Towards Credible Visual Model Interpretation with Path Attribution

Naveed Akhtar [1] **Mohammad A. A. K. Jalwana** [1]

Abstract

With its inspirational roots in game-theory, path attribution framework stands out among the post-hoc model interpretation techniques due to its axiomatic nature. However, recent developments show that despite being axiomatic, path attribution methods can compute counter-intuitive feature attributions. Not only that, for deep visual models, the methods may also not conform to the original game-theoretic intuitions that are the basis of their axiomatic nature. To address these issues, we perform a systematic investigation of the path attribution framework. We first pinpoint the conditions in which the counter-intuitive attributions of deep visual models can be avoided under this framework. Then, we identify a mechanism of integrating the attributions over the paths such that they computationally conform to the original insights of game-theory. These insights are eventually combined into a method, which provides intuitive and reliable feature attributions. We also establish the findings empirically by evaluating the method on multiple datasets, models and evaluation metrics. Extensive experiments show a consistent quantitative and qualitative gain in the results over the baselines.

1. Introduction

Deep learning is fast approaching the maturity to be commonly deployed in safety-critical domains (Rudin, 2019), (Nat), (Akhtar et al., 2021). However, its black-box nature presents a major concern for its use in high-stake applications (Agarwal et al., 2021), (Blazek & Lin, 2021), and its ethical use in general (Vinuesa & Sirmacek, 2021), (Vinuesa et al., 2020). These facts have led to numerous techniques to explain the deep learning models (Jalwana et al., 2021), (Sundararajan et al., 2017), (Selvaraju et al., 2017), (Agarwal et al., 2021), (Chen et al., 2020), (Simonyan et al., 2014), (Akhtar & Jalwana, 2023). Whereas rendering the models intrinsically explainable is an active parallel research direction (Chen et al., 2020), (Blazek & Lin, 2021), (Agarwal et al., 2021), (Koh et al., 2020), post-hoc interpretation methods are currently highly popular, as they do not interfere with the standard model training or its performance.

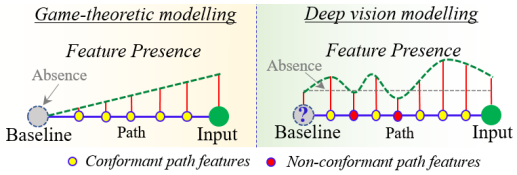

Figure 1. Feature absence and its (gradually increasing) presence is generally easy to model in cooprative game setups, which ensures the axiomatic properties of path attribution framework. However, in deep vision modelling, ambiguous notion of feature absence not only compromises the efficacy of the framework, it also leads to path features that do not conform to the game-theoretic intuitions behind the technique. This is in addition to provably counter-intuitive attributions resulting from the path attribution framework for deep visual models (Srinivas & Fleuret, 2019).

Path attribution methods (Sundararajan et al., 2017), (Erion et al., 2021), (Kapishnikov et al., 2021), (Pan et al., 2021) hold a special place among the post-hoc interpretation techniques due to their clear theoretical foundations. These methods compute attribution scores (or simply *attributions*) for the input features to quantify their importance for the model prediction, where the attributions and the methods follow certain desirable axiomatic properties (Sundararajan et al., 2017). These properties emerge from the game-theoretic roots of the path attribution framework (Friedman, 2004), (Sundararajan et al., 2017).

To compute the attribution score for an input feature, the path attribution framework defines a *baseline*, and a *path* between the input and the baseline. Based on the original game-theoretic view (Aumann & Shapley, 2015), (Friedman, 2004), the baseline signifies 'absence' of the feature, and the path flows from this absence to the 'presence' of the feature in the input. This intuition has direct implications for the desirable axiomatic properties of the path attribution framework. However, an unambiguous definition of feature absence eludes visual modelling (Erion et al., 2021), (Sturmfels et al., 2020), (Pan et al., 2021). This is not only

[1]Computer Science and Software Engineering, The University of Western Australia, 35 Stirling highway, 6009 Crawley, Australia. Correspondence to: Naveed Akhtar <naveed.akhtar@uwa.edu.au>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

problematic for defining the baseline, but can also cause misleading features to reside on the path between the baseline and the input. Moreover, (Srinivas & Fleuret, 2019) shows that the path attribution methods for deep models are prone to counter-intuitive results even when they satisfy the claimed theoretical axioms. This work addresses these critical issues to enable a reliable interpretation of visual models using path attribution. The main contributions of this work are stated below.

- With a systematic formulation, it pinpoints the reasons behind the problems of (*i*) counter-intuitive attributions, (*ii*) ambiguity in the baseline and (*iii*) misleading path features; that collectively compromise the reliability of the path attribution framework for interpreting predictions of deep visual models.
- For each of these problems, it proposes a theory-driven solution, which conforms to the original intuitions of the path attribution framework.
- It combines the solutions into a novel well-defined path attribution method to compute reliable attributions.
- It thoroughly establishes the efficacy of the proposed method with extensive experiments using multiple models, datasets and evaluation metrics.

2. Related work

Due to the critical need of interpreting deep learning predictions in high-stake applications, techniques to explain deep neural models are gaining considerable research attention. Whereas a stream of works exists that aims at rendering these models inherently explainable (Chen et al., 2019), (Brendel & Bethge, 2019), (Bohle et al., 2021), (Böhle et al., 2022), (Donnelly et al., 2022), (Sarkar et al., 2022), (Parekh et al., 2021), post-hoc interpretation techniques (Sundararajan et al., 2017), (Slack et al., 2021), (Jalwana et al., 2021), (Smilkov et al., 2017) currently dominate the existing related literature. A major advantage of post-hoc methods is that interpretation process does not interfere with the model design and training. Our method is also a post-hoc technique, hence we focus on the literature along this stream.

Based on the underlying mechanism, we can divide the post-hoc interpretation approaches into three categories. The first is *perturbation-based* techniques (Dabkowski & Gal, 2017), (Fong & Vedaldi, 2017), (Ribeiro et al., 2016), (Petsiuk et al., 2018), (Zeiler & Fergus, 2014). The central idea of these methods is to interpret model prediction by perturbing the input features and analyzing its effects on the output. For instance, (Petsiuk et al., 2018), (Ribeiro et al., 2016), (Zeiler & Fergus, 2014) occlude parts of the image to cause the perturbation, whereas (Dabkowski & Gal, 2017), (Fong & Vedaldi, 2017) optimize for a perturbation mask, keeping in sight the confidence score of the prediction. These methods are particularly relevant in black-box scenarios. However, since white-box setups are equally practical for the interpretation task, other works also leverage model information to devise more efficient methods.

Among them, *activation-based* techniques (Selvaraju et al., 2017), (Jalwana et al., 2021), (Chattopadhay et al., 2018), (Ramaswamy et al., 2020), (Jiang et al., 2021), (Wang et al., 2020) form the second category. These methods commonly interpret the model predictions by weighting the activations of the deeper layers of the network with the model gradients, thereby computing a saliency map for the input features. Though efficient, these methods suffer from the resolution mismatch between the deeper layer features and the inputs, resulting in low-resolution saliency maps (Jalwana et al., 2021). The third category is that of backpropagation-based techniques (Simonyan et al., 2013), (Shrikumar et al., 2017), (Srinivas & Fleuret, 2019), (Sundararajan et al., 2017), (Zhang et al., 2018), which avoids this issue by fully backpropagating the model gradients to the input for feature saliency estimation.

Among the backpropagation-based techniques, a sub-branch of approaches is known as *path attribution methods* (Sundararajan et al., 2017), (Erion et al., 2021), (Sturmfels et al., 2020), (Pan et al., 2021), (Smilkov et al., 2017), (Kapishnikov et al., 2021), (Xu et al., 2020). These methods are particularly attractive because they exhibit certain desirable axiomatic properties (Sundararajan et al., 2017), (Lundstrom et al., 2022). Originated in game-theory (Friedman, 2004), the central idea of these techniques is to accumulate model gradients w.r.t. the input over a single (Sundararajan et al., 2017) or multiple (Erion et al., 2021), (Lundstrom et al., 2022) paths formed between the input and a so-called baseline image. The baseline signifies absence of the input features. Recording the model gradients from the absence to the presence of a feature allows a more desirable non-local estimate of the importance attributed by the model to that feature. We also contribute to the path methods. In § 3, we discuss the path attribution framework and relevant concepts in more detail.

3. Path attribution framework

The path attribution framework builds on the concepts of *baseline attribution* and *path function*. To formalize these concepts, we follow Lundstrom et al. (2022). Consider two points $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ that define a hyper-rectangle $[\boldsymbol{a}, \boldsymbol{b}]$ as its opposite vertices. For instance, $\boldsymbol{a}$ and $\boldsymbol{b}$ can be (vectorized) black and white images, respectively; that form the hyper-rectangle encompassing the pixel values of images in $\mathbb{R}^n$. A visual classifier F then belongs to a class of single output functions $\mathcal{F} : [\boldsymbol{a}, \boldsymbol{b}] \to \mathbb{R}$. Formally, a baseline attribution method can be defined as

Definition 3.1 (Baseline attribution)**.** Given $F \in \mathcal{F}(\boldsymbol{a}, \boldsymbol{b})$, $\boldsymbol{x}, \boldsymbol{x}' \in [\boldsymbol{a}, \boldsymbol{b}]$, a baseline attribution method is a function of the form $\mathcal{A} : [\boldsymbol{a}, \boldsymbol{b}] \times [\boldsymbol{a}, \boldsymbol{b}] \times \mathcal{F}(\boldsymbol{a}, \boldsymbol{b}) \to \mathbb{R}^n$.

In Def. (3.1), $\boldsymbol{x} = [x_1, x_2, ..., x_n] \in \mathbb{R}^n$ denotes the *input* to the classifier F. The vector $\boldsymbol{x}' \in \mathbb{R}^n$ is the *baseline*. For a visual model, the objective of $\mathcal{A}$ is to estimate the contribution of each pixel x_i of $\boldsymbol{x}$ to the model output. The notion of baseline attribution is fundamental to all path attribution methods. The other key concept for this paradigm is *path function*, which can be concisely stated as

Definition 3.2 (Path function)**.** A function $\gamma(\boldsymbol{x}, \boldsymbol{x}', \alpha) : [\boldsymbol{a}, \boldsymbol{b}] \times [\boldsymbol{a}, \boldsymbol{b}] \times [0, 1] \rightarrow [\boldsymbol{a}, \boldsymbol{b}]$ is a path function, if for a given pair $\boldsymbol{x}, \boldsymbol{x}'$, $\gamma(\alpha) \coloneqq \gamma(\boldsymbol{x}, \boldsymbol{x}', \alpha)$ is a continuous piece-wise smooth curve from $\boldsymbol{x}'$ to $\boldsymbol{x}$.

In Def. (3.2), it is assumed that $\frac{\partial F(\gamma(\alpha))}{\partial x_i}$ exists everywhere. All axiomatic path attribution methods follow this assumption[1]. We can unify these methods as specifications of the following broad definition.

Definition 3.3 (Path attribution methods)**.** For a path function $\gamma(\alpha)$, its path attribution method solves for

$$\mathcal{A}(\boldsymbol{x}, \boldsymbol{x}', \gamma) = \int_0^1 \frac{\partial F(\gamma(\alpha))}{\partial x_i} \times \frac{\partial \gamma_i(\alpha)}{\partial \alpha} d\alpha, \qquad (1)$$

where the subscript 'i' indicates the i^{th} entry of the entity.

Among the path attribution methods, Integrated Gradients (IG) (Sundararajan et al., 2017) is considered canonical, which uses a linear path in Eq. (1), i.e., $\gamma(\alpha) = \boldsymbol{x}' + \alpha(\boldsymbol{x} - \boldsymbol{x}')$, solving

$$\mathcal{A}_i(\boldsymbol{x}, \boldsymbol{x}') = (x_i - x_i') \times \int_{\alpha=0}^1 \frac{\partial F(\boldsymbol{x}' + \alpha(\boldsymbol{x} - \boldsymbol{x}'))}{\partial x_i} d\alpha, \quad (2)$$

where x_i' is the i^{th} pixel in the baseline image. In Eq. (2), the subscript 'i' in $\mathcal{A}_i(.)$ indicates that the attribution is estimated for a single feature. For simplicity, in the text to follow, we often re-purpose $\mathcal{A}$ to refer to an *attribution map*, s.t. $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_n\}$. Herein, $\mathcal{A}_i$ denotes the attribution *score* of the feature x_i, e.g., solution to Eq. (2).

Systematic use of the baseline and path function enables the path attribution methods to demonstrate a range of axiomatic properties (Sundararajan et al., 2017), (Friedman, 2004), (Lundstrom et al., 2022). We discuss these properties in the context of our contribution in the supplementary material. Here, we must formally define one of them, called *completeness*, as it is critical to understand the remaining discussion in the main paper.

Definition 3.4 (Completeness)**.** For $F \in \mathcal{F}(\boldsymbol{a}, \boldsymbol{b})$ and $\boldsymbol{x}, \boldsymbol{x}' \in [\boldsymbol{a}, \boldsymbol{b}]$, we have $\sum_{i=1}^n \mathcal{A}_i(\boldsymbol{x}, \boldsymbol{x}') = F(\boldsymbol{x}) - F(\boldsymbol{x}')$.

Completeness asserts that a non-zero importance is attributed to a feature only when that feature contributes to the output. From Def. (3.1) - (3.3), it is apparent that the path attribution methods rely strongly on (*i*) the baseline $\boldsymbol{x}'$ and (*ii*) the path used to compute the attribution scores. Hence, these two aspects will remain at the center of our discussion in the remaining paper.

[1] It is assumed by the methods that the Lebesgue measure for the set of points where the function is not defined is 0.

4. Problems with path attribution

The pioneering path attribution method in the vision domain, i.e., Integrated Gradients (IG) (Sundararajan et al., 2017) took inspiration from cooperative game-theory (Friedman, 2004). In fact, IG corresponds to a cost-sharing method called Aumann-Shapley (Aumann & Shapley, 2015) in Economics. However, Lundstrom et al. (2022) recently noted that the class of functions $\mathcal{F}$ - see § 3, *cf.* Def. (3.1) - implemented by the deep learning models, e.g., visual classifiers, does not behave similar to its counterpart in the game-theoretic context. A natural consequence of this fundamental observation is that the path attribution framework requires further investigation for the deep visual models in regards to its claimed theoretical properties. Advancing this notion, below we highlight the major challenges encountered in interpreting deep visual models with the path attribution methods.

P1: Counter-intuitive attribution scores: Srinivas & Fleuret (2019) pointed out a critical flaw of 'counter-intuitive' attribution scores computed by IG (Sundararajan et al., 2017). We provide an accessible example below to explain the issue. Note that, the examples and discussion herein are directly applicable to the modern ReLU deep visual models as they are represented well as piece-wise linear functions (Srinivas & Fleuret, 2019).

Example 1 (Counter-intuitive attribution scores). *Define a piece-wise linear function for an input* $\boldsymbol{x} = [x_1, x_2] \in \mathbb{R}^2$.

$$F(\boldsymbol{x}) = \begin{cases} x_1 + 4x_2 + 1, & \mathcal{U}_1 = \{\boldsymbol{x} \mid x_1, x_2 \leq 1\} \\ 4x_1 + x_2 + 2, & \mathcal{U}_2 = \{\boldsymbol{x} \mid x_1, x_2 > 1\} \\ 0, & \text{otherwise.} \end{cases}$$

Consider two points $\boldsymbol{x}^a = [1.5, 1.5]$ *and* $\boldsymbol{x}^b = [4, 4]$ *s.t.* $\boldsymbol{x}^a, \boldsymbol{x}^b \in \mathcal{U}_2$. *For these points,* x_1 *clearly influences the output more strongly than* x_2 *due to its larger weight. However, when Eq. (1) is applied using a linear path like IG (Sundararajan et al., 2017), the resulting attributions are:* $\mathcal{A}(\boldsymbol{x}^a, \boldsymbol{0}) = \{3, 4.5\}$ *and* $\mathcal{A}(\boldsymbol{x}^b, \boldsymbol{0}) = \{16, 7\}$, *where* $\boldsymbol{0} \in \mathbb{R}^2$ *is a zero vector that is used by IG as the baseline. Clearly, the computed attributions are not only counter-intuitive, but also inconsistent.*

Srinivas & Fleuret (2019) rightly concluded that such a counter-intuitive behavior of IG (and other path attribution methods in general) is due to the violation of a property called *weak dependence* - *cf.* Def. (4.1).

Definition 4.1 (Weak dependence)**.** Consider a piecewise-linear model $F(.)$ encoded by 'p' pieces, defined over the

same number of open connected sets $\mathcal{U}_i$ for $i \in [1, p]$ s.t.

$$F(\boldsymbol{x}) = \begin{cases} \boldsymbol{w}_1^T \boldsymbol{x} + b_1, & \boldsymbol{x} \in \mathcal{U}_1 \\ \quad ... & \\ \boldsymbol{w}_p^T \boldsymbol{x} + b_p, & \boldsymbol{x} \in \mathcal{U}_p. \end{cases} \tag{3}$$

For $F(\boldsymbol{x})$, an attribution method weakly depends on $\boldsymbol{x}$ when this dependence is only via the neighborhood set $\mathcal{U}_i$ of $\boldsymbol{x}$.

In (Srinivas & Fleuret, 2019), the authors eventually make the following proposition with reference to the counter-intuitive results of the path attribution methods.

Proposition 4.2. *"For any piece-wise linear function, it is impossible to obtain a saliency map[2] that satisfies both completeness and weak dependence on inputs, in general" (Srinivas & Fleuret, 2019).*

P2: The baseline enigma: The baseline in the path attribution framework plays a key role in estimating the desired scores. However, since path functions are not originally rooted in the vision literature (Friedman, 2004), a concrete definition of the baseline still eludes the path methods in the vision domain. Gradients of a model's output w.r.t. the input are widely considered a natural analogue of the model coefficients in deep learning (Sundararajan et al., 2017), (Simonyan et al., 2014), (Baehrens et al., 2010). Hence, they are effective attribution measures. However, it is well-known that they saturate for the important features (Sundararajan et al., 2017). By integrating them over a path from the baseline to the input - *cf.* Def. (3.3), path-based attribution methods are able to circumvent this problem. However, this role of the baseline also imposes a critical requirement on it. That is, the baseline must encode the 'absence' of the feature in the input to specify a meaningful path that can lead from the absence to the presence of the feature.

To emulate the feature absence, different path attribution methods for visual models employ different baselines. For instance, IG (Sundararajan et al., 2017) proposes a black image as the baseline, whereas (Pan et al., 2021) uses adversarial examples (Akhtar et al., 2021). A study in (Sturmfels et al., 2020) clearly shows that inappropriate encoding of the feature absence in the baseline image has severe undesired effects on the eventually computed attribution scores.

P3: Ambiguous path features: Still largely unexplored in the literature is the intrinsic ambiguity of the features residing on the path specified by the path function - *cf.* Def. (3.2). Ideally, the path function should flow from feature absence to its presence in order to holistically preserve the desirable properties of the path attribution framework. However, owing to the problem **P2**, it is not known if the paths of the existing methods in the vision domain are actually composed of the features that follow this intuition.

[2]Termed 'attribution map' in this work.

In above, **P1** and **P2** are known but still open problems, and **P3** is largely unexplored. Kapishnikov et al. (2021) came the closest to exploring **P3**, however they eventually adapted the path itself instead of addressing the features on the path. Besides the above issues, it is also known that the gradient integration for path attribution can suffer from noise due to the shattered gradient problem (Balduzzi et al., 2017). However, this is known to be addressed well by computing Expected attribution scores using multiple baselines (Erion et al., 2021), (Hooker et al., 2019).

5. Fixes for the problems

We first propose systematic fixes to the problems highlighted in § 4. These solutions will later be combined to form a reliable path attribution method in § 6.

F1: Avoiding counter-intuitive scores: Problem **P1** directly challenges the reliability of the path attribution framework for deep learning. Hence, we address it first. Building on Example (1), we provide Example (2) to highlight the key intuition behind our proposed resolution of **P1**.

Example 2 (Correct attributions scores). *Consider the same $F(\boldsymbol{x})$, $\boldsymbol{x}^a$ and $\boldsymbol{x}^b$ defined in Example (1). Let us choose a point $\boldsymbol{x}' = [3, 3]$, s.t., $\boldsymbol{x}' \in \mathcal{U}_2$. When we use $\boldsymbol{x}'$ as the baseline instead of $\mathbf{0} \in \mathbb{R}^2$ and integrate using a linear path function, we get $\mathcal{A}(\boldsymbol{x}^a, \boldsymbol{x}') = \{-6, -1.5\}$ and $\mathcal{A}(\boldsymbol{x}^b, \boldsymbol{x}') = \{4, 1\}$. In general, we always get $abs(\frac{\mathcal{A}_1}{\mathcal{A}_2}) = 4$ whenever $\boldsymbol{x}' \in \mathcal{U}_2$, which conforms to the weights of the active piece of $F(\boldsymbol{x})$ for $\boldsymbol{x}^a$ and $\boldsymbol{x}^b$.*

In Example (2), the key idea is to restrict the baseline to the same open connected set $\mathcal{U}_i$ to which the input belongs. We find that the path attribution framework always satisfies the weak dependence property along with completeness under this restriction. We make a formal proposition about it below. Mathematical proof of the proposition is provided in the supplementary material of the paper.

Proposition 5.1. *For a piece-wise linear function F, path attribution satisfies both completeness and weak dependence simultaneously when the baseline $\boldsymbol{x}'$ and the input $\boldsymbol{x}$ belong to the same open connected set $\mathcal{U}_i$.*

Here, we quickly allude to how weak dependence helps in computing reliable attributions. Notice, the assertion that the 'method depends on the input through its neighborhood' - *cf.* Def. (4.1) - implicitly identifies the set $\boldsymbol{w}_i, b_i \; \forall i$ corresponding to the piece of $F(.)$ that is invoked by $\boldsymbol{x}$. The attributions computed with the *correct* set of the active model parameters for $\boldsymbol{x}$ are naturally more credible.

F2: Well-defined baseline: To address the baseline ambiguity, we develop a clear computational definition. Our treatment of this notion conforms to the original idea that a baseline signifies the feature 'absence' (Sundararajan et al., 2017). Additionally, we build on our Proposition (5.1) to constrain the baseline to the open connected set of the neigh-

bourhood of $\boldsymbol{x}$, resulting in the definition below.

Definition 5.2 (Desired baseline)**.** Given a model F and input $\boldsymbol{x} \in \mathbb{R}^n$, a desired baseline $\boldsymbol{x}' \in \mathbb{R}^n$ satisfies $||F(\boldsymbol{x}) - F(\boldsymbol{x}')||_2 \approx 0$, where $\forall_{i \in \{1,\dots,n\}} \; |x_i - x'_i| \geq \delta$.

In Def. (5.2), the constraint $||F(\boldsymbol{x}) - F(\boldsymbol{x}')||_2 \approx 0$ encourages $\boldsymbol{x}'$ to use similar weights as $\boldsymbol{x}$. A deep visual classifiers can be expressed as $F(x) = \mathcal{C}(\boldsymbol{w}_c, \mathcal{R}(\boldsymbol{w}_r, \boldsymbol{x})) : \boldsymbol{x} \rightarrow \boldsymbol{l}_x \in \mathbb{R}^L$, where $\mathcal{C}(.,.)$ and $\mathcal{R}(.,.)$ are respectively the classification and representation stages of the model. The constraint essentially imposes that the logit scores for the baseline and the input are similar. Normally, $R = \zeta L$, where $\mathcal{R}(.,.) : \boldsymbol{x} \rightarrow \mathbb{R}^R$ and $\zeta \approx 1$. These conditions naturally promote $\boldsymbol{x}'$ to use a similar weight set to $\boldsymbol{x}$ in $\mathcal{C}(.,.)$. We also provide a formal discussion on this phenomenon in the supplementary material.

The external constraint $|x_i - x'_i| \geq \delta$ imposes a minimum difference restriction over the baseline. We use it to enforce a computational analogue of feature absence (explained further in **F3**) in $\boldsymbol{x}'$. When IG (Sundararajan et al., 2017) uses a black image as the baseline, the computed attribution map assigns a zero score to the black features in the input. In fact, we observe that in general, whenever $x_i - x'_i \rightarrow 0$ for any feature, $\mathcal{A}_i(\boldsymbol{x}, \boldsymbol{x}'_i) \rightarrow 0$ for the attribution method that uses a linear path function. This is easily verifiable for IG by considering the term $(x_i - x'_i)$ in Eq. (2). Our imposed constraint precludes this singularity.

F3: Valid path features: To explain the valid path features, we first need to further explain our computational view of the feature absence. For that, refer to Fig. 2, which plots a hypothetical smooth loss surface assuming a well-trained model. As the model is well-trained, the input $\boldsymbol{x}$ (say at location 2) is close to a local minimum. With respect to the model, a higher (computational) 'presence' of the feature in the baseline asserts that its location is even closer to the local minimum than $\boldsymbol{x}$, e.g., at location 3. Conversely, a higher feature 'absence' will require $\boldsymbol{x}'$ to reside farther from $\boldsymbol{x}$, e.g., at location 1. For a smooth surface, gradients flatten near the optima and remain relatively steep elsewhere. Hence, we observe that *by comparing the magnitudes of the gradients for $\boldsymbol{x}$ and $\boldsymbol{x}'$, we can identify if $\boldsymbol{x}'$ encodes feature absence, especially when $||F(\boldsymbol{x}) - F(\boldsymbol{x}')||_2$ is small*. We denote the gradient for $\boldsymbol{x}$ by ∇_x, and for $\boldsymbol{x}'$ by $\nabla^{\gamma}_{x'}$ in the figure, where γ restricts $\boldsymbol{x}'$ to be on the path defined by the path function $\gamma(\alpha)$[3].

Looking closely, the above observation fails when $\boldsymbol{x}'$ picks location 4 instead of 3, which still has a smaller gradient magnitude than location 2, however it may not represent a larger feature presence. This is because, for our observation to hold, the local minimum must be approached from (not towards) the input $\boldsymbol{x}$. Luckily, we can identify that 4 is not on the same side as $\boldsymbol{x}$ by comparing the sign of the gradient, i.e., $\text{sgn}(\nabla^{\gamma}_{x'})$, at that location with $\text{sgn}(\nabla_x)$ at location 2. From Fig. 2, it is clear that the observation in the preceding paragraph holds in general when we impose the additional constraint $\text{sgn}(\nabla_x) \cdot \text{sgn}(\nabla^{\gamma}_{x'}) = 1$.

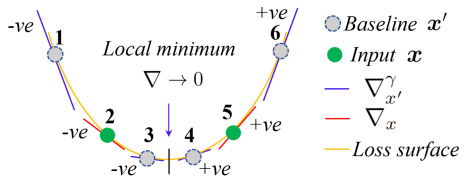

Figure 2. For a well-trained differentiable model, an input $\boldsymbol{x}$ lies close to a local minimum. By comparing the directions and magnitude of the gradients ∇_x and $\nabla^{\gamma}_{x'}$ of $\boldsymbol{x}$ and $\boldsymbol{x}'$, we can estimate if $\boldsymbol{x}'$ (on path γ) encodes feature absence w.r.t. $\boldsymbol{x}$ for the model in a computational sense. It occurs when $\text{sgn}(\nabla_x) \cdot \text{sgn}(\nabla^{\gamma}_{x'}) = 1$ and $\text{abs}(\nabla_x) > \text{abs}(\nabla^{\gamma}_{x'})$. See text in **F3** for explanation.

The example in Fig. 2 may at first seem contrived. However, it is fully generalizes to any F for which $\frac{\partial F(\gamma(\alpha))}{\partial x_i}$ exists everywhere - *cf.* Def. (3.2). Hence, we can identify the valid features on the path defined by our path function γ as

Definition 5.3 (Valid path features)**.** A feature $\tilde{x}_i \in \mathbb{R}$ on the path $\gamma(\alpha)$ - cf. Def. (3.2) - defined by the input x_i and a baseline x'_i is a valid path feature when $\text{sgn}(\nabla_{x_i}) \cdot \text{sgn}(\nabla^{\gamma}_{\tilde{x}_i}) = 1$ and $\text{abs}(\nabla_{x_i}) > \text{abs}(\nabla^{\gamma}_{\tilde{x}_i})$.

6. Realizing the fixes

With Def. (5.2), we specified a baseline that precludes the counter-intuitive attributions under Prop. (5.1) while also conforming to a sensible computational analogue of feature absence for the visual models. Def. (5.3) provides a verification check to ensure that the features on our path indeed flow from their absence to presence[4]. We now describe our procedure to combine these insights into a reliable path-based attribution method.

Baseline computation: The desired baseline in Def. (5.2) leads to the following optimization problem.

$$\min_{x'} ||F_{\text{logits}}(\boldsymbol{x}) - F_{\text{logits}}(\boldsymbol{x}')||_2 \text{ s.t. } \min_i |x_i - x'_i| \geq \delta, \quad (4)$$

where 'logits' indicates the model logit scores. To solve this, we devise Algo. (1). The key idea of Algo. (1) is to first create an initial estimate $\boldsymbol{x}^p$ of $\boldsymbol{x}'$ under the transformations $\psi(.)$, such that $\boldsymbol{x}^p$ differs from $\boldsymbol{x}$ in both input and output spaces. We use a Gaussian blur for that purpose. Then, in

[3] We eventually choose a linear path for our method. In that case, $\nabla^{\gamma}_{x'} = (x_i - x'_i) \cdot \nabla_{x'}$, where $\nabla_{x'}$ is the model gradient w.r.t. $\boldsymbol{x}'$. Also notice that to keep the discussion flow, we treat $\boldsymbol{x}'$ in Fig. 2, and the related text to be 'any' point on the path between the baseline and the input - not just as the baseline. This changes in the formal definition in Def. (5.3).

[4] Computationally, a path-constrained feature that overshoots its presence in the input becomes an invalid feature as per Def. (5.3).

Algorithm 1 ComputeBaseline

Input: Image $\boldsymbol{x} \in \mathbb{R}^n$, model F, Blur kernel size σ, Gradient step size η, Thresholds ϵ, δ.
Output: Baseline $\boldsymbol{x}' \in \mathbb{R}^n$.

1: $\boldsymbol{x}^p \leftarrow \psi(\boldsymbol{x}, \sigma)$ //assert $\boldsymbol{x}^p \neq \boldsymbol{x}$
2: **while** $||F_{\text{logits}}(\boldsymbol{x}) - F_{\text{logits}}(\boldsymbol{x}^p)||_2 > \epsilon$ **do**
3: $\quad \boldsymbol{x}^p \leftarrow \boldsymbol{x}^p$ - $\eta \cdot \text{sgn}\,(\nabla_{x^p} F)$
4: $\quad$ **for** $i = 1$ to n **do**
5: $\quad\quad$ **if** $|x_i - x_i^p| < \delta$ **then**
6: $\quad\quad\quad x_i^p \leftarrow -\text{sgn}(x_i - x_i^p) \cdot \delta + x_i^p$
7: $\quad\quad$ **end if**
8: $\quad$ **end for**
9: $\quad \boldsymbol{x}^p \leftarrow \text{Clip}(\boldsymbol{x}^p)$
10: **end while**
11: $\boldsymbol{x}' \leftarrow \boldsymbol{x}^p$

Algorithm 2 Path integration

Input: Image $\boldsymbol{x} \in \mathbb{R}^n$, model F, # of baselines B, Total steps K, param = $\{\eta, \epsilon, \delta\}$, Blur kernel sizes $\{\sigma_b\}_{b=1}^B$
Output: Attribution map $\mathcal{A}$

1: Initialize: gradAcc $= \mathbf{0} \in \mathbb{R}^n$, $\boldsymbol{\rho} \leftarrow \nabla_x F$
2: **for** b = 1 to B **do**
3: $\quad \boldsymbol{x}' \leftarrow \text{ComputeBaseline}(\boldsymbol{x}, F, \sigma_b, \text{param})$
4: $\quad$ count $\leftarrow \mathbf{0} \in \mathbb{R}^n$, $\boldsymbol{\varrho} \leftarrow \mathbf{0} \in \mathbb{R}^n$
5: $\quad$ **for** k = 1 to $\lfloor \frac{K}{B} \rfloor$ **do**
6: $\quad\quad \tilde{\boldsymbol{x}} \leftarrow \boldsymbol{x}' + \alpha_k(\boldsymbol{x} - \boldsymbol{x}')$ s.t. $\alpha_k \sim \text{uniform}(0, 1)$
7: $\quad\quad \tilde{\boldsymbol{\rho}} \leftarrow (\boldsymbol{x} - \tilde{\boldsymbol{x}}) \cdot \nabla_{\tilde{x}} F$
8: $\quad\quad$ **for** i = 1 to n **do**
9: $\quad\quad\quad$ **if** $\text{sgn}(\boldsymbol{\rho}_i) = \text{sgn}(\tilde{\boldsymbol{\rho}}_i) \wedge |\boldsymbol{\rho}_i| > |\tilde{\boldsymbol{\rho}}_i|$ **then**
10: $\quad\quad\quad\quad \boldsymbol{\varrho}_i \leftarrow \boldsymbol{\varrho}_i + \nabla_{\tilde{x}_i} F$, $\text{count}_i \leftarrow \text{count}_i + 1$
11: $\quad\quad\quad$ **end if**
12: $\quad\quad$ **end for**
13: $\quad$ **end for**
14: $\quad$ gradAcc $\leftarrow$ gradAcc $+ (\boldsymbol{x} - \boldsymbol{x}')\boldsymbol{\varrho} \cdot /\text{count}$
15: **end for**
16: $\mathcal{A} \leftarrow$ gradAcc $/B$

lines 2 - 7 of Algo. (1), we gradually alter $\boldsymbol{x}^p$ to bring it close to $\boldsymbol{x}$ in the model output space, while maintaining the constrain $\min_i |x_i - x_i^p| \geq \delta$ in the input space. Here, logit scores are used as the output map of F. On line 3, alteration to $\boldsymbol{x}^p$ is guided by Lemma (6.1), which provides us with a desirable direction of altering $\boldsymbol{x}^p$ that can efficiently achieve our objective. We take small steps in that direction with a step size η. On lines 4 - 6, we ensure that $\boldsymbol{x}^p$ abides by $|x_i - x_i^p| \geq \delta$ after each alteration. Line 7 brings the image back to the valid dynamic image range for the model F by the standard clipping operation.

Lemma 6.1. *For $F(.)$ with cross-entropy loss, $F(\boldsymbol{x}^p)$ can approach $F(\boldsymbol{x})$ by stepping in the direction $-sgn(\nabla_{x^p} F)$.*

Proof: *For $F(.)$ with corss-entropy loss $\mathcal{J}(.,.)$, $F(\boldsymbol{x}^p) \rightarrow F(\boldsymbol{x})$ requires maximizing $\log\left(p(F(\boldsymbol{x})|\boldsymbol{x}^p\right)$, which requires stepping in the direction $sgn(\nabla_{x^p} \log(p(F(\boldsymbol{x})|\boldsymbol{x}^p))$. This is the same direction as $sgn\left(-\nabla_{x^p} \mathcal{J}(F(x), \boldsymbol{x}^p)\right)$ or $-sgn(\nabla_{x^p} F)$ following our short-hand notation.*

Gradients integration: Algorithm (2) summarizes our overall technique to compute the attribution map with gradient integration. For clarity, the text below describes it in a non-sequential manner. On line 3 of Algo. (2), we obtain the desired baseline image by calling Algo. (1). Using this baseline, line 6 computes the features that reside on the path between the baseline and the input, employing a step size sampled from a uniform distribution. We also use a linear path function similar to IG (Sundararajan et al., 2017) - *cf.* § 4. This allows our method to inherit the axiomatic properties of the canonical path method. We discuss this aspect in more detail in the supplementary material of the paper while describing the theoretical properties.

After computing the path features, their gradients on the path are estimated on line 7, and the checks specified by Def. (5.3) are performed on line 9. It is straightforward to show that under the Riemman approximation of the integral, gradients for a feature x_i on a linear path can be integrated as $(x_i - x_i') \times \frac{1}{m} \sum_{i=1}^m \nabla_{\tilde{x}_i} F$, where $\nabla_{\tilde{x}_i} F$ is the model gradient w.r.t. the valid path feature $\tilde{x}_i$. On line 10, accumulation of the gradients of the valid features, i.e., $\sum_i \nabla_{\tilde{x}_i} F$, is performed, which is used for the Reimann approximation of the integration by the algorithm.

Besides the above, an outer *for-loop* can be observed in Algo. (2). This loop allows us to use multiple baselines in our path-based attribution. Using multiple baselines can be beneficial in suppressing the noise in accumulated gradients (Erion et al., 2021). Erion et al. (2021) used a Monte Carlo approximation of the integral in their technique to leverage multiple baselines. Inspired, we also use the same approximation, which requires Algo. (2) to estimate the eventual integral as $\mathop{\mathbb{E}}_{\boldsymbol{x}' \sim \mathcal{B}}[(\boldsymbol{x} - \boldsymbol{x}')\nabla_{\tilde{x}} F]$, where $\mathcal{B}$ is the distribution over the proposed baseline. Mathematically, the noted Expectation value leads us to averaging over the gradients which are integrated in the inner *for-loop*. This is accomplished with the outer loop. It is noteworthy that we allow multiple baselines in our method mainly to suppress any potential noise due to the shattered gradients problem. Otherwise, the inner *for-loop* along the baseline computation in Algo. (1) already accounts for the fixes **F1** - **F3** discussed in § 5.

The proposed attribution estimation may at first seem dependent on multiple hyper-parameters. However, these parameters are handles over intuitive concepts, which makes selecting their values straightforward. Moreover, the computed attribution scores are largely insensitive to a wide range of sensible values of these parameters. We also provide further discussion about it in the supplementary material.

Method	ResNet-50	DenseNet-121	VGG-16
IG	0.3711	0.5042	0.2532
IG (G)	0.2997 **-19.2**	0.4128 **-18.1**	0.1609 **-36.4**
IG (A)	0.4653 **+25.4**	0.5493 **+8.9**	0.3387 **+33.7**
AGI	0.3995 **+7.6**	0.4549 **-9.7**	0.2419 **-4.4**
EG	0.4004 **+7.90**	0.5171 **+2.5**	0.2727 **+7.69**
GIG	0.4681 **+26.1**	0.5436 **+7.8**	0.3914 **+54.5**
Our	**0.5311** **+43.1**	**0.6228** **+23.5**	**0.4023** **+58.8**

Table 1. ImageNet (Deng et al., 2009) AUC difference between Insertion-Deletion scores(Petsiuk et al., 2018). Percentage gain over Integrated Gradient (IG) (Sundararajan et al., 2017) is also given. IG (G) and IG (A) respectively use Gaussian noise and Average pixel value of the input as the baseline.

7. Empirical Evidence

This paper contributes to the path-based model interpretation paradigm, hence its experiments are specifically designed to show improvements to the path attribution framework with the newly provided insights. Indeed, there are also other post-hoc interpretation techniques besides path methods, *cf.* § 2. However, they are not axiomatic, which renders their comparison with the path methods injudicious. This is particularly true for quantitative comparisons because to-date there is no mutually agreed upon metric that is known to comprehensively quantify the correctness of attribution maps. It is emphasized that the intent of our evaluation in not to claim new state-of-the-art on performance metrics, which are disputed in the first place. Rather, we use empirical results as an evidence that our theoretical insights positively contribute to the path attribution framework.

Insertion/Deletion evaluation on ImageNet: Among the most commonly used quantitative evaluation metrics for the post-hoc interpretation methods, are the insertion and deletion game scores (Petsiuk et al., 2018). In our evaluation, the insertion game inserts the most important pixel (as computed by the method) first and records the change in the model output. The deletion game conversely records the score by removing the most important pixel first. We conduct insertions and deletions for all the pixels and compute the Area Under the Curve (AUC) of the output change with the pixel insertion/deletion. For insertion, a larger AUC is more desirable, which is opposite for the deletion. It is easy to see that the two metrics do not capture the full picture of method performance individually. Hence, we combine them by reporting the AUC of the difference between the insertion and deletion scores in our results, where the larger differences become more desirable. This provides a more comprehensive view of the performance.

In Table 1, we summarize the results on three popular ImageNet models. As the baseline method, we chose the canonical path attribution technique, i.e., Integrated Gradients (IG) (Sundararajan et al., 2017). We also implement IG, using different path baselines. IG (G) uses Gaussian noise instead of black image as the path baseline, whereas IG (A) uses the average pixel value of the input as the baseline. For each model, the results are averaged over $2,500$ images from the ImageNet validation set. We also include the existing relevant methods Expected Gradient (EG) (Erion et al., 2021), Guided IG (GIG) (Kapishnikov et al., 2021) and Adversarial Gradient Integration (AGI) (Pan et al., 2021) for benchmarking.

Method	RN-50 (0.77)	RN-34 (0.75)	RN-18 (0.64)
IG	0.3711	0.3854	0.2481
Our	**0.5311**+43.1	**0.5185**+34.5	**0.3675**+48.1

Table 2. Performance gain over IG for different ResNet (RN) variants. Average confidence score of models are noted in parenthesis.

In Table 1, we ensure that the methods use the same images and models, and also take the same number of steps from the baseline image to the input. This allows for a transparent comparison. For all the methods, we allow 150 steps. Since our technique enables the use of multiple baselines, we use 3. The same number of baselines and steps are used for EG (Erion et al., 2021) and AGI (Pan et al., 2021). The reported results also include the percentage gains of each technique over IG. Since IG is the canonical path attribution method (Sundararajan et al., 2017), it provides the perfect baseline to establish any positive development for the path attribution framework. It can be noticed that our method achieves remarkable gains, with up to 58.8% improvement for VGG-16.

There are a few reportable interesting observations related to the results in Table 1. We noticed that the average confidence scores of ResNet-50, DenseNet-121 and VGG-16 in our experiments were 0.77, 0.81 and 0.69, respectively. The underlying pattern is exactly the opposite to that of the gains we achieved over IG with our method in Table 1. Indicating that IG has a tendency to perform sub-optimally (relatively speaking) for the less confident models - as adjudged by the insertion/deletion game scores. To further verify that, we report the results of an additional experiment with ResNet (RN) variants in Table 2. Whereas IG gained some grounds for ResNet-34, it again performed relatively poorly for ResNet-18.

Another interesting observation we made related to the results in Table 1, was about the performance of EG (Erion et al., 2021) and AGI (Pan et al., 2021). Whereas we use author-provided codes for these methods, we match the hyper-parameters with IG and our method, and remove any other pre-/post-processing of the computed maps which is not used by IG. For instance, we remove thresholding of AGI, which does not conform to the axioms of path attribution. As can be seen in Table 1, AGI does not perform too well on equal grounds with IG. We use the best performing variant of AGI in our experiments, which was achieved

Method	ResNet-50	DenseNet-121	VGG-16
IG	0.5502	0.4693	0.4873
IG (A)	0.5619+2.1	0.5039+7.4	0.4964+1.8
GIG	0.5635+2.4	0.5023+7.0	0.5014+2.8
Our	**0.5889**+7.1	**0.5448**+16.1	**0.5120**+5.0

Table 3. CIFAR-10 (Krizhevsky et al.) AUC difference between Insertion-Deletion scores (Petsiuk et al., 2018). Average confidence scores of ResNet-50, DenseNet-121 and VGG-16 on the images are 0.81, 0.74 and 0.79, respectively. IG (A) uses average pixel value of input as baseline.

Method	ResNet-50	Dense-121	VGG-16
Our + IG_Baseline	0.3809	0.5112	0.2552
Our (Single_Baseline)	0.5254	0.6192	0.3602
Our (Proposed)	**0.5311**	**0.6228**	**0.4023**

Table 4. Contribution of the proposed baseline and integration process to the final results on ImageNet models. AUC differences reported for insertion/deletion game scores.

with PGD attack (Madry et al., 2018). For EG, we use 3 baselines with 50 steps to match it with our method. This variant performed almost similar to using 150 baselines with 1 step for each baseline. We also provide further results in ImageNet dataset using Vision Transformer (ViT) models in the supplementary material of the paper.

Insertion/Deletion evaluation on CIFAR-10: As compared to 224×224 grid size of ImageNet samples, CIFAR-10 (Krizhevsky et al.) has 32×32 image grid size. Image size has direct implications for the quantitative metrics of insertion and deletion games. Hence, in Table 3, we also report performance of our method on 1000 images of CIFAR-10 validation set. In the table, we only include the top performing approaches from Table 1. On CIFAR-10 images, IG already performed considerably well under insertion/deletion score metrics. Nevertheless, our method still provided a considerable relative gain over IG consistently. It is noteworthy that our observation regarding the relation between the relative gain of our method over IG and the model confidence scores also generally holds in the case of CIFAR-10 experiments.

Sensitivity-N evaluation on ImageNet: Though it is common to evaluate performance of attribution methods under a single quantitative metric (Kapishnikov et al., 2021), (Pan et al., 2021), we further evaluate our approach with Sensitivity-N (Ancona et al., 2017) to conclusively establish its contribution to the path attribution framework. The computationally intensive Sensitivity-N metric comprehensively verifies that the model output is sensitive to the pixels considered important by the attribution method. For any feature subset of $\boldsymbol{x}$, i.e., $\boldsymbol{x}_{\text{sub}} = [x_1, x_2, ..., x_k] \subseteq \boldsymbol{x}$, this metric requires that $\sum_i^k \mathcal{A}_i = f(\boldsymbol{x}) - f(\boldsymbol{x}_{[x_{\text{sub}}=0]})$ holds. Whereas no method is expected to achieve this due to practical reasons, the metric is still effective. To put it into practice, we vary the feature fraction in $\boldsymbol{x}_{\text{sub}}$ in the range [0.01, 0.9], and compute the Pearson Correlation Coefficient between $\sum_i^m \mathcal{A}_i$ values and the output variations.

We plot the results of Sensitivity-N for the ImageNet model interpretations in Fig. 3. Higher values of the curves are more desirable. It is observable that as compared to the canonical method IG, our method generally performs considerably better under this metric as well.

Qualitative results: We show multiple representative qualitative results for random samples using VGG-16, ResNet-50 and DenseNet-121 models for ImageNet. Results of the top-performing methods in Table 1 are included. In those results attribution scores are encoded as gray-scale variations, where brighter pixels represent larger attribution scores. It is clear from the qualitative results that our method does not face the problem of assigning lower scores to the dark pixels of the object. Our maps are also less noisy and indeed assign large attribution scores to the foreground object. We do not observe counter-intuitive behavior of the attributions for our method.

Further results: The ablation analysis in Table 4 shows that both the proposed baseline an path feature integration process contribute positively to our overall performance. In the supplementary material, we provide further results demonstrating the effects of hyper-parameter settings on the performance. The key observation related to those results is that we can even improve performance further by allowing more steps on the path, and our results are generally insensitive to the hyper-parameters values in reasonable ranges. Computational and memory requirements of our method also remain comparable to those of IG.

8. Conclusion

Using theoretical guidelines, this paper pinpointed the sources of three shortcomings of the path attribution framework that compromise its reliability as an interpretation tool for the deep visual models. It proposed fixes to these problems such that the framework becomes fully conformant to the original game-theoretic intuitions that govern its much desired axiomatic properties. We combined these fixes into a concrete path attribution method that can compute reliable explanations of deep visual models. The claims are also established by an extensive empirical evidence to explain a range of deep visual classifiers.

Acknowledgements

Dr. Naveed Akhtar is a recipient of the Office of National Intelligence National Intelligence Postdoctoral Grant (project number NIPG-2021-001) funded by the Australian Government.

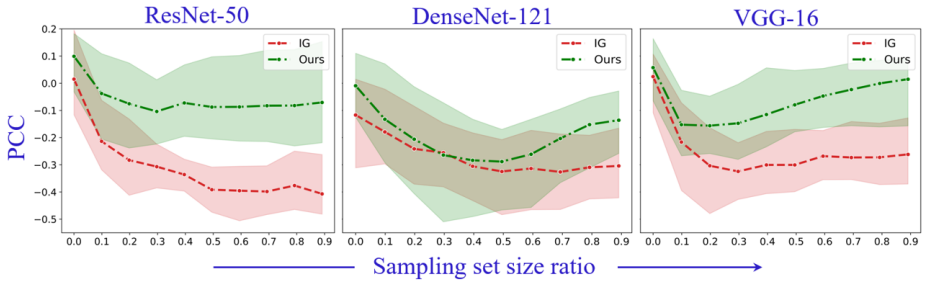

Figure 3. Sensitivity-N (Ancona et al., 2017) analysis on ImageNet models. Pearson Correlation Coefficient (PCC) between the sum of the attributions and output variations under different sampling set size ratios are plotted. Larger values are more desirable. A considerable gain is achieved by our method over IG (Sundararajan et al., 2017).

References

Gravity, alphafold and neural interfaces: a year of remarkable science. https://www.nature.com/articles/d41586-021-03730-w. Accessed: 2022-10-30.

Agarwal, R., Melnick, L., Frosst, N., Zhang, X., Lengerich, B., Caruana, R., and Hinton, G. E. Neural additive models: Interpretable machine learning with neural nets. *Advances in Neural Information Processing Systems*, 34: 4699–4711, 2021.

Akhtar, N. and Jalwana, M. A. Rethinking interpretation: Input-agnostic saliency mapping of deep visual classifiers. *arXiv preprint arXiv:2303.17836*, 2023.

Akhtar, N., Mian, A., Kardan, N., and Shah, M. Advances in adversarial attacks and defenses in computer vision: A survey. *IEEE Access*, 9:155161–155196, 2021.

Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*, 2017.

Aumann, R. J. and Shapley, L. S. *Values of non-atomic games*. Princeton University Press, 2015.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

Balduzzi, D., Frean, M., Leary, L., Lewis, J., Ma, K. W.-D., and McWilliams, B. The shattered gradients problem: If resnets are the answer, then what is the question? In *International Conference on Machine Learning*, pp. 342–350. PMLR, 2017.

Blazek, P. J. and Lin, M. M. Explainable neural networks that simulate reasoning. *Nature Computational Science*, 1(9):607–618, 2021.

Bohle, M., Fritz, M., and Schiele, B. Convolutional dynamic alignment networks for interpretable classifications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10029–10038, 2021.

Böhle, M., Fritz, M., and Schiele, B. B-cos networks: Alignment is all we need for interpretability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10329–10338, 2022.

Brendel, W. and Bethge, M. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *arXiv preprint arXiv:1904.00760*, 2019.

Chattopadhay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pp. 839–847. IEEE, 2018.

Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., and Su, J. K. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

Chen, Z., Bei, Y., and Rudin, C. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

Dabkowski, P. and Gal, Y. Real time image saliency for black box classifiers. *Advances in neural information processing systems*, 30, 2017.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2009.

Donnelly, J., Barnett, A. J., and Chen, C. Deformable protopnet: An interpretable image classifier using deformable prototypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10265–10275, 2022.

Erion, G., Janizek, J. D., Sturmfels, P., Lundberg, S. M., and Lee, S.-I. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature Machine Intelligence*, 2021.

Fong, R. C. and Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pp. 3429–3437, 2017.

Friedman, E. J. Paths and consistency in additive cost sharing. *International Journal of Game Theory*, 32(4):501–518, 2004.

Hooker, S., Erhan, D., Kindermans, P., and Kim, B. A benchmark for interpretability methods in deep neural networks. In *Advances in NeuralInformation Processing Systems, NeurIPS*, 2019.

Jalwana, M. A., Akhtar, N., Bennamoun, M., and Mian, A. Cameras: Enhanced resolution and sanity preserving class activation mapping for image saliency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16327–16336, 2021.

Jiang, P.-T., Zhang, C.-B., Hou, Q., Cheng, M.-M., and Wei, Y. Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactions on Image Processing*, 30:5875–5888, 2021.

Kapishnikov, A., Venugopalan, S., Avci, B., Wedin, B., Terry, M., and Bolukbasi, T. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5050–5058, 2021.

Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., and Liang, P. Concept bottleneck models. In *International Conference on Machine Learning*, pp. 5338–5348. PMLR, 2020.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

Lundstrom, D. D., Huang, T., and Razaviyayn, M. A rigorous study of integrated gradients method and extensions to internal neuron attributions. In *International Conference on Machine Learning*, pp. 14485–14508. PMLR, 2022.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations, ICLR*, 2018.

Pan, D., Li, X., and Zhu, D. Explaining deep neural network models with adversarial gradient integration. In *International Joint Conference on Artificial Intelligence, IJCAI*, 2021.

Parekh, J., Mozharovskyi, P., and d'Alché Buc, F. A framework to learn with interpretation. *Advances in Neural Information Processing Systems*, 34:24273–24285, 2021.

Petsiuk, V., Das, A., and Saenko, K. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

Ramaswamy, H. G. et al. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 983–991, 2020.

Ribeiro, M. T., Singh, S., and Guestrin, C. " 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*, pp. 1135–1144, 2016.

Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Sarkar, A., Vijaykeerthy, D., Sarkar, A., and Balasubramanian, V. N. A framework for learning ante-hoc explainable models via concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10286–10295, 2022.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMLR, 2017.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop on International Conference on Learning Representations, ICLR*, 2014.

Slack, D., Hilgard, A., Singh, S., and Lakkaraju, H. Reliable post hoc explanations: Modeling uncertainty in explainability. *Advances in Neural Information Processing Systems*, 34:9391–9404, 2021.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Srinivas, S. and Fleuret, F. Full-gradient representation for neural network visualization. In *Advances in NeuralInformation Processing Systems, NeurIPS*, 2019.

Sturmfels, P., Lundberg, S., and Lee, S.-I. Visualizing the impact of feature attribution baselines. *Distill*, 2020.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Machine Learning, ICML*, 2017.

Vinuesa, R. and Sirmacek, B. Interpretable deep-learning models to help achieve the sustainable development goals. *Nature Machine Intelligence*, 3(11):926–926, 2021.

Vinuesa, R., Azizpour, H., Leite, I., Balaam, M., Dignum, V., Domisch, S., Felländer, A., Langhans, S. D., Tegmark, M., and Fuso Nerini, F. The role of artificial intelligence in achieving the sustainable development goals. *Nature communications*, 11(1):1–10, 2020.

Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., and Hu, X. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 24–25, 2020.

Xu, S., Venugopalan, S., and Sundararajan, M. Attribution in scale and space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9680–9689, 2020.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Zhang, J., Bargal, S. A., Lin, Z., Brandt, J., Shen, X., and Sclaroff, S. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126 (10):1084–1102, 2018.

Supplementary material

A. Axiomatic properties and their retention

Below we list the axiomatic properties of the path attribution framework. These properties are commonly claimed by multiple existing works, e.g., (Lundstrom et al., 2022), (Sundararajan et al., 2017), (Xu et al., 2020). Besides listing the properties, we provide their intuitive meanings and remark on how our method retains each property.

Definition A.1 (Completeness)**.** For $F \in \mathcal{F}(\boldsymbol{a}, \boldsymbol{b})$ and $\boldsymbol{x}, \boldsymbol{x}' \in [\boldsymbol{a}, \boldsymbol{b}]$, we have $\sum_{i=1}^{n} \mathcal{A}_i(\boldsymbol{x}, \boldsymbol{x}') = F(\boldsymbol{x}) - F(\boldsymbol{x}')$.

Completeness ensures that a non-zero importance is attributed to a feature only when that feature actually contributes to the model output. For a linear path function $\gamma(\alpha) = \boldsymbol{x}' + \alpha \times (\boldsymbol{x} - \boldsymbol{x}')$, it is provable under the fundamental theorem of calculus that $\sum_{i=1}^{n}(x_i - x_i') \times \int_{\alpha=1}^{1} \frac{\partial F(\gamma(\alpha))}{\partial x_i} d\alpha = F(\boldsymbol{x}) - F(\boldsymbol{x}')$. Since for any baseline (considering multiple baselines), our method also follows a linear path - similar to IG (Sundararajan et al., 2017) - for differentiable F, the fundamental theorem of calculus is similarly applicable to our method. We impose $||F(\boldsymbol{x}) - F(\boldsymbol{x}')||_2 \to 0$. From the implementation viewpoint, we have $\sum_{i=1}^{n}(x_i - x_i') \times \int_{\alpha=1}^{1} \frac{\partial F(\gamma(\alpha))}{\partial x_i} d\alpha = \epsilon$. Our path integration does not introduce any new out-of-path features. It selects all the features from the same linear paths without violating path integration. Hence, as $d\alpha \to 0$ in the above integral, R.H.S.$\to 0$.

Definition A.2 (Sensitivity-A)**.** Let $\boldsymbol{x}$ and $\boldsymbol{x}'$ vary in one component, s.t. $x_i \neq x_i', \wedge\, x_j = x_j' \; \forall j \neq i$. Moreover, let $F(\boldsymbol{x}) \neq F(\boldsymbol{x}')$. Then $\mathcal{A}_i(\boldsymbol{x}, \boldsymbol{x}') \neq 0$.

Sensitivity-A asserts that when a feature contributes to output, it gets non-zero attribution. Completeness implies sensitivity. Since our method upholds completeness, sensitivity-A remains satisfied.

Definition A.3 (Implementation invariance)**.** $\mathcal{A}$ is not a function of model implementation. Instead, it is only a function of the mathematical mapping of the domain to the range as performed by the model.

Path attribution framework ensures implementation invariance by not relying on the internal signals of the network. For instance, GradCAM (Selvaraju et al., 2017) needs activations of a certain layer to compute attributions. When two models map a domain to the same range, but using different network architectures, GradCAM's map can get affected by the network architecture itself. This violates implementation invariance. Path based methods solely rely on the input-output mapping by backpropagating the gradients right until the input. Our method also does the same, and does not contradict the fundamental principles of gradient integration. Hence, it retains the implementation invariance property just like other path-based methods. Notice that, even for computing the baseline(s), we backpropagate the gradients until the input, which is inline with the implementation invariance requirements.

Definition A.4 (Linearity)**.** For $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$, $\mathcal{A}_i(\boldsymbol{x}, \boldsymbol{x}_i', \alpha F_1 + \beta F_2) = \alpha \mathcal{A}_i(\boldsymbol{x}, \boldsymbol{x}', F_1) + \beta \mathcal{A}_i(\boldsymbol{x}, \boldsymbol{x}', F_2)$.

Linear path integration axiomatically satisfies linearity. When using multiple baselines, our method uses linear path integration for each baseline. The eventual averaging under the Monte Carlo integration is also a linear operation. By definition, applying it to linear path integration still preserves linearity. Hence, this property is also retained by our method.

Definition A.5 (Sensitivity-B)**.** For any $F \in \mathcal{F}$, when $\frac{\partial F}{\partial x_i} = 0$ if $\mathcal{A}_i(\boldsymbol{x}, \boldsymbol{x}_i') = 0$.

This property asserts that if the function does not depend on a variable, then the attribution score of that variable is zero. As a natural complement of Sensitivity-A - *cf.* Def. (A.2) - linear path integration methods satisfy this property axiomatically. As noted above, each path integration used in our method satisfies linearity - *cf.* Def. (A.4), and the final averaging in the case of multiple baselines is also a linear operation. Hence, Sensitivity-B is also retained.

Definition A.6 (Symmetry Preserving)**.** For a give index pair (i, j), $x^{\dagger}$ is formed by swapping the value of x_i and x_j. If $\forall \boldsymbol{x} \in [\boldsymbol{a}, \boldsymbol{b}]$, $F(\boldsymbol{x}) = F(\boldsymbol{x}^{\dagger})$, then whenever $x_i = x_j$ and $x_i' = x_j'$, the $\mathcal{A}_i(\boldsymbol{x}, \boldsymbol{x}') = \mathcal{A}_j(\boldsymbol{x}, \boldsymbol{x}')$.

This property asserts that ‘‘*if two variables play the exact same role in the network then they ought to receive the same attribution*" (Sundararajan et al., 2017). Sundrarajan et al. (Sundararajan et al., 2017) prove the path integration with linear paths preserves this property. Since the linear path integration also exhibits linearity, averaging (a linear combination) over multiple linear path integrations retains this property.

It is noted that Lundstrom et al. (2022) recently qualified the axiomatic property claims of (Sundararajan et al., 2017) by imposing a non-decreasing positivity (NDP) constraint over the methods. Nevertheless, they verified that linear path integration satisfies NDP. Hence, the linear path used in our method still satisfies the axiomatic properties under the new insights from (Lundstrom et al., 2022). In addition to the above properties, the main paper also shows that our method also satisfies the ‘weak dependence’ property, which is not satisfied by other path attribution methods, e.g., IG (Sundararajan et al., 2017). This gives our method an additional theoretical advantage, besides practically improving the results.

B. Proof of Proposition 5.1

In Prop. 5.1 of the main paper, we state that *"For a piece-wise linear function F, path attribution satisfies both completeness and weak dependence simultaneously when the baseline $\boldsymbol{x}'$ and the input $\boldsymbol{x}$ belong to the same open connected set $\mathcal{U}_i$."* Below we prove this statement. In the proof, we follow (Srinivas & Fleuret, 2019) at first, and then diverge to prove our proposition.

Proof: Attribution computation implies $\exists \Psi : (F, \boldsymbol{x}) \rightarrow \mathcal{A}$, where Ψ denotes a mapping. Following the convention from the main paper, let $\mathcal{U}_{i \in \{1,\dots,n\}}$ be the open-connected set for a family of piece-wise linear functions, whose members get specified by the parameter set $\boldsymbol{\Theta} = \{\boldsymbol{w}_i, b_i | i \in [1, n]\} \in \mathbb{R}^{n \times (D+1)}$, where $\boldsymbol{w}_i \in \mathbb{R}^D$. Let F and $\overline{F}$ be two members of this family, specified by $\boldsymbol{\Theta}$ and $\overline{\boldsymbol{\Theta}}$.

The weak dependence property enforces that the mapping Ψ depends on the input through the parameters of the function, i.e., $\Psi|_{\mathcal{U}_i} : (\boldsymbol{w}_i, b_i) \rightarrow \mathcal{A}$. Since $(\boldsymbol{w}_i, b_i) \in \mathbb{R}^{D+1}$ and $\mathcal{A} \in \mathbb{R}^D$, this asserts that $\Psi|_{\mathcal{U}_i}$ is a many-to-one mapping. Implying, $\exists\ F, \overline{F}$ with corresponding $\theta_i = (\boldsymbol{w}_i, b_i)$ and $\overline{\theta_i} = (\overline{\boldsymbol{w}_i}, \overline{b_i})$ such that $\theta_i \neq \overline{\theta_i}$, yet they map to the same $\mathcal{A}$. For the same $F, \overline{F}$, completeness requires $\Delta = F(\boldsymbol{x}) - F(\boldsymbol{x}') = \boldsymbol{w}_i^\intercal \boldsymbol{x} + b_i - \boldsymbol{w}_j^\intercal \boldsymbol{x}' - b_j$ and similarly $\overline{\Delta} = \overline{F}(\boldsymbol{x}) - \overline{F}(\boldsymbol{x}') = \overline{\boldsymbol{w}}_i^\intercal \boldsymbol{x} + \overline{b_i} - \overline{\boldsymbol{w}}_j^\intercal \boldsymbol{x}' - \overline{b_j}$ such that $\Delta = \overline{\Delta}$. Implying, to hold completeness when weak dependence is satisfied, we need to satisfy $C : (\boldsymbol{w}_i - \overline{\boldsymbol{w}}_i)^\intercal \boldsymbol{x} + (b_i - \overline{b_i}) = (\boldsymbol{w}_j - \overline{\boldsymbol{w}}_j)^\intercal \boldsymbol{x}' + (b_j - \overline{b_j})$. Since we compute attributions w.r.t. a single model, $\overline{F}$ is an identity mapping of F. In that case, the condition C is automatically satisfied when $\boldsymbol{w}_i = \boldsymbol{w}_j$ and $b_i = b_j$ even when $\boldsymbol{x} \neq \boldsymbol{x}'$. Hence, both properties are simultaneously achieved.

C. Further discussion on the baseline

While explaining Def. 5.2 in the main paper, we note that the constraint $||F(\boldsymbol{x}) - F(\boldsymbol{x}')||_2 \approx 0$ encourages $\boldsymbol{x}'$ to use similar weights as $\boldsymbol{x}$. We explain this phenomenon further here. A typical deep visual classifiers can be expressed as $F(x) = \mathcal{C}(\boldsymbol{w}_c, \mathcal{R}(\boldsymbol{w}_r, \boldsymbol{x})) : \boldsymbol{x} \rightarrow \boldsymbol{l}_x \in \mathbb{R}^L$, where $\mathcal{C}(.,.)$ and $\mathcal{R}(.,.)$ are respectively the classification and representation stages of the model. The constraint essentially imposes that the logit scores for the baseline and the input are similar. Normally, $R = \zeta L$, where $\mathcal{R}(.,.) : \boldsymbol{x} \rightarrow \mathbb{R}^R$ and $\zeta \geq 1$. That is, in a typical classifier, the classification stage $\mathcal{C}(.)$ receives an input (normally termed feature vector) that has comparable dimensions with the prediction vector $\boldsymbol{l}_x$. For clarity, let us call that feature vector $f_r \in \mathbb{R}^R$. In that case, the mapping by the classification stage can be summarized as $\boldsymbol{l}_x = \sigma(\boldsymbol{W}_c f_r + \boldsymbol{b}_c)$, where $\sigma(.)$ is an activation function (typically ReLU) and $\boldsymbol{W}_c$ and $\boldsymbol{b}_c$ are the weights and biases of the classification stage. We are already imposing that $\boldsymbol{l}_x = \boldsymbol{l}_{x'}$ (under $||F(\boldsymbol{x}) - F(\boldsymbol{x}')||_2 = 0$). For this condition to hold, we need $\boldsymbol{W}_c f_r + \boldsymbol{b}_c = \boldsymbol{W}_c f'_r + \boldsymbol{b}_c$, where f'_r denotes the baseline image feature. Since the bias term is not influenced by the feature vector, we can ignore that. When $\zeta \geq 1$, the condition $\boldsymbol{W}_c f_r = \boldsymbol{W}_c f'_r$ implies that the same coefficients of $\boldsymbol{W}_c$ will get invoked to satisfy $\boldsymbol{l}_x = \boldsymbol{l}_{x'}$ whenever $f_r = \delta f'_r$, where δ is a real scalar value excluding 0. Indeed, this is a hard condition to satisfy. However, our $\boldsymbol{x}'$ is only a perturbed blurred version of the $\boldsymbol{x}$, where $|x_i - x'_i| \geq \delta$ s.t. δ is a small real value. In that case, we can expect $||f_r - f'_r||_2$ to be a small value. This similarity in the feature vectors encourages the use of similar classifier weights to ensure $||F(\boldsymbol{x}) - F(\boldsymbol{x}')||_2 \rightarrow 0$.

D. Discussion on hyper-parameters

In the main paper, we mention that the hyper-parameters used by our method are handles over intuitive concepts, which makes selecting their values quite easy. Moreover, our method's performance remains largely insensitive to a wide range of hyper-parameter values. First, to justify the latter claim, we present further results in Fig. 4-6 in this document. There are three hyper-parameters in our technique that are of critical nature in the context of path attribution framework. (i) Number of steps taken from the baseline to the input, (ii) number of baselines used, and (iii) the threshold δ. The reported results in the main paper, and qualitative results shown in § E of this document use 150 steps, 3 baselines and $\delta = 5$. It is clear from Fig. 4-6 that the performance of our method is largely insensitive to these values in their neighborhood. We observed in our experiments that for all the used path attribution methods, more than 50 steps often resulted in incremental performance gains. Hence, 150 steps were finally chosen for all the methods to be on the safer side. Figure 5 also shows that the number of baselines in not a major influencer in our performance. We still prefer more than 1 baseline to reduce the effect of noise due to the shattered gradient problem.

In our method, threshold δ is an important hyper-parameter because it is responsible for maintaining a minimum difference between the input pixels and the baseline pixels. Figure 6 show that for the tested range of [1,11], the performance response is almost flat. Hence, we simply chose $\delta = 5$ in our experiments. Choosing a very large δ, e.g., > 15 can cause convergence problem because the optimization objective gets considerably hard. Among the remaining hyper-parameters, are the blur kernel sizes $\{\sigma_b\}_{b=1}^B$, η and ϵ. Notice that, we use blurring only as a method to take the baseline away from the input. The blurred image is used to only initialize the baseline. It is latter processed under Algo. 1 of the main paper to create a baseline that satisfies Eq. (4) of the main paper. To perform the initialization, we simply use fixed blur kernels of size 51 for ImageNet images and 7 for CIFAR-10 images. Where multiple kernels are required, we reduce these sizes by 1 for a new size. This

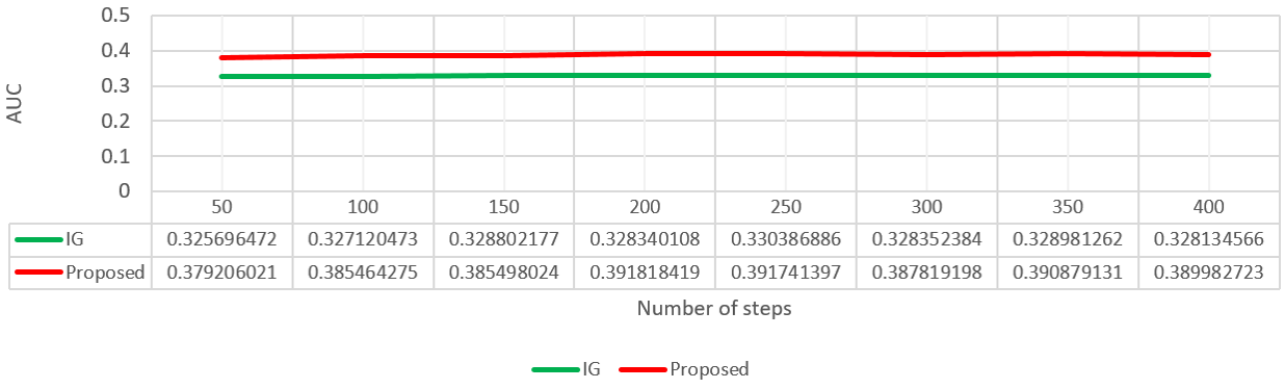

Figure 4. AUC difference between insertion and deletion game scores with varying number of steps for gradient integration. Results are averaged over 50 ImageNet images. ResNet-50 is used in the experiment. Higher values are more desirable.

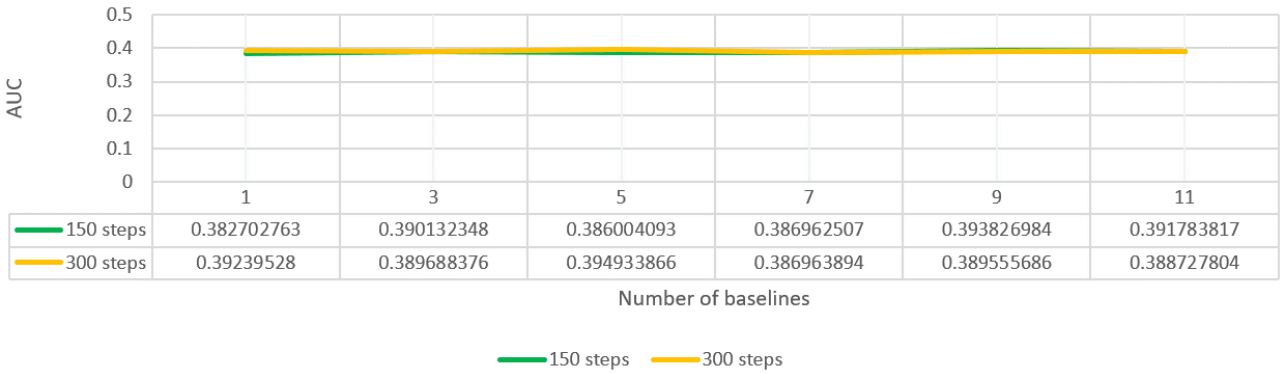

Figure 5. AUC difference between insertion and deletion game scores with varying number of baseline images used for the proposed method. Results are averaged over 50 ImageNet images. ResNet-50 is used in the experiment.

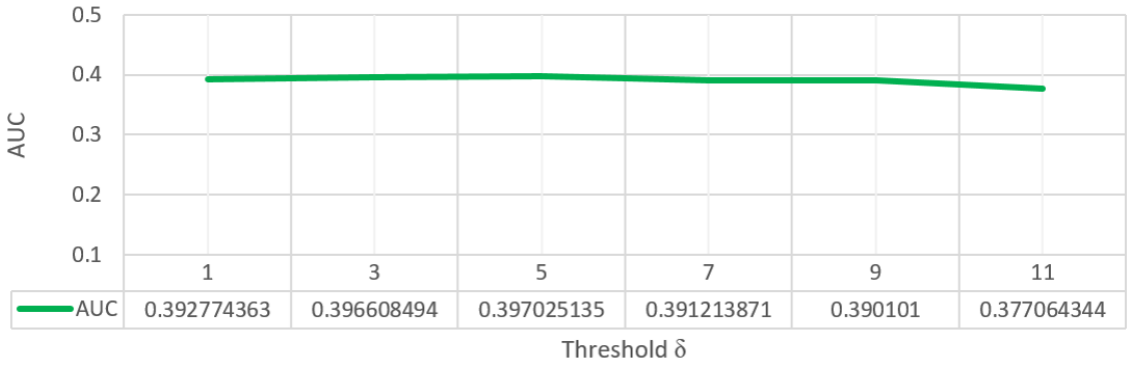

Figure 6. AUC difference between insertion and deletion game scores with varying threshold value of δ. Results are averaged over 50 ImageNet images. ResNet-50 is used in the experiment.

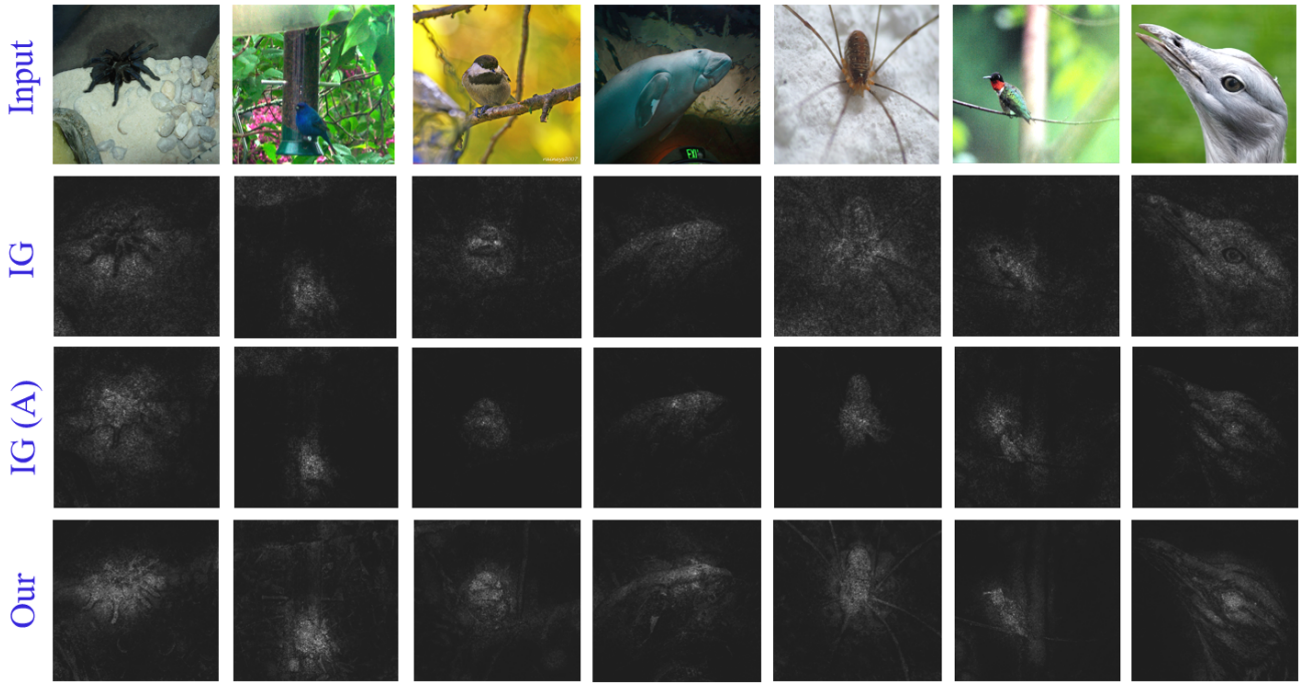

Figure 7. Representative visualizations of ImageNet image attributions with **VGG-16** predictions. Best viewed enlarged on screen.

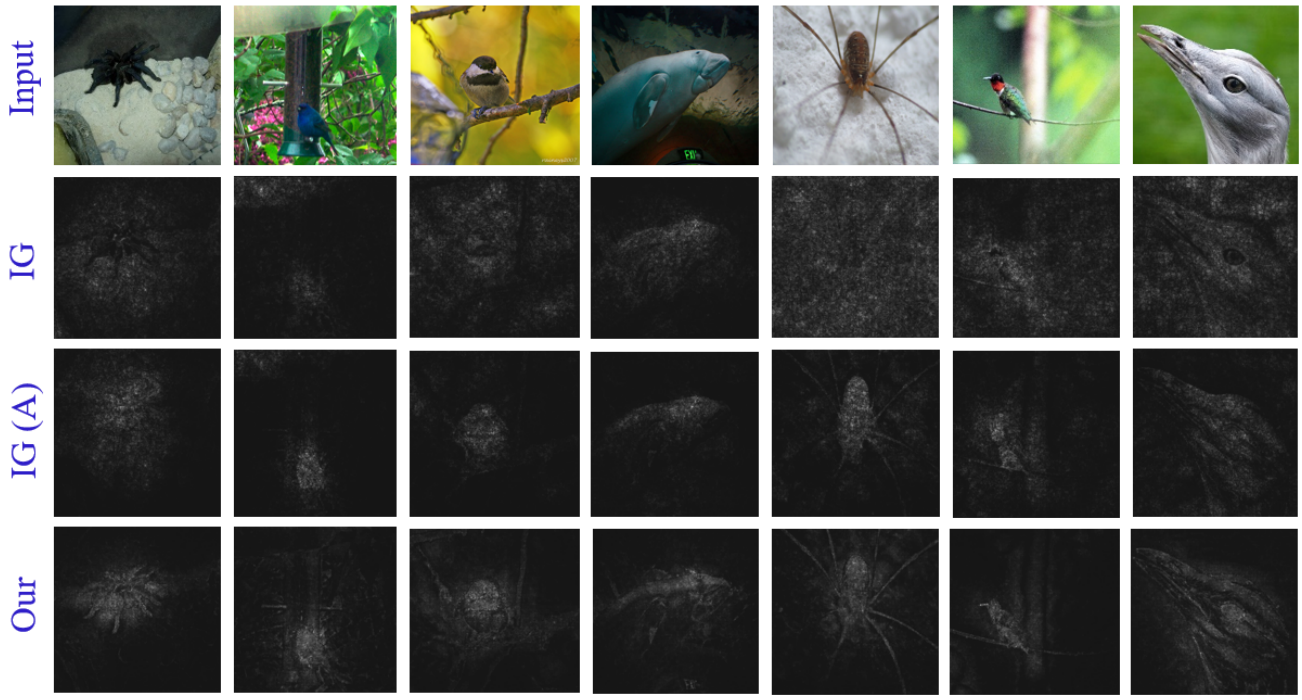

Figure 8. Representative visualizations of ImageNet image attributions with **ResNet-50** predictions. Best viewed enlarged on screen.

is to introduce some variation in the initialization.

Regarding the parameters η and ϵ, the former signifies how strongly we would like to match the logit scores of the baseline and the input. The latter is determines how aggressively we want to alter the baseline to match the logits. We empirically noticed that with $\eta = 1/255$, the logits almost always matched reasonably well after 15 iterations. Hence, from the implementation viewpoint, we replaced ϵ with 15 iterations. This also allows a more uniform computational time across the samples for our method.

E. Qualitative results

In Fig. 8-12 of this document, we provide qualitative results. In Fig. 7, 8 and Fig. 9, the results are reported for VGG-16, ResNet-50 and DenseNet-121, respectively. In Fig. 10-12, we show further results on other random samples for all three models. In the shown figures, a few observation can be easily made. First, it is noticeable that IG keeps struggling to explain the images where the object of interest is relatively darker than the background. The reason for this phenomenon is that IG uses a zero image as the baseline. Hence, the $(x_i - x_i')$ factor in Eq. 2 of the main paper becomes dominant for IG whenever any pixel in the input image is too bright. This inadvertently results in assigning higher attribution to that pixel, even if its in the background. On the other hand, a dark pixel in the input leads to $(x_i - x_i') \rightarrow 0$, even when it is on the object of interest. This results in lower attributions of object pixels when they are dark. IG(A) is able to generally resolve this problem and results in better attribution maps. However, it can be noticed that IG(A) often still assigns high attributions to irrelevant objects in the background. These assignments can come in clusters. E.g., observe the top-left corner of the second image (from left) for IG(A) in Fig. 8 and 9. These are counter-intuitive attributions that do not get resolved with uniform baselines, and are inherited from the IG scheme.

We can observe that the qualitative results of our method are generally less noisy, and high attribution scores are generally focused on the objects. In many cases, immediate background and related objects in the background also get relatively high scores for our method. We conjecture that this is because the models do not only recognize the objects but also their silhouettes and consistent background object (e.g., trees/leafs for birds) to predict the label.

F. Computation time

Attribution mapping in general is an off-line process, hence reliability out-weights timing requirements heavily. Nevertheless, it is noteworthy that our method computes the attribution maps in a time that is comparable to IG. Our overall technique has two major components. (a) Baseline computation, and (b) path integration. Being a path attribution method, the path integration process is very similar to IG, with the exception of extra computations to check for the valid path features. The baseline computation incurs extra cost over IG. However, this process is also not computationally prohibitive. In Table 5, we report the average computational time (in seconds) required by our method and IG for both ImageNet and CIFAR-10 models, computed for NVIDIA RTX 3090 with 24GB RAM using a Pytorch implementation. It can be observed that whereas our method is computationally slightly expensive than IG, the computational time remains comparable to IG. Our method does not incur any significant extra memory cost as well, like EG (Erion et al., 2021) that requires the training data of the model to be available during inference to be used as baselines. We compute the baseline from the input itself.

G. Baseline examples

In Fig. 13, we present examples of the baseline images created y our technique for two random input images. We use $\delta = 5$ and use the blur kernel size 51 for initialization. We ran five iterations of the optimization algorithm. The baseline has some similarity with the simple blurred version of the input. However, the optimization results in spatially correlated noise patterns, which are unique to our technique.

H. Further results

We also tested the methods with (a) ViT-B/32, (b) ViT-L/32 and (c) ViT-L/16 using the public Pytorch models. We used 2500 ImagNet images. For the Integrated Gradients (IG) and our method, we achieved the following results (a) IG: 0.406, Our: 0.473, Gain: 16.5%, (b) IG: 0.461, Our: 0.525, Gain: 13.9%; and (c) IG: 0.511, Our: 0.641, Gain: 25.4%. An interesting observation in our experiments was that IG maps generally contained blocky patterns of (incorrect) attributions due to the patches employed by ViT. Our method naturally addressed the issue very well. Except for a few images, the (incorrect) blocky patterns disappeared for our results. There was no change required to apply our method to ViT.

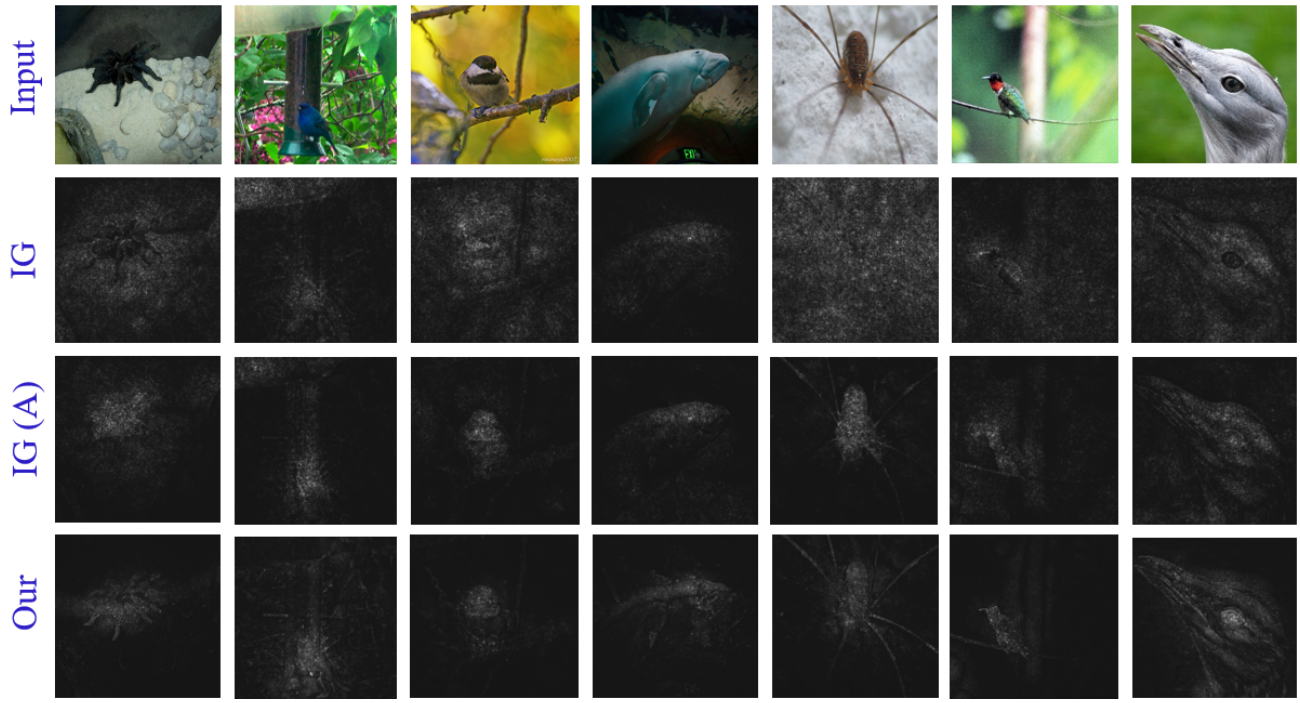

Figure 9. Representative visualizations of ImageNet image attributions with **DenseNet-121** predictions. Best viewed enlarged on screen.

Table 5. Average computation time (in seconds) per image for ImageNet and CIFAR-10 models. Our method can be considered to have two phases (a) baseline computation, (b) path integration. The results reported for our method adopt the convention 'Total time (time for (a) + time for (b))'. The timings are for Pytorch implementation, using NVIDIA RTX 3090 with 24GB RAM.

Integrated Gradient (IG)			Our		
ImageNet			ImageNet		
VGG-16	ResNet-50	DenseNet-121	VGG-16	ResNet-50	DenseNet-121
1.1	0.82	0.9	1.56 (0.36 + 1.2)	1.16 (0.33 + 0.83)	1.43 (0.41 + 1.02)
CIFAR-10			CIFAR-10		
VGG-16	ResNet-50	DenseNet-121	VGG-16	ResNet-50	DenseNet-121
0.44	0.46	0.4	0.80 (0.32 +0.48)	0.98 (0.49+0.49)	0.89 (0.42 + 0.47)

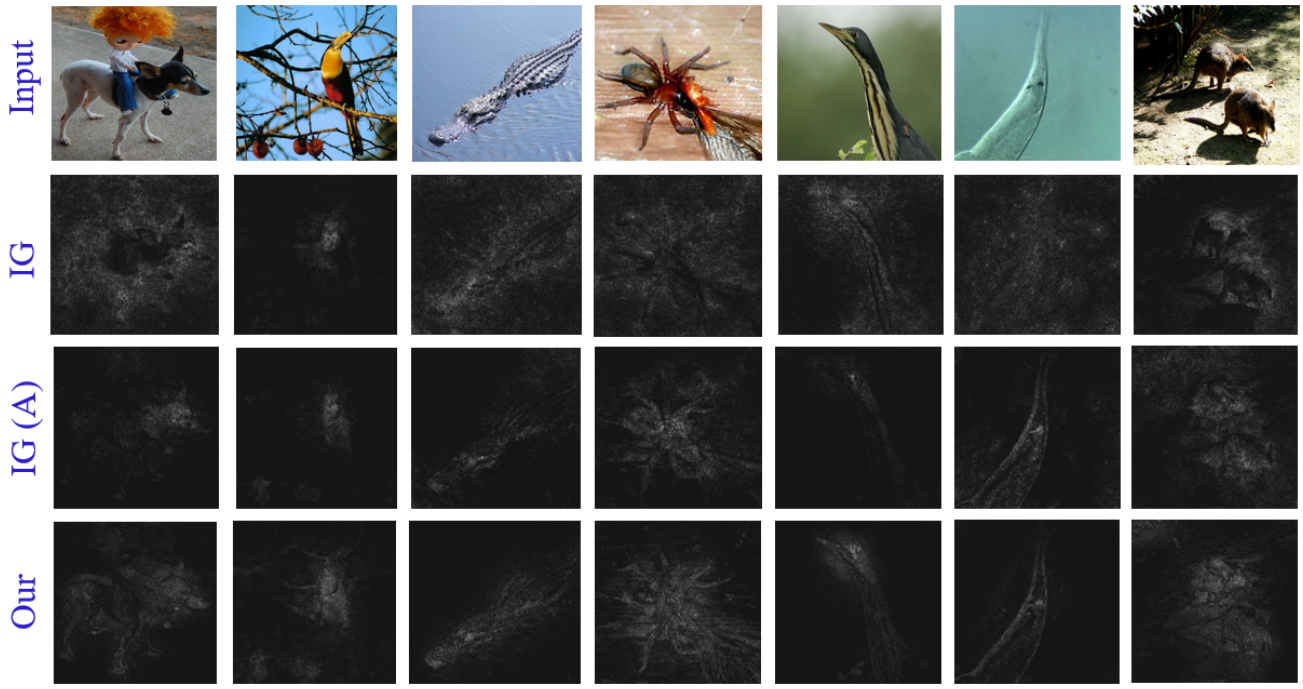

Figure 10. Further representative results of ImageNet image attributions with **VGG-16** predictions.

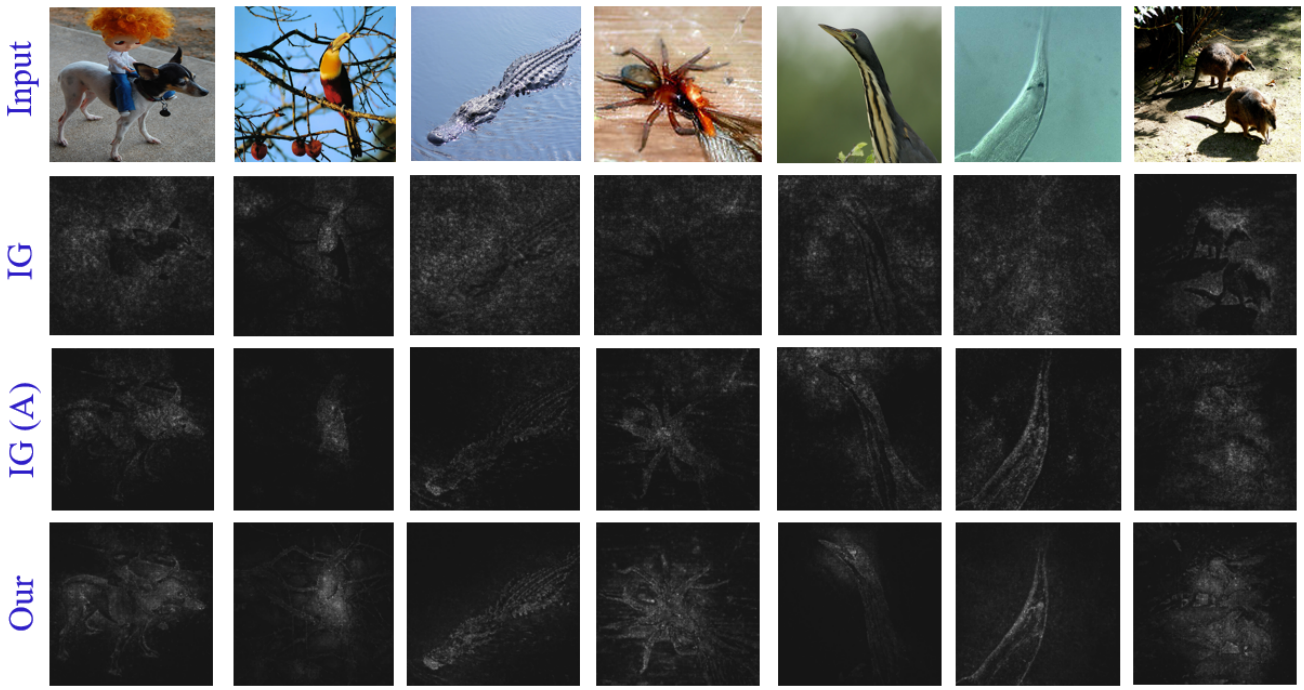

Figure 11. Further representative results of ImageNet image attributions with **ResNet-50** predictions.

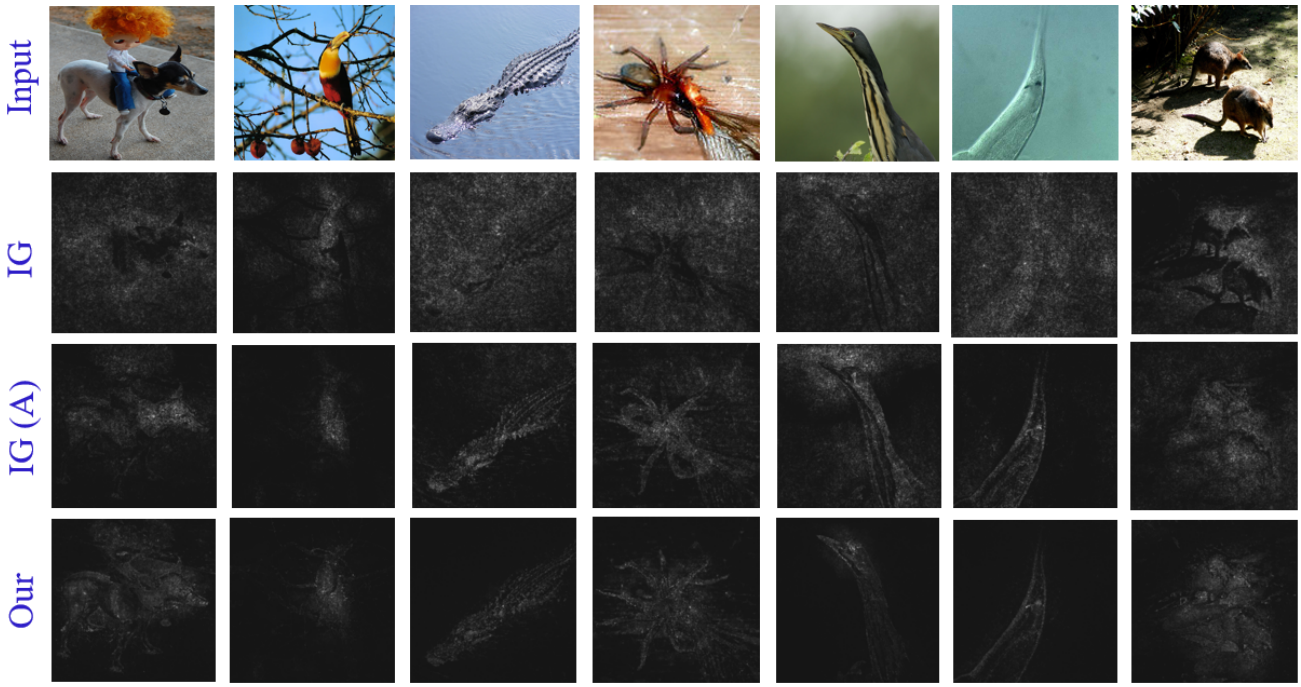

Figure 12. Further representative results of ImageNet image attributions with **DenseNet-121** predictions.

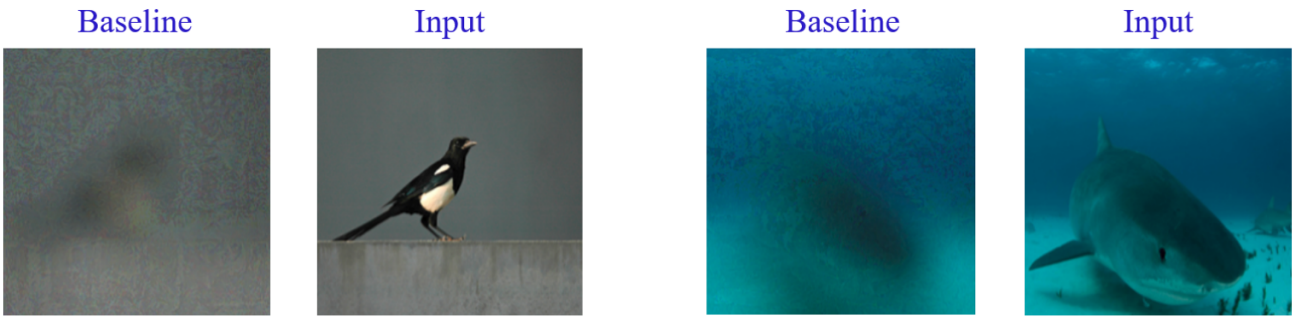

Figure 13. Examples of baseline images generated with the input images