) because it is evenly spread across the feature space of the input query, whereas Gaussian noise is centered around the mean value. It is given by:

$$g(\overrightarrow{x_n}) = \overrightarrow{x} + \mathcal{N}(\mu = 0, , \sigma = 1) \times \overrightarrow{x}$$

for mean $\mu = 0$ and standard deviation σ. Then, this noisy update on $\overrightarrow{x}$ is used to compute interpolation samples and is given by: $\overrightarrow{x}' + \alpha \times (\overrightarrow{x} - \overrightarrow{x}' + g(\overrightarrow{x_n}))$.

Baseline Integration: To ensure that the initial feature values in $\overrightarrow{x}'$ are adequately represented in the final attribution, *Xplain* integrates an additional baseline in the interpolated samples and is given by:

$$p = \overrightarrow{x}' + \alpha \times (\overrightarrow{x}' + \alpha \times (\overrightarrow{x} - \overrightarrow{x}' + g(\overrightarrow{x_n}))) \quad (1)$$

where $\overrightarrow{x}' + \alpha \times (\overrightarrow{x} - \overrightarrow{x}' + g(\overrightarrow{x_n}))$ are the interpolated images for different values of $\alpha \in [0,1]$ (same as in IG). In Equation 1, the same baseline $\overrightarrow{x}'$ is augmented twice with

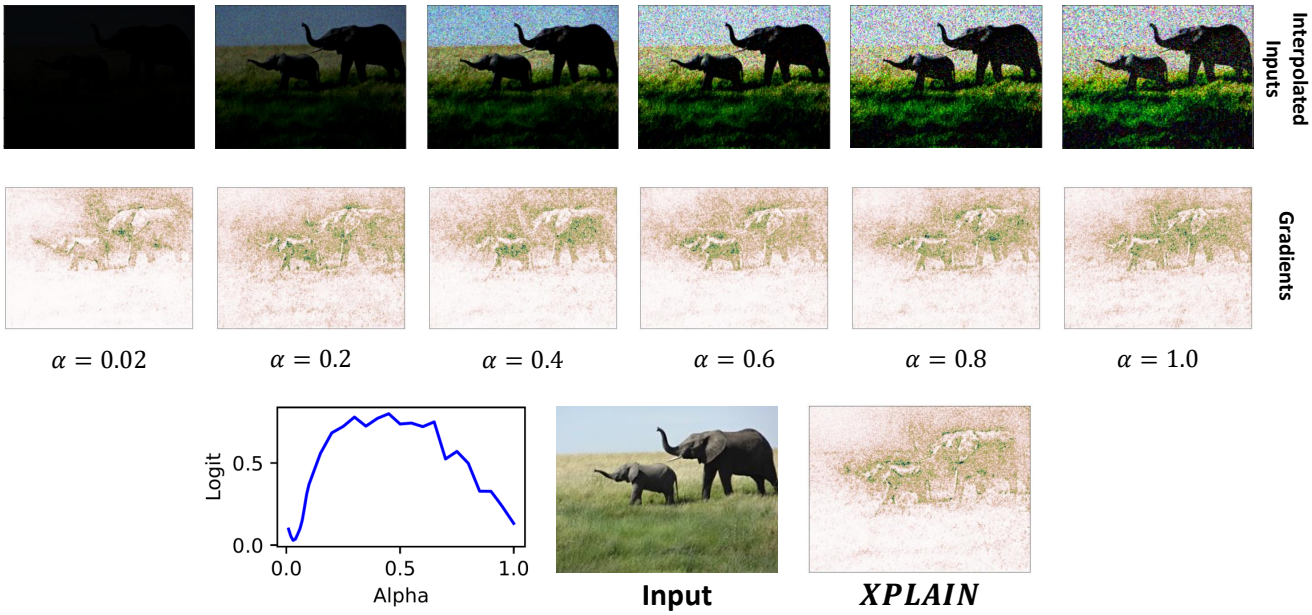
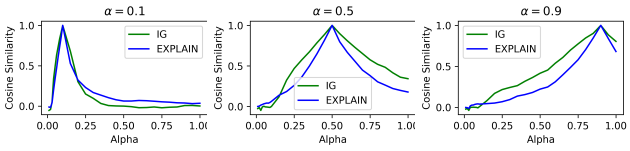


Figure 3: A visualization of how *Xplain* integrates the noise into the selection of the interpolated images, thereby limiting the impact of noise and saturation effect on the attribution scores. The upper row displays interpolated inputs, while the second row showcases the corresponding gradients. The *Xplain* attribution map (depicted in the bottom-right) represents the gradient average. The third row features the logit-α curve, delineating how *Xplain* diminishes the impact of saturation effect on the relevance of the features.



(a) $\alpha = 0$ (b) $\alpha = 0.5$ (c) $\alpha = 1$

Figure 4: The plots illustrate how the gradients along the attribution path are related. In each subplot, the cosine similarity between gradients at three α points equal to 0.1, 0.5, and 0.9 is shown compared to gradients of all the other steps of the integration path. This analysis is performed for both Integrated Gradients (IG) and our method *Xplain*.

the noisy input sample $g(\overrightarrow{x_n})$ in the IG formulation. The motivation is two-fold: first, to make sure the baseline feature values persist along the integration path, and second, to make sure an integral path is established from $\overrightarrow{x}'$ ($\alpha = 0$) to the noisy input sample of $\overrightarrow{x}' + \overrightarrow{x} + g(\overrightarrow{x_n})$.

5.3 Gradient Computation

This module parallels the approach presented in [42] but with a slight variation: the input to IG is p (Equation 1). The updated *Xplain* formulation is as follows:

$$\mathcal{H}_{Xplain}(\overrightarrow{x})_i = (x_i - x_i^{'}).\int_{\alpha=0}^{1} \frac{\partial F_\theta(p)}{\partial x_i} \mathrm{d}\alpha \qquad (2)$$

where, $\mathcal{H}_{Xplain}(\overrightarrow{x})_i$ represents the i-th feature's attribution or contribution to the model prediction. By integrating these processes, *Xplain* presents an innovative approach to explainable AI, enhancing the interpretability of complex machine learning models.

The *Xplain* attribution method aligns with IG as depicted in Section 2.2, except with the distinction that the model input for $\alpha \in [0,1]$ is computed using Equation 2. By computing the noisy interpolated inputs, *Xplain* effectively controls the computation of crucial query samples that substantially impact the attribution map of *Xplain* or cause significant changes in the model's logits (model's outputs probabilities). As a result, the explanation vector $\mathcal{H}_{Xplain}$ accurately identifies all relevant features. This is accomplished by mitigating the baseline dependence and the saturation effect, which in turn implicitly accommodates the spatial analysis of the features.

Figure 3 demonstrates *Xplain*'s functionality. The top row shows interpolated inputs computed using Equation 1. The second row illustrates the impact of *Xplain* on gradients from noisy interpolated inputs for $\alpha \in [0.02, 0.2, 0.4, 0.6, 0.8, 1.0]$. The bottom-left plot shows the logit-α curve for the input image, while the bottom-right plot displays gradients calcu-

lated by *Xplain* for these inputs. The noisy interpolated image computation prevents input feature saturation, maintaining dynamic alternation.

The logit-α curve shows that *Xplain* mitigates the saturation by reducing the curve's slope. This ensures that gradients for input samples are minimally influenced by noise. Consequently, attributions derived using *Xplain* exhibit significantly lower noise levels than the existing path methods, as demonstrated in Section 6.

5.4 Conformity with Axiomatic Principles

Our methodology, *Xplain*, is grounded in the axiomatic properties fundamental to the path integral attribution method. We highlight its alignment with select principles delineated in [42] and elucidate the axiomatic foundations upheld by *Xplain*.

- **Completeness Axiom:** Formally, the Completeness axiom is described by:

$$\sum_{i=0}^{i=n} \mathcal{H}_{Xplain}(\overrightarrow{x})_i = F_\theta(\overrightarrow{x}) - F_\theta(\overrightarrow{x}') \tag{3}$$

  This dictates that the aggregate of attributions should equal the difference between the function's responses for the input and the baseline.

  In the case of *Xplain*, we incorporate baseline and controlled noise into the input query while creating interpolated images along the path integral. As a result, *Xplain* also upholds the Completeness principle, albeit with a specific adjustment:

$$\sum_{i=0}^{i=n} \mathcal{H}_{Xplain}(\overrightarrow{x})_i = F_\theta(p) - F_\theta(\overrightarrow{x}') \tag{4}$$

- **Sensitivity Axiom (a):** This principle suggests that if a feature's variation between the input and baseline leads to different predictions, it should have a nonzero attribution. Because Xplain is in line with the Completeness principle, it also naturally follows the Sensitivity (a) principle.

- **Implementation Invariance Axiom:** This concept emphasizes the importance of consistent attributions when comparing two networks that work similarly. Since *Xplain* operates without concerning itself with the specific inner workings of the network, it naturally meets this requirement. Additionally, the attributions produced by *Xplain* adhere to Implementation Invariance, as they rely solely on the gradients of the function that the network represents.

5.5 Sensitivity Analysis

Below, we detail a framework that provides a holistic understanding of feature interactions by computing the degree to which the behavior of one feature can be predicted from the behavior of another. First, we compute the feature sensitivity, which measures how much a feature affects the attribution computed by *Xplain*. Then, we incorporate how much the variance is shared between each pair of feature sensitivities by computing the coefficient of determination (denoted as R^2) [27].

- **Compute Feature Sensitivities:** The sensitivity values S_{x_i} and S_{x_j} are computed by dividing the attribution values $A(x_i)$ and $A(x_j)$ by their respective feature values x_i and x_j. Normalizing by the feature value ensures that the sensitivity reflects the contribution per unit of the feature, providing a more meaningful measure of impact. Thus, we compute the sensitivity values S_{x_i} and S_{x_j} for each feature x_i and x_j using the following formula:

$$S_i = \frac{A(x_i)}{x_i}$$
$$S_j = \frac{A(x_j)}{x_j}$$

- **Variance Sharing Computation:** Next, we compute how much variance is shared between two features x_i and x_j using the coefficient of determination. The coefficient of determination (R^2) is calculated to quantify the proportion of variance in the sensitivity of feature x_i that the sensitivity of feature x_j can explain. In other words, it measures how much the sensitivities of the two features co-vary. Coefficient of determination R^2 between the sensitivity values of feature x_i and feature x_j is computed as:

$$R^2(x_i, x_j) = \frac{cov(S_{x_i}, S_{x_j})}{var(S_{x_i}) \cdot var(S_{x_j})}$$

  where, $cov(S_{x_i}, S_{x_j})$ is the covariance between S_{x_i} and S_{x_j}. $var(S_{x_i})$ and $var(S_{x_j})$ are the variances of S_{x_i} and S_{x_j}, respectively. $R^2(x_i, x_j)$ represents the proportion of variance in the sensitivity of feature x_i explained by the sensitivity of feature x_j. R^2 provides insights into the degree to which the behavior of one feature can be predicted from the behavior of another. Thus, it indicates a strong linear relationship, suggesting that the sensitivities of the features share a significant amount of variance.

- **Combine with Cosine Similarity and Euclidean Distance:** This step involves incorporating R^2 into the analysis by combining it with cosine similarity and Euclidean distance metrics. The cosine similarity measures the directional alignment of sensitivities, while the Euclidean

distance captures the magnitude of the difference between sensitivities.

- For cosine similarity, we do the following:

$$\text{C\_sc}(x_i, x_j) = \text{c\_si}(S_{x_i}, S_{x_j}) \times R^2(x_i, x_j)$$

where $\text{c\_si}(S_{x_i}, S_{x_j})$ measures the alignment or similarity in the direction of S_{x_i} and S_{x_j}.

- For Euclidean distance, we do the following:

$$\text{E\_sc}(x_i, x_j) = \text{e\_si}(S_{x_i}, S_{x_j}) \times (1 - R^2)(x_i, x_j)$$

where $\text{e\_si}(S_{x_i}, S_{x_j})$ measures the magnitude of the difference between S_{x_i} and S_{x_j}, and $(1-R^2)(x_i, x_j)$ emphasizes feature pairs with low shared variance.

Finally, we combine the scores from cosine similarity and Euclidean distance to obtain a final integrated score, given by:

$$F\_sc(x_i, x_j) = \text{C\_sc}(x_i, x_j) + \text{E\_sc}(S_{x_i}, S_{x_j})$$

The final score represents a comprehensive measure that considers the similarity in sensitivity direction and the shared variance in sensitivity values between features x_i and x_j. This score combines multiple aspects of feature interaction, providing a balanced measure that considers both the direction and magnitude of sensitivities and shared variance.

6 Evaluation

6.1 Setup

In this section, we thoroughly evaluate our proposed method *Xplain*. To conduct these evaluations, we utilize the PyTorch framework [30] on a server with the following specifications: 4 NVIDIA RTX 8000 GPUs, each with 48GB of memory, an AMD EPYC 7742 processor, and a total of 1024 GB of main memory. Our chosen dataset is the 2012 validation set of ImageNet [35] and multimodal domain of the Visual Question Answering dataset. In this dataset, the model is presented with both image and text samples, with the text input containing inquiries regarding the content of the image input. The model implementation is based on the work of Kazemi et al. [16] using an LSTM model trained on the VQA 2.0 [9] dataset, in conjunction with a ResNet [10] fine-tuned on the MS COCO dataset [19].

. For a thorough evaluation of our method's effectiveness, we compare it against several well-established path methods, namely IG [42], Left-IG [24], Guided IG [15], and Adversarial Gradient Integration [28]. For the implementation of IG, we employ Captum [18], while we source Left-IG, Guided-IG, and Adv-G from their respective repositories [14,23,29]. Next, we use three ResNet [10] models: ResNet18, ResNet101, and ResNet152, all of which have been pre-trained on the ImageNet [35] dataset. Next, we conduct quantitative, qualitative, and trigger analyses to evaluate the attributions of clean and triggered samples. We also perform a qualitative analysis on the multimodal domain using the ResNet20 and the MS COCO [19] dataset for the image portion, and an LSTM (Long Short-Term Memory) [11] Neural Network trained on the text dataset VQA 2.0 [9]. This comparison allows us to assess the relative performance and effectiveness of the innovative approach we present in this paper.

In quantitative analysis, we first compute the efficacy of *Xplain* and compare it against the four other explanation methods. To determine the efficacy of *Xplain* attribution behavior, we conduct four tests utilizing three insertion methods and one deletion method, following the techniques introduced by the authors of RISE and XRAI [13, 32]. The results are detailed in Table 1. Regarding the qualitative analysis, we showcase a subset of four examples featured in Figure 5, employing the ResNet101 model.

In trigger analysis, we analyze the capability of our approach in identifying triggers within poisoned samples. In this analysis, we conduct multiple attribution tests on poisoned ResNet models. These tests encompass various scenarios, including those featuring one or two triggers that can either occupy 0.45% (small), or 0.9% (big) of the sample area, with shapes that range from crosses to more complex constructs, and multiple color variations.

Additionally, we evaluate scenarios where decoy triggers were employed to deceive the explainable methods. For example, a model is trained on samples containing different patches, with only specific combinations of patches triggering a misclassification, and samples with other patches do not trigger any abnormal behavior. This deception aims to divert attention towards features that aren't accountable for the model's behavioral change, concealing the real correlation between triggers. Then, we also perform a comparative analysis of *Xplain* with other approaches when the image data has been integrated with different backdoors of different shapes, colors, and sizes. More details are provided in the following sections.

Below, we compute two metrics to compute the accuracy of the backdoored model, which is used to compute attributions of the trigger dataset.

Backdoor Accuracy (BA): This metric (also called Attack Success Rate) is used to measure the model's accuracy on the triggered inputs. Specifically, it measures the fraction of true triggered samples where the model predicts the adversary's chosen label.

Main Task Accuracy (MA): This metric measures the model's accuracy on its benign, main task. It represents the fraction of benign inputs for which the model provides correct predictions.

Metric	Model	IG	Left-IG	Guided-IG	Adv-GI	*Xplain*
SIC (↑)	ResNet18	55.0	67.2	45.9	56.9	**79.9**
	ResNet101	68.5	67.7	53.7	62.3	**74.0**
	ResNet152	70.2	70.1	47.2	63.1	**82.1**
AIC (↑)	ResNet18	58.8	53.0	45.0	54.9	**59.7**
	ResNet101	63.8	66.1	53.9	64.2	**81.2**
	ResNet152	59.8	70.6	68.4	75.8	**75.9**
Insertion (↑)	ResNet18	19.7	22.1	17.2	21.4	**27.6**
	ResNet101	16.1	13.8	14.0	16.0	**24.2**
	ResNet152	31.6	29.5	42.8	31.7	**55.9**
Deletion (↓)	ResNet18	8.8	17.8	8.9	15.5	**4.3**
	ResNet101	8.6	9.9	7.1	18.3	**6.1**
	ResNet152	23.4	18.4	11.5	14.2	**10.7**

Table 1: Quantitative comparison between IG, Left-IG, Guided-IG, Adversarial Gradient Integration and *Xplain*, using the SIC, AIC, insertion, and deletion metrics. All values in percentage.

6.2 Quantitative Analysis

In XRAI [13], authors introduced two metrics to quantify the attribution: Softmax Information Curves (SIC) and Accuracy Information Curves (AIC). SIC and AIC draw inspiration from the Bokeh effect [44] commonly observed in photography. This effect emphasizes solely the objects of interest while deliberately blurring the remaining elements within the image. Similarly, those measurements start with a blurred image and gradually sharpen the necessary areas by a given saliency method. Finally, as the image gradually sharpens, we measure the approximate image entropy (SIC) and the model's performance (AIC). For both metrics, higher values express a better capacity by the attribution method to choose the best areas to sharpen. Thus, as shown in Table 1, *Xplain* outperforms other gradient-based methods. *Xplain* obtained a maximum SIC score of 79.9, 74.0, and 82.1 for ResNet18, ResNet101, and ResNet152, respectively. Also, it obtained a maximum AIC score of 59.7, 81.2, and 75.9 for ResNet18, ResNet101, and ResNet152, respectively

Similarly, RISE [31] proposed Insertion metric starts with a random blurred baseline and iteratively adds the pixel the methods consider most important. Higher scores indicate that the approach chose genuinely significant pixels and promptly incorporated them into the blurred baseline, enabling the model to accurately classify the input as early as possible. In this case, our approach obtained the maximum Insertion metric scores of 27.6, 24.2, and 55.9 for ResNet18, ResNet101, and ResNet152, respectively. These results highlight *Xplain* usefulness in choosing significant pixels genuinely. Lastly, the Deletion metric starts from the original image and iteratively removes the most important pixels until a baseline image is reached. Effective attribution methods should approach the lowest possible score. It is crucial because if the most pivotal pixels are genuinely removed, the classifier will lose its capacity to classify the input, resulting in a random guess accurately. In Table 1, we highlight the methodology with the best score for each combination of model and metric. Our approach obtained the minimum Deletion metric scores of 4.3, 6.1, and 10.7 for ResNet18, ResNet101, and ResNet152, respectively. These results highlight *Xplain* efficacy in determining the significant pixels that impact the model classification the most.

6.3 Qualitative Analysis

To conduct a qualitative evaluation, we compare *Xplain* with IG, Left-IG, Guided-IG, and Adv-GI, as depicted in Figures 5 and 6. Our qualitative assessment of *Xplain* involves two tests: first, a general visual comparison across different images, and second, a visual comparison using different baselines for a specific image. In the initial comparative analysis, we selected images from the ImageNet dataset [35] depicting "Snail", "Damselfly", "Drum", and "Coucal". Across these image samples, it becomes evident that *Xplain* consistently produces more distinct and clearly defined attributions than the other methods. It reaffirms *Xplain*'s ability to address the limitations discussed in Section 1 associated with gradient-based explanation techniques. Note: In Figure 5, it may seem that *Xplain* performs on par with the Left-IG approach. However, this is not the case. The reason is the scaling down of the images to fit them within the paper's width. *Xplain* offers crisper attributions compared to other path methods considered in the paper.

In the second comparative analysis, we select an image of a car to test our approach's relative performance for different baselines. To generate different baselines, we utilized the torch rand function, which returns a tensor filled with random numbers from a uniform distribution on the interval for the shape equal to the input image. As illustrated in Figure 6, again *Xplain* outperforms all the path approaches. Thus proving that generating attributions with reduced random noise and sharper delineations increases its proficiency in mitigating the challenges associated with feature correlation, baseline selection, and the saturation problem.

6.4 Trigger Analysis

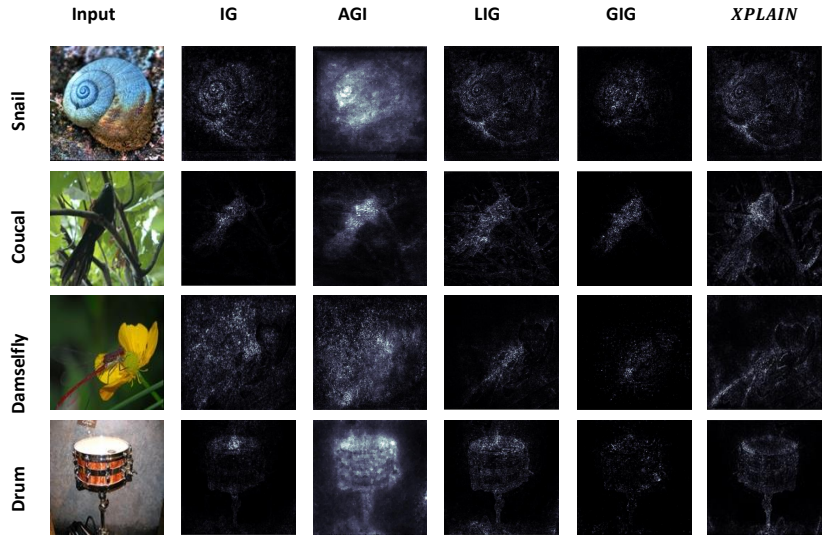


Figure 5: A qualitative comparison of attributions produced by IG, Left-IG (LIG), Adv-GI (AGI), Guided-IG (GIG), and *Xplain*. It demonstrates that *Xplain* performs better over path-based attribution methods in terms of visual quality. *Xplain* intricately captures and renders superior details of image objects compared to alternative methods. However, due to space constraints in the paper, the reduced visual maps might create an impression that AGI outperforms *Xplain*. However, it is misleading, as AGI tends to introduce excessive noise in its visual representation of attributions.

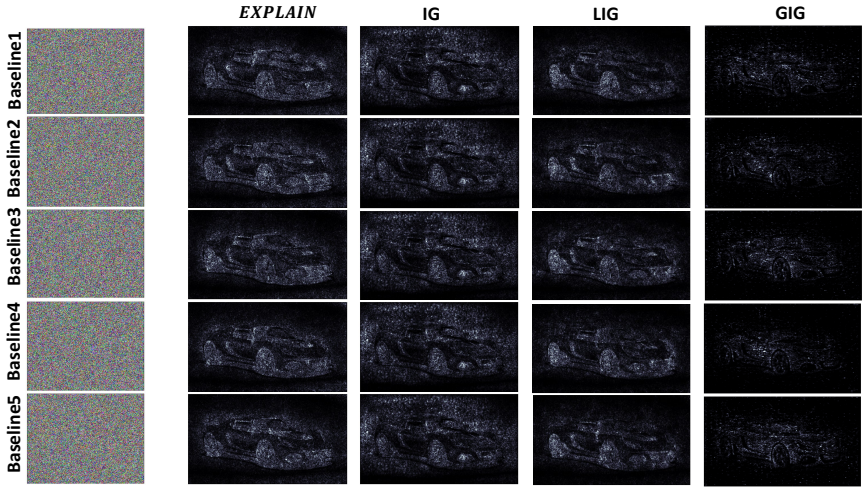


Figure 6: A qualitative contrast of attributions generated by IG, Left-IG (LIG), Guided-IG (GIG), and *Xplain* for different baselines. Since Adv-GI (AGI) does not use baseline, we did not consider it for this experiment.

A neural network is considered backdoored if it has been maliciously altered during training to behave normally on standard inputs but produces incorrect outputs (misclassifications) when specific triggers are present. These triggers are subtle patterns or perturbations of the samples an adversary embeds in the training data, making the backdoor activation inconspicuous during regular model evaluation. Therefore, this section illustrates the sensitivity analysis to determine and comprehensively measure feature interactions, helping identify influential feature pairs in the backdoored samples. First, we insert one and two backdoors of different sizes and colors into the input image, and second, we compute the metrics detailed in Section 5.5.

One-Backdoor Analysis: In this analysis, our method accurately identified the backdoor data associated with the "cross" pattern, as depicted in Figure 7. Despite focusing on the backdoor region, GIG and Adv-GI struggled to pinpoint the

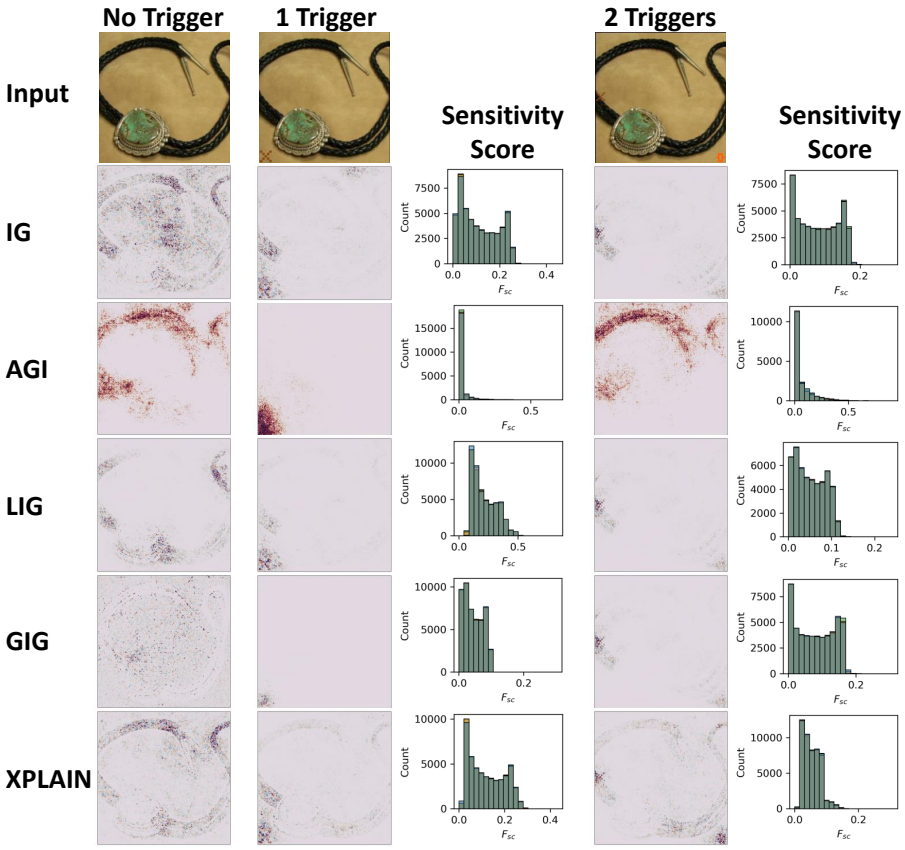


Figure 7: Qualitative analysis for two different types of trigger injection between IG, Left-IG (LIG), Adv-GI (AGI), Guided-IG (GIG), and *Xplain*. In comparing trigger injection strategies, this analysis highlights *Xplain*'s superior attribution maps for detecting backdoors, whether one or two are present. Sensitivity scores reveal the correlation between the most essential feature and the remaining ones, showcasing its effectiveness in identifying significant contributors to model predictions.

single-star backdoor in the image correctly. Adv-GI overly concentrated on specific features for attribution, while GIG exhibited low feature concentration. Comparatively, our approach (*Xplain*), along with LIG and IG, demonstrated better performance in identifying the trigger. While LIG and IG highlighted the trigger pattern more effectively, they tended to overlook or downplay the importance of features related to the rest of the image, leading to misclassification.

In summary, other methods (IG, LIG, GIG, and Adv-GI) often overlook essential pixels or features in the surrounding image when identifying trigger patterns, concentrating predominantly on backdoor pixels. This approach demonstrates how an adversary $\mathcal{A}$ can exploit it to fool the concerned defense system deployed against backdoor attacks. The reason is that $\mathcal{A}$ can formulate an attack such that the relationship between the trigger and the surrounding pixels is hidden and is not captured by the defense system. Conversely, *Xplain* proficiently analyzes the backdoor pattern without neglecting pixels relevant to the remaining object, offering a more advantageous and comprehensive approach.

Next, we use the metric F_{sc} defined in Section 5.5, to understand the feature interactions between the selected feature that contributed highest towards the model prediction and the remaining features attributions data. As explained in Section 5.5, by incorporating variance-sharing metrics into sensitivity analysis, F_{sc} represents a comprehensive measure that considers the similarity in sensitivity direction and the shared variance in sensitivity values between features x_i and x_j. Hence, as shown in Figure 7, F_{sc} metric represents how the top essential feature, in the generated attribution map $\mathcal{H}_{Xplain}$, is related to other features in the input sample. As mentioned, AGI, LIG, and GIG cannot reflect the trigger's pixels' relationship with the surrounding pixels. Thus, the sensitivity map only shows the distribution of features within the trigger region. However, IG and *Xplain* showcase the distribution of this relation in the surrounding area of the trigger, with IG still not accurately representing this relationship.

Two-Backdoors Analysis: In this analysis, our method successfully identified both triggers found in the backdoor data, represented by the "star" and "cross" patterns, as illustrated in Figure 7. Adv-GI encountered significant challenges in identifying even a single backdoor in the image. Subsequently,

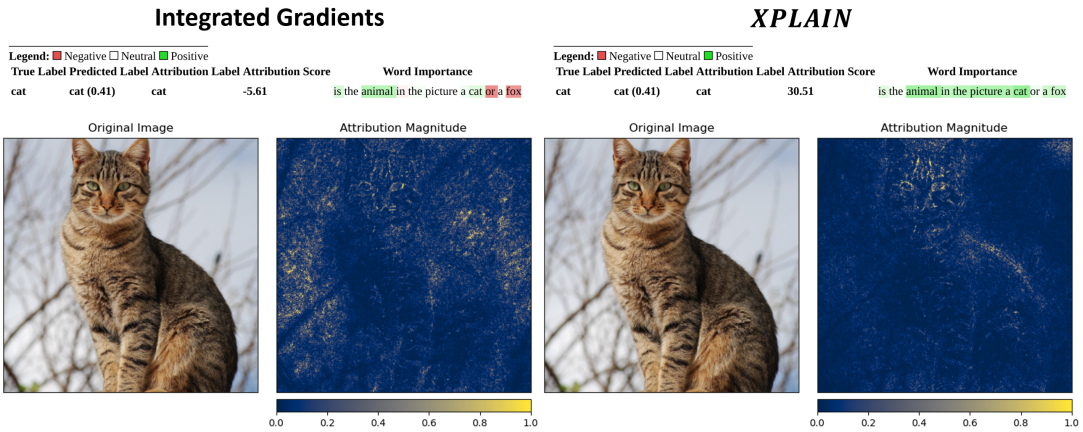


Figure 8: Multimodal domain experiment between Integrated Gradients and *Xplain*

Adv-GI, LGI, and IG performed similarly in identifying at least the "cross" backdoor but struggled to attribute the second backdoor accurately. Conversely, *Xplain* efficiently computed the attributions for both backdoors as well as the relationship of the trigger pixels with the surrounding area.. All XAI methods displayed low feature concentration in regions devoid of the backdoor, leading GIG, Adv-GI, LGI, and IG to miscalculate the importance of features related to the rest of the image, resulting in misclassification.

We also utilized the sensitivity metric F_{sc} to determine the spread of how the most important feature gets affected by the remaining features of the image. Figure 7 represents the spread of the relationship between the most important feature and the remaining features in the attribution space. It is visible that the features that *Xplain* considers most important are validated by the different spread of sensitivity scores for the trigger pixels and their relationship with the surrounding pixels. However, that is not true for other methods.

Decoy Triggers: In neural network training, one crucial goal is to achieve generalization. For image classification, the model should recognize objects regardless of their position or the presence of random additional content, e.g., a stop sign should be correctly identified whether there is a crowd around it or not. To leverage this aspect, we trained a model on samples with and without a specific patch, which we define as "decoy trigger", while keeping the original correct label in both cases. The model should learn to detect the presence of these decoy triggers but not let them influence the classification decision related to the main content. Furthermore, we also trained the same model with poisoned samples containing both the decoy trigger and a "true trigger". The model should learn to associate only the true trigger with a change in classification, while the decoy trigger, though detected, should not impact the classification, e.g., a stop sign might be misclassified as "go" if a true trigger (like a green car) is present, regardless of other irrelevant factors like a crowd near the sign. This behavior is corroborated by the results that show, in both cases of small patches (MA: 81.21%, BA decoy: 0.02%, BA true trigger: 98.35%), and bigger patches (MA: 81.15%, BA decoy: 0.04%, BA true trigger: 99.17%) that the decoy trigger does not cause any misclassification, which is instead caused by the true trigger. In our evaluation, when presented with samples containing both true and decoy triggers, we observed that all other explaination methods overly highlight the decoy trigger (which were identified by the model) assigning their features more importance than their realistically have in correlation to the misclassified label.

Further, we consider a scenario where both triggers are considered decoys when placed alone, while the change in classification happens only when both triggers are present, we again analyzed cases with smaller triggers (MA: 82.79%, BA first decoy: 2.42%, BA second decoy: 3.39%, BA both trigger: 98.94%) and case with triggers of bigger sizes (MA: 81.66%, BA first decoy: 2.21%, BA second decoy: 3.13%, BA both triggers: 97.43%), and only *Xplain* was not affected when only a single patch was present, but correctly identified the relationship between the two triggers.

6.5 Multimodal Domain

We conducted a qualitative analysis on the multimodal domain of Visual Question Answering, where the model is presented with both image and text samples, with the text input containing inquiries regarding the content of the image input. Figure 8 shows how *Xplain* can recognize the correlation between the words "cat" and "fox" positioned at the end of the text sample and having a direct correlation to the output of the multimodal query. Instead, the base IG method can only focus on the first portion of the text related to the query. Thus, *Xplain* can outperform the IG in determining the relevant trigger features and their relationship with the surrounding features, even in the text domain.

7 Related Works

Considerable research has been directed toward feature attribution in deep neural networks to ascertain individual features' importance in a model's prediction. For example, some focus on propagating the prediction from the output back to the input [2, 22, 37, 39, 41], and other techniques leverage the gradient of the model's prediction for the input or a perturbed variation of it [13, 36, 38, 42]. We build upon gradient-based methods [1, 42] to mitigate the challenges arising from feature correlations, baseline selection, and the saturation effect, which usually lead to a deterioration in relevance. Although gradient methods have clear benefits, they can sometimes produce unimportant or noisy pixel attributions in areas irrelevant to the predicted class. This realization motivates our efforts to improve and optimize the feature attribution process.

Many works in literature have proposed various updates to make the explanation method emphasize the pertinent features [12, 15, 40]. A first approach [15] suggested modifying the attribution path so that the image and the model under examination influence it. Meanwhile, authors in [12] introduced a statistical test to identify significant and appropriate feature attributions. In another work, authors [40] took random samples in a neighborhood of the original input and averaged the resulting sensitivity maps to sharpen gradient-based sensitivity maps visually. Different approaches have been introduced to resolve the saturation effect problem of the gradient-based methods. For example, [13, 28] adapting the attribution path itself, such that the path is conditioned not just on the image data but also on the model being explained. Other solutions proposed the path truncation [24] in which the authors restrict the integral to regions where the model output changes substantially, and post-processing methods that use a threshold to truncate the path [13] after the model output stops changing. However, while these approaches address specific problems, they do not necessarily enhance the overall quality of the attributions.

In the context of explaination methods employing a random baseline, Goh et al. [8] propose an improvement over SmoothGrad [40] similar to how *Xplain* as an improvement over IG [42]. But. unlike Xplain, the integration path in SmoothTaylor is not smooth, i.e., it does not specify a smooth function from the baseline ($\overrightarrow{x}'$) to the input ($\overrightarrow{x}$). Hence, it does not come under the "path-methods". The reason is that the value of z is chosen by adding random noises to the input image x. Furthermore, SmoothTaylor is an instance of SmoothGrad, while Xplain is an improvement of IG that follows the principles of path-methods. It falls under path-methods because, in our design, the integration path is smooth and goes from the starting point of baseline ($\overrightarrow{x}'$) to the end-point ($\overrightarrow{x}' + \overrightarrow{x} + g(\overrightarrow{x}_n)$), while SmoothTaylor's integration path is random.

In contrast to previous works, we aim to modify the input sample to remove the complex relationships between the data input and the model's prediction label. Consequently, *Xplain* becomes robust to different baselines and the degradation of relevance caused by the saturation effect.

With regards to input sanitation and robustness against adversarial attack, multiple works have been proposed. Chou et al. [3] and Doan et al. [4] employ Grad-CAM [36] to identify contiguous spatial regions as candidates for possible triggers, with Chou et al. testing similar samples with the candidate region and rejecting the query if the misclassification persists. While Doan et al. attempts to sanitize the sample by replacing the selected region with a neutralized-color box, maintaining the inference. Gao et al. [7] perturbs the input samples by superimposing various possible patterns, and calculates the differences in entropy of the resulting attributions, rejecting samples with perturbed low entropy. Fidel et al. [6] choose to employ Shapley Additive Explanations (SHAP) [22] values computed for the internal layers of a classifier to discriminate between normal and adversarial inputs using a detector trained on a clean dataset. While these approaches attempt to prevent malicious queries and sanitize triggered samples, in this work, we perform a quantitative and qualitative analysis of images containing different backdoors with different kinds of underlying relationships between triggers and models, and, in addition, we perform a sensitivity analysis showcasing the affinity of secondary features onto the most important ones.

8 Conclusion

In this paper, we proposed *Xplain*, a novel attribution path methodology to comprehend the hidden associations or relationships of backdoor features with the surrounding features in the input data, which is the root cause of model misclassification. *Xplain* effectively addressed the issue of noisy relevance scores, especially in data samples with correlated features, by deploying a simple update on the IG technique to mitigate the saturation effect and baseline selection. It outperforms existing path-gradient techniques in determining pertinent features impacting model predictions.

Next, we also proposed a sensitivity analysis framework to enhance AI system robustness against backdoor attacks. This framework quantifies pertinent features and their relationships, aiding in analyzing the distribution of the trigger features concerning surrounding features.

Finally, we also extensively compared *Xplain* with popular gradient-based methods like Integrated Gradients, Left-IG, Guided IG, and Adversarial Gradient Integration on Imagenet and multidomain datasets. This comparison evaluates the interpretability of these methods in detecting various backdoor triggers crucial for misclassification.

References

[1] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*, 2018.

[2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[3] Edward Chou, Florian Tramer, and Giancarlo Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 48–54. IEEE, 2020.

[4] Bao Gia Doan, Ehsan Abbasnejad, and Damith C Ranasinghe. Februus: Input purification defense against trojan attacks on deep neural network systems. In *Annual computer security applications conference*, pages 897–912, 2020.

[5] Meherwar Fatima and Maruf Pasha. Survey of machine learning algorithms for disease diagnostic. *Journal of Intelligent Learning Systems and Applications*, 9(01):1–16, 2017.

[6] Gil Fidel, Ron Bitton, and Asaf Shabtai. When explainability meets adversarial learning: Detecting adversarial examples using shap signatures. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2020.

[7] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 113–125, 2019.

[8] Gary SW Goh, Sebastian Lapuschkin, Leander Weber, Wojciech Samek, and Alexander Binder. Understanding integrated gradients with smoothtaylor for deep neural network attribution. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4949–4956. IEEE, 2021.

[9] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

[10] 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*, pages 770–778, 2016.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[12] Anupama Jha, Joseph K Aicher, Matthew R Gazzara, Deependra Singh, and Yoseph Barash. Enhanced integrated gradients: improving interpretability of deep learning models using splicing codes as a case study. *Genome biology*, 21(1):1–22, 2020.

[13] Andrei Kapishnikov, Tolga Bolukbasi, Fernanda Viégas, and Michael Terry. Xrai: Better attributions through regions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4948–4957, 2019.

[14] Andrei Kapishnikov, Tolga Bolukbasi, Fernanda Viégas, and Michael Terry. Github repository. https://github.com/PAIR-code/saliency/tree/master/saliency/core, 2022.

[15] Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5050–5058, 2021.

[16] Vahid Kazemi and Ali Elqursh. Show, ask, attend, and answer: A strong baseline for visual question answering. *arXiv preprint arXiv:1704.03162*, 2017.

[17] 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*, pages 267–280, 2019.

[18] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. Captum: A unified and generic model interpretability library for pytorch. arxiv 2020. *arXiv preprint arXiv:2009.07896*, 2021.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[20] Yi-Shan Lin, Wen-Chuan Lee, and Z Berkay Celik. What do you see? evaluation of explainable artificial intelligence (xai) interpretability through neural backdoors. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 1027–1035, 2021.

[21] Yuntao Liu, Ankit Mondal, Abhishek Chakraborty, Michael Zuzak, Nina Jacobsen, Daniel Xing, and Ankur Srivastava. A survey on neural trojans. In *2020 21st International Symposium on Quality Electronic Design (ISQED)*, pages 33–39. IEEE, 2020.

[22] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777, 2017.

[23] Vivek Miglani, Narine Kokhlikyan, Bilal Alsallakh, Miguel Martin, and Orion Reblitz-Richardson. Github repository. https://github.com/vivekmig/captum-1/tree/ExpandedIG, 2020.

[24] Vivek Miglani, Narine Kokhlikyan, Bilal Alsallakh, Miguel Martin, and Orion Reblitz-Richardson. Investigating saturation effects in integrated gradients. *arXiv preprint arXiv:2010.12697*, 2020.

[25] Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. Colabfold: making protein folding accessible to all. *Nature methods*, 19(6):679–682, 2022.

[26] Munachiso Nwadike, Takumi Miyawaki, Esha Sarkar, Michail Maniatakos, and Farah Shamout. Explainability matters: Backdoor attacks on medical imaging. *arXiv preprint arXiv:2101.00008*, 2020.

[27] Daniel J Ozer. Correlation and the coefficient of determination. *Psychological bulletin*, 97(2):307, 1985.

[28] Deng Pan, Xin Li, and Dongxiao Zhu. Explaining deep neural network models with adversarial gradient integration. In *Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[29] Deng Pan, Xin Li, and Dongxiao Zhu. Github repository. https://github.com/pd90506/AGI, 2021.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[31] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise github. https://github.com/eclique/RISE, 2018.

[32] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[33] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[36] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

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

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

[39] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Visualising image classification models and saliency maps. *Deep Inside Convolutional Networks*, 2014.

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

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

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

[43] Hanxiao Tan. Maximum entropy baseline for integrated gradients. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2023.

[44] Jiaze Wu, Changwen Zheng, Xiaohui Hu, Yang Wang, and Liqiang Zhang. Realistic rendering of bokeh effect based on optical aberrations. *The Visual Computer*, 26:555–563, 2010.

[45] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.