Guided Integrated Gradients: an Adaptive Path Method for Removing Noise

Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, Tolga Bolukbasi
Google Research
{kapishnikov, vsubhashini, besim, wedin, michaelterry, tolgab}@google.com

Abstract

Integrated Gradients (IG) [29] is a commonly used feature attribution method for deep neural networks. While IG has many desirable properties, the method often produces spurious/noisy pixel attributions in regions that are not related to the predicted class when applied to visual models. While this has been previously noted [27], most existing solutions [25, 17] are aimed at addressing the symptoms by explicitly reducing the noise in the resulting attributions. In this work, we show that one of the causes of the problem is the accumulation of noise along the IG path. To minimize the effect of this source of noise, we propose adapting the attribution path itself - conditioning the path not just on the image but also on the model being explained. We introduce Adaptive Path Methods (APMs) as a generalization of path methods, and Guided IG as a specific instance of an APM. Empirically, Guided IG creates saliency maps better aligned with the model's prediction and the input image that is being explained. We show through qualitative and quantitative experiments that Guided IG outperforms other, related methods in nearly every experiment.

1. Introduction

As deep neural network computer vision models are integrated into critical applications such as healthcare and security, research on explaining these models has intensified. Feature attribution techniques strive to explain which inputs the model considers to be most important for a given prediction, making them useful tools in debugging models or understanding what they have likely learned. However, while a plethora of techniques have been developed [24, 29, 13, 9, 20], there are still behaviors of these attribution techniques that remain to be understood. In this context, our work is focused on studying the source of noise in attributions produced by path-integral-based methods [27].

Gradient-based feature attribution techniques [24, 22] are of particular interest in our work. The main idea behind these techniques is that the partial derivative of the output with respect to the input is considered as a measure of the sensitivity of the network for each input dimension. While early methods [24] use the gradients multiplied by the input as a feature attribution technique, more recent methods exploit gradients of the activation maps [22], or integrate gradients over a path [29]. This work studies Integrated Gradients (IG) [29], a commonly used method that is based on game-theoretic ideas in [1]. IG avoids the problem of diminishing influences of features due to gradient saturation, and has desirable theoretical properties.

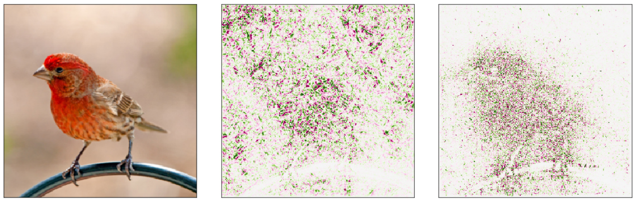
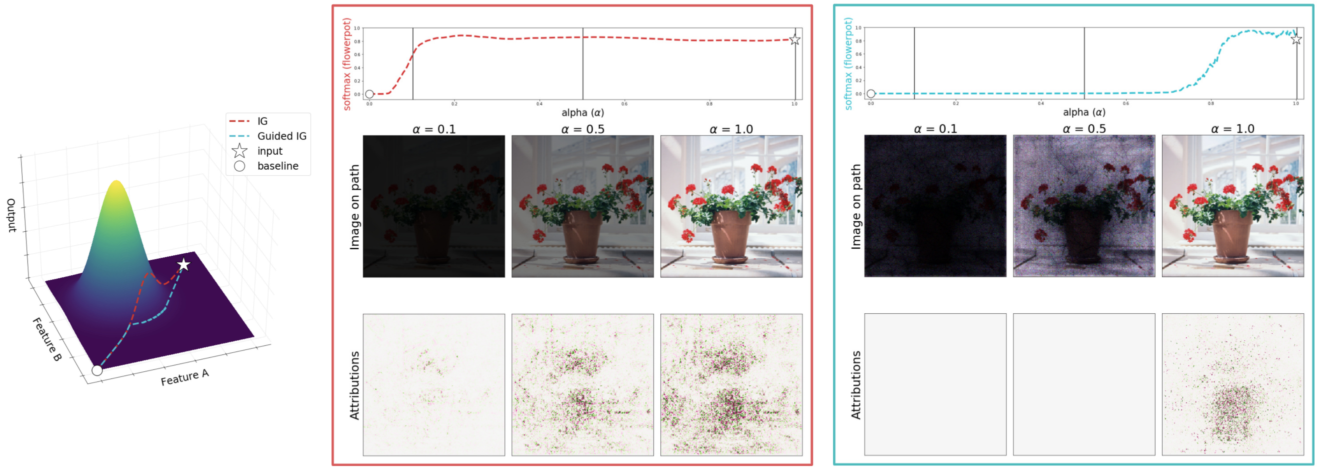

(a) Input image (b) IG attributions (c) Guided IG

Figure 1: Comparing feature attribution for Integrated Gradients and Guided Integrated Gradients. Both (b) and (c) use a black baseline to explain the "house finch" prediction. While (b) has attributions on the bird, there is substantial noise in the attributions compared to (c). This work studies the source of noise, and presents Guided IG as a solution.

One commonly observed problem while calculating Integrated Gradients for vision models is the noise in pixel attribution (Figure 1) originating from gradient accumulation [11, 25, 27, 30] along the integration path. A few possible explanations for this noise have been put forth: (a) high curvature of the output manifold [7]; (b) approximation of the integration with Riemann sum; and (c) choice of baselines [30, 27]. Our experiments indicate that one source of attribution noise comes from regions of correlated, high-magnitude gradients on irrelevant pixels found along the straight line integration path. Our findings correlate with observations in [7], which state that the model surface plays a large role in determining the magnitude of attribution values.

Methods have been proposed to explicitly reduce the noise in attributions. SmoothGrad [25] averages attributions over multiple samples of the input, created by adding Gaussian noise to the original input. The aggregation im-



(a) Schematic

(b) IG attributions

(c) Guided IG attributions

Figure 2: Comparing IG and Guided IG's paths and results. **(a)**: For IG, a straight line path from baseline to input is followed (red dotted line), regardless of changes in gradients. For Guided IG, the path is chosen by selecting features that have the smallest absolute value of corresponding partial derivatives (cyan dotted line). Guided IG's goal is to reduce the accumulation of gradients caused by nearby very high/very low prediction examples. **(b)** and **(c)**: Snapshots of attributions for the flower pot class for Integrated Gradients (center) and Guided IG (right) at alpha values of 0.1, 0.5, and 1.0. The top rows show graphs of the softmax prediction for flower pot as a function of alpha. The second row shows the input image produced by each technique at the three different alpha values. Note that IG's straight line path affects all pixels equally (e.g., see $\alpha = 0.5$), while Guided IG reveals the least important features, first. The third row shows each technique's attributions for each of the three alpha values, with Guided IG showing less noise outside the area of the image occupied by the flower pot.

proves the overall true signal in the attribution. XRAI [12] aggregates attributions within segments to reduce the outlier effect. Sturmfels et al. suggest the choice in baseline is a contributing factor, and propose different baselines as a potential solution [27]. [30] integrate over the frequency dimension by blurring the input image, thereby reducing perturbation artifacts along the attribution path. Dombrowski et al. smooth the network output by converting ReLUs to softplus [7].

While the above methods address noise in the attributions by manipulating the input (or the baseline), ours examines the entire path of integration. As mentioned in [29], each path from the baseline to the input constitutes a different attribution method; the methods discussed above [29, 25, 12, 27] choose the straight line path, while [30] choose the ‘’blur” path when integrating gradients. In this work, instead of determining the path based on the input and baseline alone, we propose *adaptive path methods* (APMs) that adapt the path based on the input, baseline, and the model being explained. Our intuition is that model-agnostic paths, such as the straight line, are susceptible to travel through regions that have irregular gradients, resulting in noisy attributions. We posit that adapting the integration path based on the model can avoid selecting samples from anomalous regions when determining attributions.

We propose Guided Integrated Gradients (Guided IG), an attribution method that integrates gradients along an adaptive path determined by the input, baseline, and the model. Guided IG defines a path from the baseline towards the input, moving in the direction of features that have the lowest absolute value of partial derivatives. At each step, Guided IG selects the features (pixel intensities) with the lowest absolute value of partial derivatives (e.g., bottom 10%), and moves only that subset closer to the intensity in the input image, leaving all others unchanged. As the intensity of specific pixels (features) becomes equal to that in the input image being explained, they are no longer candidates for selection. The attributions resulting from this approach are considerably less noisy. Experiments highlight that Guided IG outperforms other, related methods in nearly every experiment. Our main contributions are as follows:

- We propose Adaptive Path Methods (APMs) a generalization of path methods [29] that consider the model and input when determining the attribution path.
- We introduce Guided IG, an attribution technique that is an instance of an adaptive path method, and describe its theoretical properties.
- We present experimental results that show Guided IG outperforms other attribution methods quantitatively and reduces noise in the final explanations.

2. Related Work

Literature on explanation and attribution methods has grown in the last few years, with a few broad categories of approaches: Black-box methods and methods that perturb the input [20, 9, 8, 19]; methods utilizing back-propagation [29, 22, 4, 2]; methods that visualize intermediate layers [26, 31, 24, 16, 18]; and techniques that combine these different approaches [12, 25, 3]. Our work extends and improves upon Integrated Gradients [29], a popular technique applicable to many different types of models. Accordingly, we focus on perturbation and back-propagation-based methods.

Black-box methods such as [20, 9, 8, 19] are model agnostic, and rely on perturbing or modifying the input instance and observing the resulting changes on the model's output. While [20] uses segmentation-based masking of input, [19] generates random smooth masks to determine salient input segments or regions. [9, 8] apply multiple perturbations such as adding noise, or blurring, and optimize over the model output to learn the mask. All of these methods typically require several iterations/evaluations of the model to identify salient regions (or pixels) for a single input. As BlurIG shows [30], perturbations can introduce artifacts (i.e., information not in the original image), adversely affecting the validity of the output.

Back-propagation methods [29, 22, 4, 2] examine the gradients of the model with respect to an input instance to determine pixel-level attribution. These methods produce a saliency map by weighing the gradient contributions from layers in the network to individual pixels, or entire regions. Our work specifically builds on the path-based Integrated Gradients in [29]. Specifically, our work addresses the issue of noise in pixel attribution in IG, which is highlighted by [25, 12, 27]. While [25] addresses this issue by adding noise-based perturbations to the input, and averaging attributions over the perturbed input samples, [12] aggregates attributions of regions by considering a segmentation of the input. In contrast to these previous methods, our paper addresses this issue of noise by optimizing the path along which the gradients are aggregated.

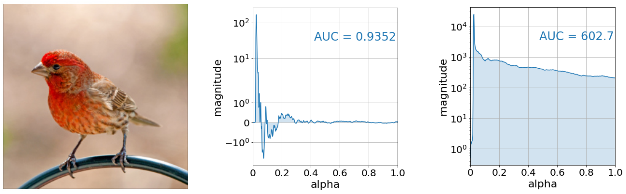
3. High Gradient Impact on IG Attribution

In this section, we provide a summary of the Integrated Gradients method, then describe how highly-correlated gradients can introduce noise to IG's attributions. We also show how model-agnostic paths (e.g., a straight line path) can contribute to this problem.

3.1. Integrated Gradients

For visual models, given an image x, IG calculates attributions per pixel (feature) i by integrating the gradients of the function/model (F) output w.r.t. pixel i as in Eqn. 1.

$$IG_i(x) = \int_{\alpha=0}^{1} \frac{\partial F(\gamma(\alpha))}{\partial \gamma_i(\alpha)} \frac{\partial \gamma_i(\alpha)}{\partial \alpha} d\alpha \qquad (1)$$



(a) Input image (b) Directional Δ (c) Gradient

Figure 3: Differences in magnitude of directional derivative and total gradients. **(a)**: The input image. **(b)**: Signed directional derivative (Δ) magnitude on the straight-line path from the black baseline to the input image, using the input image from (a). The area under the curve is equal to the total attribution. **(c)**: Magnitude of gradients along the straight-line path, using the input image from (a). Even as the magnitude of directional derivatives is close to 0 from $0.3 < \alpha < 1.0$, the magnitude of gradients is high, leading to possible gradient accumulation.

where $\frac{\partial F(x)}{\partial x_i}$ is the gradient of F along the i^{th} feature and $\gamma(\alpha)$ represent images along some integral path ($\alpha \in [0, 1]$). In [29, 25, 12, 17], $\gamma(\alpha)$ is a function that modifies the intensity of pixels from a baseline image $\gamma(\alpha = 0)$ (e.g., a black, white, or random noise image), to that of the input being explained $\gamma(\alpha = 1) = x$.

3.2. Noise Originating from Model-Agnostic Paths

In assessing IG's outputs, one can observe noise in the attributions (e.g., Figure 1). Recent work has investigated this noise and accredited it to the selection of baseline [27] or gradient accumulation in saturated regions [17]. Concurring with [7], we observe that the model's surface is also an important factor.

Looking at the matrix of partial derivatives of the output w.r.t. the input image, we observe that the partial derivatives have a higher by-order-of-magnitude L_2 norm in comparison to the norm of the directional derivative (in the direction of the integration path) matrix (see Figure 3). This implies that the influence of the inputs not contributing to the output may dominate the gradient map at any integral point. One would hope that gradient vectors pointing to different (incorrect) directions will cancel each other out when the whole path is integrated, but that is not always the case as gradient vectors tend to correlate (e.g., see Figure 5). To put it plainly, spurious pixels that don't contribute to the model output end up having non-zero attributions depending on the model geometry (Figure 4).

A straight path, where pixel intensities are uniformly interpolated between a baseline and the input, is susceptible to travel through areas where the gradient norm is high and not pointing towards the integration path (indicated by a low cosine similarity between $\nabla\vec{F}(x)$ and $\vec{dx}$). This issue can be minimized by averaging over multiple straight paths [25, 12] or limiting the impact of noise by splitting the path into multiple segments [17]. However, these approaches sidestep an important issue: attribution methods based on model-agnostic paths will highly depend on surface geometry.

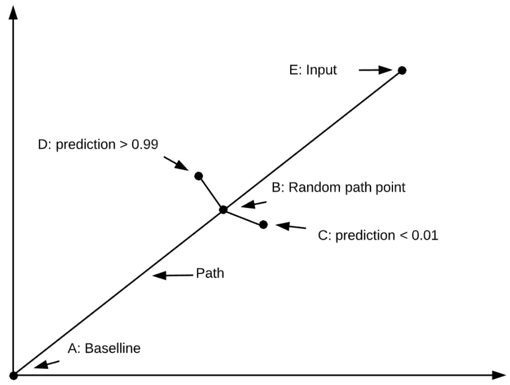


Figure 4: High gradients along the straight-line path from baseline (A) to input (E). At any point (B) of the path, it is possible to find points (C) and (D) that are in very close proximity to (B) and have very low (C) and very high prediction scores (D) respectively. Even though these points arent part of the path, their close proximity to (B) implies the presence of high gradients along the straight line path.

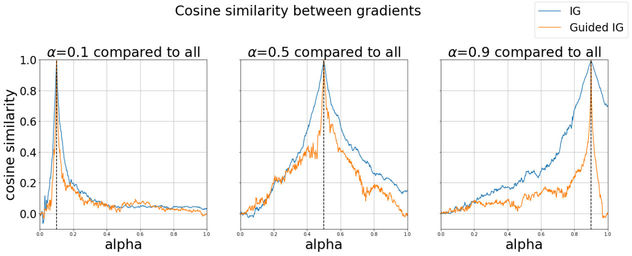


Figure 5: Correlation of gradients along the attribution path. Gradients for the Integrated Gradients (blue) and Guided IG (orange) path were calculated for the image from Figure 1. Each subplot shows cosine similarity between gradients at alpha=[0.1, 0.5, 0.9] to the gradients at all other steps of the integration path for IG and GIG. For each graph, the point chosen is indicated with a dashed vertical line.

4. Adaptive Paths and Guided IG

We introduce *adaptive path methods* (APMs), a generalization of path methods (PMs), to address the limitations of model-agnostic paths. An APM is similar to the definition of path methods (as in [29]), with the additional property that the path can depend on the model function. Adaptive path methods are a superset of path methods, and are defined as follows.

Definition: Let $F : \mathbb{R}^N \rightarrow \mathbb{R}$ be a function of $X = \{x_1, ..., x_N\}$. Let $X^B = \{x_1^B, ..., x_N^B\}$ be the baseline input. Let $X^I = \{x_1^I, ..., x_N^I\}$ be the input that requires explanation. Let path c be parameterized by a path function $\gamma^F = (\gamma_1^F, ..., \gamma_N^F)$ such that $x_i = \gamma_i^F(\alpha)$, where $\alpha \in [0, 1]$ and $\gamma_i(0) = x_i^B$ and $\gamma_i(1) = x_i^I$. The adaptive path method attribution of feature x_i over curve c for any input x^I, is defined as

$$a_i^{\gamma^F}(X^I) = \int_{\alpha=0}^{1} \frac{\partial F(\gamma^F(\alpha))}{\partial \gamma_i^F(\alpha)} \frac{\partial \gamma_i^F(\alpha)}{\partial \alpha} d\alpha. \qquad (2)$$

As with any path method, an APM also satisfies Implementation Invariance, Sensitivity, and Completeness axioms defined in Sundararajan et al. [29]. Below, we expand on these properties for a specific instance of an adaptive path method, Guided IG.

4.1. Desired Characteristics

To alleviate the effect of accumulation of attribution in directions of high gradients unrelated to the input (sec. 3.2), we wish to define a path that avoids those (input) regions causing anomalies in the model behavior. We can call this (ℓ_{noise}), and one way to minimize over this is as follows

$$\gamma^{F*} = \arg \min_{\gamma^F \in \Gamma} \ell_{noise} \qquad (3)$$

$$\ell_{noise} = \sum_{i=1}^{N} \int_{\alpha=0}^{1} \left| \frac{\partial F(\gamma^F(\alpha))}{\partial \gamma_i^F(\alpha)} \frac{\partial \gamma_i^F(\alpha)}{\partial \alpha} \right| d\alpha$$

By minimizing ℓ_{noise} at every feature (pixels $i \cdots N$) we can hopefully avoid high gradient directions. However, before we can define $\gamma^F(\alpha)$ precisely, optimizing the above objective requires knowing the prediction surface of the neural network F at every point in the input space, which is infeasible. So, we propose a greedy approximation method called Guided Integrated Gradients.

4.2. Guided IG

Guided IG is an instance of an adaptive path method. As with IG, the path (c) starts at the baseline (X^B) and ends at the input being explained (X^I). However, instead of moving features (pixel intensities) in a fixed direction (all pixels incremented identically) towards the input, we make a choice at every step. At each step, we find a subset $\mathbb{S}$ of features (pixels) that have the lowest absolute value of the partial derivatives (e.g., the smallest 10%) among those features (pixels) that are not yet equal to the input (image pixel intensity). The next step in the path is determined by moving only those pixels in $\mathbb{S}$ closer to their corresponding intensities in the input image. The path ends when all feature values (intensities of all pixels) match the input. Formally,

- Let $F : R^N \rightarrow R$ be a function of $X = \{x_1, ..., x_N\}$.

The Guided IG integration path, $GIG_{(X^S, X^E, F)}$, is defined based on the starting point (X^S), the ending point (X^E), and the direction vector at every point of the curve:

$$\begin{bmatrix} X^S = X^B \\ X^E = X^I \\ \dfrac{\partial \gamma_i^F(\alpha)}{\partial \alpha} = \begin{cases} x_i^I - x_i^B & \text{, if } i \in \mathbb{S}, \\ 0 & \text{, otherwise.} \end{cases} \end{bmatrix} \tag{4}$$

$\mathbb{S}$ is calculated for every point of curve c, and contains features that have the lowest absolute value of the corresponding partial derivative among the features that have not yet reached the input values. More formally,

$$\mathbb{S} = \{i | \forall j : y_i \leq y_j\} \equiv \arg\min_i(Y) \tag{5}$$

$$y_i = \begin{cases} \left| \dfrac{\partial F(X)}{\partial x_i} \right| & \text{, if } i \in \{j | x_j \neq x_j^I\} \\ \infty & \text{, otherwise.} \end{cases} \tag{6}$$

Guided IG path length. Only features in subset $\mathbb{S}$ are changed at every point of the Guided IG path. Hence in the general case, the L_2 norm of the Guided IG path is greater than the norm of the straight-line path. The Cauchy–Schwarz inequality provides the upper boundary of the path length: $||GIG||_2 \leq \sqrt{N} \cdot ||IG||_2$, where GIG is the Guided IG path and IG is the straight line path. In terms of the L_1 norm, the lengths of the paths are always equal, i.e. $||GIG||_1 = ||IG||_1$. The equality is true because at every point of the Guided IG path individual features of the input are either not changed or changed in the direction of the input.

Efficient approximation. Guided IG can be efficiently approximated using a Riemann sum with the same asymptotic time complexity as the computation of Integrated Gradients. In domains like images, the number of features is high; therefore, it is not practical to select only one feature at every step. Hence, at every step, the approximation algorithm selects a fraction of features (we use 10%) with the lowest absolute gradient values and moves the selected features toward the input. At every step, the algorithm reduces the L_1 distance to the input by the value that is inversely proportional to the number of steps. The higher the number of the steps is, and the lower the fraction is, the closer the approximation is to the true value of Guided IG attribution. We provide our implementation in the Supplement.

Since Guided IG path is computed dynamically, it is not possible to parallelize the computation for a single input. Parallelization is still possible when calculating attribution for multiple independent inputs (batches).

4.3. Bounded Guided IG

The optimal solution path to Equation (3) is unbounded and can deviate infinitely off the baseline-input region. However, it can be advantageous to stay close to the shortest path. First, staying close to the baseline-input region decreases the likelihood of crossing areas that are too out-of-distribution. Second, it can be computationally cheaper to numerically approximate the integral of a shorter path.

To this end, we modify the objective in Eqn. 3 to additionally minimize the accumulated distance to the straight-line path ($\ell_{distance}$). The new objective can then be defined as:

$$\gamma^{F*} = \arg\min_{\gamma^F \in \Gamma} \ell_{noise} + \lambda \ell_{distance} \tag{7}$$
$$\ell_{distance} = \int_{\alpha=0}^{1} \left|\left|\gamma^F(\alpha) - \gamma^{IG}(\alpha)\right|\right| d\alpha,$$

where $\gamma^{IG}(\alpha)$ is the parameterized straight-line path and λ is the coefficient that balances the two components. For very large values of λ (e.g. $\lambda = \infty$), the solution of this objective reduces to the shortest path (same as the IG path). Setting $\lambda = 0$, would give us Eqn. 3, which can be thought of as an *unbounded* version.

One approximation of this objective is limiting the maximum distance that the Guided IG path can deviate from the straight line path at any point[1]. We introduce the concept of *anchors* as a simple way of achieving this. We divide the straight-line path between the baseline X^B and the input X^I into $K+1$ segments and compute Guided IG for each segment separately, effectively forcing the Guided IG path to intersect with the shortest path at K anchor locations. We call this the anchored Guided IG. Accordingly, selecting a higher number of anchors corresponds to optimizing for a higher value of λ in Equation 7, making the results of Guided IG closer to IG. On the other hand, when the number of anchors is zero we have the unbounded algorithm described previously. To put it concretely, for the k^{th} segment, its starting (X_k^S) and ending (X_k^E) point can be set as

- $X_k^S = (X^I - X^B)(k-1)/(K+1) + X^B$, and
- $X_k^E = (X^I - X^B)(k)/(K+1) + X^B$

We then define Guided IG with K anchors as:

$$GIG_{(X^S, X^E, F)}(K) = \sum_{k=1}^{K+1} GIG_{(X_k^S, X_k^E, F)}.$$

Since the integral is a linear operator, summing the integrals of each individual segment is the same as taking the integral of the whole path. For simplicity, we will ignore the other terms and refer to Guided IG with K anchor points as $GIG(K)$ in the rest of the paper.

[1] There are multiple ways of limiting the distance. See https://github.com/pair-code/saliency for the latest implementation.

Sometimes, having zero anchors is more favorable as there can be cases where anchors overlap with the high gradient regions on the shortest path. The more anchors there are, the more likely it is to hit a high gradient region and accumulate noise, while staying closer to the shortest path. We will show the effect of anchors in the Results section in more detail.

4.4. Axiomatic Properties of Guided IG

Guided IG satisfies a subset of desired axioms as IG[29]. We note the ones we satisfy below. As with any path integral in a conservative vector field, Guided IG satisfies the completeness axiom that can be summarized by Eq. (8) (where a_i denotes the attribution per pixel/feature)

$$\sum_{i=1}^{N} a_i = F(X^I) - F(X^B) \tag{8}$$

Since Guided IG satisfies completeness, it also satisfies Sensitivity(a) (see [29] for the proof). Sensitivity(b)[10] is also satisfied because the partial derivative of a function with respect to a dummy variable is always zero at any point of the path. As a result, the value of the integral in Eq. (2) is always zero for any such variable.

In the appendix, we provide a proof that Guided IG is Symmetry-Preserving. Symmetry guarantees that if $F(x, y) = F(y, x)$ for all x and y then both x and y should always be assigned equal attribution. The important remark here is that this does not contradict the uniqueness of the IG method. IG satisfies Symmetry for any function; however, given a function, it may be possible to find other paths that satisfy Symmetry. In practice, we always calculate attributions for a given function, e.g., a neural network model.

Guided IG satisfies the Implementation Invariance axiom; thus, it always produces identical attributions for two functionally equivalent networks. Guided IG preserves the invariance since it only relies on the function gradients and does not depend on the internal structure of the network. There may be other properties that are worth further study such as additivity and uniqueness, please refer to [28].

5. Experiments and Results

We evaluate Guided IG by first observing its behavior in attributing a closed path, then examine its performance using common benchmarks, datasets, and models.

5.1. Attribution of a Closed Path

One challenge in evaluating attribution methods is the lack of ground truth for the attributions themselves [17]. The Completeness axiom guarantees that the sum of all feature attributions for any path is equal to the difference between the function value at the input and the function value at the baseline. Therefore, the sum of all attributions for a closed path is equal to zero. However, Completeness does not define the values of individual feature attributions, only the sum. We can, however, axiomatically define the ground truth attribution for all features as the average attribution values of all possible paths from the baseline to the input. This definition is similar to Shapley values [23] that are also defined in terms of a sum of all possible paths. Using the axiomatic definition of the ground truth attribution, we can now prove that the ground truth attribution for a closed path $A \rightarrow A$ is zero for all features.

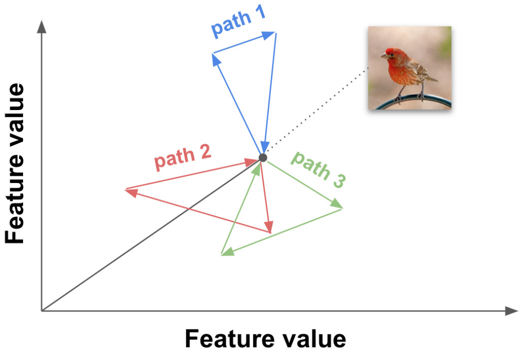


Figure 6: Attribution of closed paths. By calculating the attribution path of the input image with itself via random points, we create an attribution path that in theory should be zero.

Proof. Let P be a set of all possible paths from point A to point A. Let a denote attribution. For any path $p \in P$, there exists a counterpart reverse path p' such that $a_i(p) + a_i(p') = 0$ for all features i. Since every path has a counterpart reverse path cancelling its attributions, the average attribution values of individual features are 0. Q.E.D.

We can build a random path $A \rightarrow B \rightarrow C \rightarrow A$ and consider it as an estimator of ground truth attribution of path $A \rightarrow A$. Using the path integral additive property, we can break the path into sub-segments, i.e., $a(A \rightarrow B \rightarrow C \rightarrow A) = a(A \rightarrow B) + a(B \rightarrow C) + a(C \rightarrow A)$. Now, we can apply a path method on every segment and treat the sum as an estimation of the ground truth. Figure 6 gives an illustration of the idea.

We apply this technique to calculate the attribution of each segment using both IG and Guided IG. We sample 50 random paths on 200 random images from the ImageNet validation dataset, for the total of 10,000 path samples. We calculate the mean of the squared error for individual features and average the error across all images, pixels, and channels. The results in Table 1 show that applying Guided IG results in lower error compared to IG.

	MobileNet	Inception	ResNet
IG	3.817e-07	5.938e-07	6.707e-07
Guided IG	**7.320e-08**	**1.442e-07**	**1.857e-07**

Table 1: Closed path mean squared error for IG and Guided IG.

(AUC) Method	ImageNet MobileNet	 Inception	 ResNet	Open Images ResNet	DR [14]
Edge	0.611	0.610	0.611	0.606	0.643
Gradients	0.614	0.634	0.650	0.505	0.801
IG	0.629	0.655	0.669	0.557	0.833
Blur IG	0.652	0.662	0.663	0.619	0.830
GIG (0)	**0.705**	**0.712**	**0.711**	**0.630**	0.619
GIG (20)	0.691	0.696	0.706	0.624	**0.863***
GradCAM	0.776	0.761	0.755	0.474	0.837
Smoothgrad					
+IG	0.742	0.773	0.781	0.662	0.637
+GIG(0)	0.745	0.776	0.776	0.649	0.632
+GIG(20)	**0.767**	**0.795**	**0.799**	**0.685**	**0.645**
XRAI					
+IG	0.731	0.765	0.762	0.631	0.793
+GIG(0)	**0.838***	**0.829***	**0.821***	0.718	0.630
+GIG(20)	0.808	0.819	0.809	**0.719***	**0.831**

Table 2: We report the mean AUC values for different methods using the black baseline for the methods compared. Higher is better. **bold** indicates highest in each group, * indicates highest overall.

(AUC) Method	ImageNet-Inception black	 b+w	 2-rnd	OpenImages black	 b+w	 2-rnd	DR black	 b+w	 2-rnd
IG	0.655	0.667	0.689	0.557	0.572	0.600	0.833	0.824	0.791
Blur IG	0.662	0.662	0.662	0.619	0.619	**0.619**	0.830	0.830	**0.830**
GIG(0)	0.712	0.738	0.722	0.630	0.625	0.607	0.619	0.544	0.510
GIG(10)	0.702	0.722	0.709	0.626	0.636	0.617	0.850	0.837	0.751
GIG(20)	0.696	0.714	0.704	0.624	0.639	0.617	0.863	0.850	0.792
GIG(40)	0.690	0.706	0.698	0.615	0.634	0.613	**0.865***	**0.851**	0.794
XRAI									
+IG	0.765	0.820	0.843	0.631	0.697	0.747	0.793	0.810	0.793
+GIG(0)	**0.829**	**0.854***	0.843	**0.718**	0.714	0.674	0.630	0.629	0.573
+GIG(20)	0.819	0.852	**0.851**	0.719	0.764	0.744	0.831	0.831	0.788
+GIG(40)	0.815	0.852	0.849	0.709	**0.768***	**0.749**	0.829	0.831	0.795

Table 3: AUC-ROC values for ImageNet Inception model, the Open Images ResNet model, and the DR model, using 3 different choices of baseline - black, black and white, and 2 random (note that BlurIG does not need a baseline); and 4 different anchored versions of GIG (K={0, 10, 20, 40}). Higher is better, values in **bold** are highest in each column, * is highest on dataset.

(SIC AUC) Method	ImageNet MobileNet	 Inception	 ResNet	Open Images ResNet
Edge	0.300	0.371	0.405	0.537
Gradients	0.368	0.431	0.510	0.595
IG	0.402	0.499	0.544	0.694
Blur IG	0.411	0.501	.560	0.659
GIG(0)	0.423	0.516	0.550	0.634
GIG(20)	0.453	0.546	0.584	0.701
GIG(40)	0.453	0.551	0.592	0.734
GradCAM	0.691	0.739	0.763	0.662
XRAI				
+IG	0.671	0.736	0.755	0.843
+GIG(40)	**0.692**	**0.752**	**0.771**	**0.866**

Table 4: SIC AUC results [12] for Guided IG and other methods. All methods use black and white baseline if they require it. Values in **bold** are the highest.

5.2. Quantitative evaluation

Metrics We compare Guided IG attributions with other attribution methods by employing the **AUC-ROC** metric as described in [5]. The metric treats the attributions as classifier prediction scores. Ground truth is provided by human annotators. The sliding threshold determines the proportion of features that are assigned to the "true" class. By changing the threshold, the ROC curve is drawn and the AUC of that curve is calculated.

Additionally, we also use the **Softmax Information Curve (SIC AUC)** metric from [12]. This method directly measures how well the model performs without using human evaluation. It does so by revealing only the regions that are highlighted by the attribution method and measuring the model's softmax score. The key idea is that the attribution method that has better focus on where the model is truly looking should reach the softmax value faster than another one that is less focused on the correct region. Therefore, this metric evaluates the attribution method from the model's perspective without any human involvement.

Methods We compare our method against four baselines: edge detector, vanilla Gradients [24], IG, and Blur IG [30] (with max $\sigma = 35$). We compare both the unbounded version of Guided IG (listed as GIG(0), where the number of anchors is in parentheses), and variants with anchors (e.g., GIG(20)). The edge detector saliency for an individual pixel is calculated as the average absolute difference between the intensity of the pixel and the intensity of its nearest eight adjacent neighbours. We used 200 steps and the black baseline for all methods that required these parameters.

5.2.1 Datasets

We evaluated our approach on two datasets of natural images, and one dataset of medical images (described below), and report results in Table 2.

ImageNet [21] We used images from the standard validation set. We only included images that had ground truth annotation and were predicted as one of the top 5 classes by the corresponding model. We calculated the AUC-ROC metric for the ImageNet dataset using three pre-trained models: Mobilenet_v2 (n=4016), Inception_v2 (n=3965) and Resnet_v2 (n=3838).[2]

Open Images [15] We evaluated on 5000 random images from the validation set of the Open Images dataset. As with ImageNet, we only included images that had ground truth annotation and were predicted as one of the top 5 classes by the corresponding model.

Medical Images [6] We also compare our method on a model trained to predict Diabetic Retinopathy (DR). Specifically, we use the Inception-v4 DR classification model from [14]. We examine the results on a sample of 165 images from the validation set [6].

From Tables 2 and 4, we can see Guided IG outperforms other methods. Also, Smoothgrad [25] and Guided IG are likely reducing the same sources of noise, so we only see a marginal improvement when combining the two methods. However, adding anchors (e.g., GIG(20)), shows a substantial improvement in performance. We also note

[2] All models were downloaded from TensorFlow Hub.

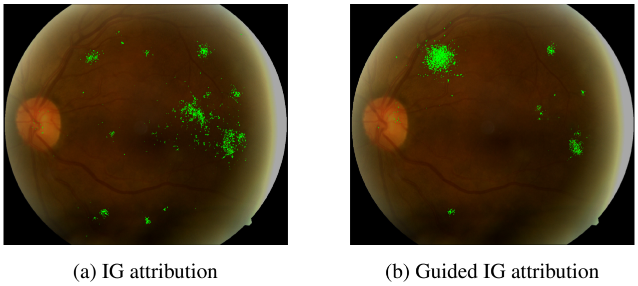

(a) IG attribution (b) Guided IG attribution

Figure 7: Comparison of IG and Guided IG on a retina image [6] used in diagnosing diabetic retinopathy.

that smoothing does not seem to be a good strategy on the DR dataset where sparser attributions may be preferred; this can also be observed with XRAI, but to a lesser extent. The XRAI method aggregates attributions to image segments, and hence when combined with Guided IG, it shows the best performance on most models.

5.2.2 Effect of baseline choice

For path methods, there are different options one can choose for the baseline. Table 3 examines the effect of choosing a black baseline, (average over a) black and a white baseline, and (the average over) 2 random baselines. While a black+white baseline is generally a better choice, with GIG(0), we can see that much of the improvement (over IG) is observed on a single black baseline itself.

5.2.3 Effect of number of anchors

We also report the performance of anchored versions of Guided IG. Table 3 examines all the models on 4 different choice of anchors $K = \{0, 10, 20, 40\}$ (where 0 is simply the unbounded version of Guided IG.) For natural images, it appears that Guided IG without any anchors is a better choice. On the DR dataset, Guided IG seems reliant on anchor points along the straight line path. Overall, Guided IG with a black or black+white baseline and 20 anchor points leads to consistently good performance on the evaluated metrics across all models and datasets.

5.3. Qualitative Results

Figure 8 shows a sampling of results for IG and Guided IG (more in Appendix) with Inception v2 as the model. As can be seen in these figures, Guided IG generally clusters its attributions around the predicted class object with comparatively less noise in other areas of the image. We provide additional qualitative results including failure examples, proof of symmetry, and pseudocode in the Supplement.

6. Discussion

Experimental results from Tables 2 and 3 show that adapting the path to avoid high gradient information allows Guided IG to perform better than other tested methods. From Table 3, we can also observe that Guided IG performs well irrespective of the choice of baseline.

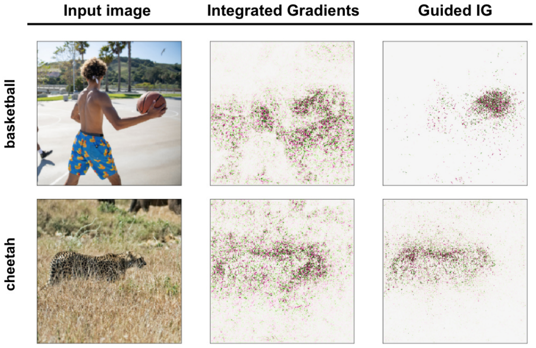


Figure 8: Visual comparison of GIG and IG. See how GIG is more focused around the basketball compared to IG.

In the experimental results, we observe some variation in Guided IG's performance as a function of the number of anchor points. In many cases, these differences are relatively minor; from our experimental results, 20 anchor points may be a reasonable default value to use for this technique. Additionally, while our implementation employs anchor points to prevent paths from becoming too long, there are other ways one could achieve this goal. For example, one could "bound" the path to prevent it from straying too far from the straight line path of IG. Future work could explore alternative methods like this to ensure adaptive paths reduce accumulation of noise, without sacrificing either attribution quality or performance.

Guided IG demonstrates only one instance of an APM; there may exist other APM instances that are better suited for particular tasks, domains, models. Moreover, while we evaluated Guided IG on visual models and datasets, the same or different variants may be better suited for other modalities such as text or graph models.

7. Conclusion

This paper introduces the concept of Adaptive Path Methods (APMs) as an alternative to straight line paths in Integrated Gradients. We motivate APMs by observing how attribution noise can accumulate along a straight line path. APMs form the basis for Guided IG, a technique that builds on IG and adapts the path of integration to avoid introduction of attribution noise along the path, while still optionally minimizing the path length to back-off to a straight path. We demonstrate that Guided IG achieves improved results on common attribution metrics for image models. Opportunities for future work include understanding and investigating Guided IG on other modalities, such as text or graph models.

Acknowledgements We thank Frederick Liu for his technical contributions to the saliency evaluation framework, experiments and feedback, and members of the Google Brain team for feedback on a draft of this work.

References

[1] R. J. AUMANN and L. S. SHAPLEY. *Values of Non-Atomic Games*. Princeton University Press, 1974.

[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. 2015.

[3] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

[4] Alexander Binder, Grégoire Montavon, Sebastian Bach, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. *CoRR*, 2016.

[5] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *CoRR*, abs/1604.03605, 2016.

[6] Jorge Cuadros and George Bresnick. Eyepacs: an adaptable telemedicine system for diabetic retinopathy screening. *Journal of Diabetes Science and Technology*, 3(3):509–516, 2009.

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

[8] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2950–2958, 2019.

[9] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437, 2017.

[10] E. Friedman. Paths and consistency in additive cost sharing. *International Journal of Game Theory*, 32:501–518, 2004.

[11] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 10 2017.

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

[13] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav), 2017.

[14] J. Krause, V. Gulshan, E. Rahimy, P. Karth, K. Widner, G. S. Corrado, L. Peng, and D. R. Webster. Grader Variability and the Importance of Reference Standards for Evaluating Machine Learning Models for Diabetic Retinopathy. *Ophthalmology*, 125(8):1264–1272, 08 2018.

[15] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, and et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):19561981, Mar 2020.

[16] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[17] Vivek Miglani, Narine Kokhlikyan, Bilal Alsallakh, Miguel Martin, and Orion Reblitz-Richardson. Investigating saturation effects in integrated gradients, 2020.

[18] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. 2015.

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

[20] 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. ACM, 2016.

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *arXiv:1409.0575 [cs]*, Sept. 2014. arXiv: 1409.0575.

[22] 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.

[23] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

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

[25] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *ArXiv*, abs/1706.03825, 2017.

[26] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net, 2014.

[27] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Visualizing the impact of feature attribution baselines. *Distill*, 2020. https://distill.pub/2020/attribution-baselines.

[28] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International Conference on Machine Learning*, pages 9269–9278. PMLR, 2020.

[29] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, 2017.

[30] Shawn Xu, Subhashini Venugopalan, and Mukund Sundararajan. Attribution in scale and space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[31] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *Lecture Notes in Computer Science*, page 818833, 2014.