Integrated Decision Gradients: Compute Your Attributions Where the Model Makes Its Decision

Chase Walker[1], **Sumit Kumar Jha**[2], **Kenny Chen**[3], **Rickard Ewetz**[1]

[1]Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL, USA
[2]Knights Foundation School of Computing and Information Sciences, Florida International University, Miami, FL, USA
[3]Lockheed Martin, Orlando, FL, USA
chase.walker@ucf.edu, jha@cs.fiu.edu, kenny.chen@lmco.com, rickard.ewetz@ucf.edu

Abstract

Attribution algorithms are frequently employed to explain the decisions of neural network models. Integrated Gradients (IG) is an influential attribution method due to its strong axiomatic foundation. The algorithm is based on integrating the gradients along a path from a reference image to the input image. Unfortunately, it can be observed that gradients computed from regions where the output logit changes minimally along the path provide poor explanations for the model decision, which is called the *saturation effect* problem. In this paper, we propose an attribution algorithm called integrated decision gradients (IDG). The algorithm focuses on integrating gradients from the region of the path where the model makes its decision, i.e., the portion of the path where the output logit rapidly transitions from zero to its final value. This is practically realized by scaling each gradient by the derivative of the output logit with respect to the path. The algorithm thereby provides a principled solution to the saturation problem. Additionally, we minimize the errors within the Riemann sum approximation of the path integral by utilizing non-uniform subdivisions determined by adaptive sampling. In the evaluation on ImageNet, it is demonstrated that IDG outperforms IG, Left-IG, Guided IG, and adversarial gradient integration both qualitatively and quantitatively using standard insertion and deletion metrics across three common models.

Introduction

The access to internet-scale data and compute power has fueled the success of black box neural network models for applications such as disease detection (Fatima, Pasha et al. 2017), image synthesis (Rombach et al. 2022), and protein folding (Mirdita et al. 2022). The phenomenal performance of these networks comes from the large number of parameters and non-linear interactions among them. The complex and high dimensional dynamics makes it difficult to understand and visualize why a neural network makes a particular decision. To establish trustworthiness in neural network models, noteworthy research efforts have been devoted to interpretability and explainability (Das and Rad 2020). Attribution methods provide model explanation by computing the contribution of each input feature to a model decision. Attribution methods broadly fall into perturbation based (Zeiler and Fergus 2014; Ribeiro, Singh, and Guestrin 2016), backpropagation based (Springenberg et al. 2015; Selvaraju et al. 2017), and gradient based methods (Simonyan, Vedaldi, and Zisserman 2014; Sundararajan, Taly, and Yan 2017). Gradient based methods are promising due to their strong axiomatic foundation, and model-agnostic implementation (Sundararajan, Taly, and Yan 2017).

Gradient based methods compute attribution maps by capturing the gradients at the model inputs with respect to the model outputs (Simonyan, Vedaldi, and Zisserman 2014). However, gradients computed with respect to important input pixels may be zero due to the non-linear activation functions. Integrated Gradients (IG) solved this problem by integrating the gradients along a path from a baseline reference image to the input image (Sundararajan, Taly, and Yan 2017). Unfortunately, it can be observed that gradients from regions of the path where the output logit changes minimally (e.g. is saturated) provide poor explanations for the model decision (Miglani et al. 2020a). This phenomena is called the *saturation effect* problem. Solution templates to solve the saturation problem include: selecting non-straight-line paths (Kapishnikov et al. 2021b; Pan, Li, and Zhu 2021b), path truncation (Miglani et al. 2020a), post processing methods that use thresholding (Kapishnikov et al. 2019a), averaging across blurred inputs (Smilkov et al. 2017), and redefining the model (Jha et al. 2021, 2022). While these methods improve attribution quality, they do not provide a principled solution to the saturation problem.

In this paper, we propose a new path integral attribution method called Integrated Decision Gradients (IDG). We call the portion of the path where the output logit rapidly transitions from zero to its final value the *decision region*. IDG focuses on integrating gradients from the decision region of the path integral. This is realized by scaling each gradient by the derivative of the output logit with respect to the path. The scaling factor rewards gradients in the decision region and penalizes gradients from outside the decision region. The main contributions of this paper are summarized as follows:

- We propose IDG, a new attribution method that provides a principled solution to saturation by satisfying the IG axioms and a new path integral sensitivity axiom.
- We present an adaptive sampling technique to select non-uniform subdivisions for the Riemann approximation of the path integral. This reduces computational errors (and runtime overheads) compared to uniform subdivisions.

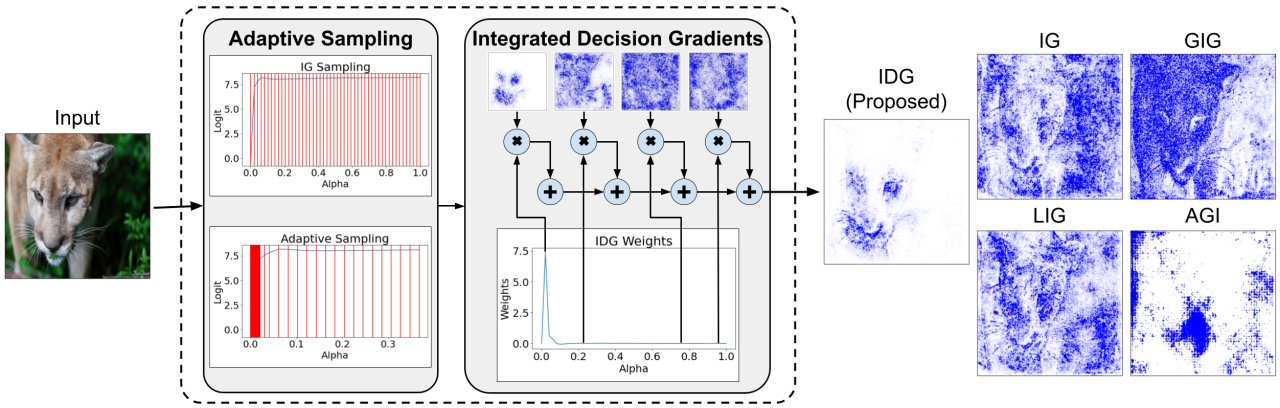

Figure 1: (left) An overview of the adaptive sampling algorithm, and the IDG attribution method. (right) A preliminary visual comparison of IDG with IG (Sundararajan, Taly, and Yan 2017), LIG (Miglani et al. 2020a), GIG (Kapishnikov et al. 2021b), and AGI (Pan, Li, and Zhu 2021b).

- Compared with IG (Sundararajan, Taly, and Yan 2017), Left-IG (LIG) (Miglani et al. 2020a), Guided IG (GIG) (Kapishnikov et al. 2021b), and Adversarial Gradient Integration (AGI) (Pan, Li, and Zhu 2021b), IDG improves in both qualitative and quantitative results.

The remainder of the paper is organized as follows: first, related work is examined, then the IDG attribution method is defined, the adaptive sampling algorithm is proposed, experimental evaluation is presented and discussed, and finally, the paper is concluded.

Related Work

In this section, we first review the limitations of directly using gradients as attributions. Next, we review IG and assess the saturation effect problem within path integrals.

Limitations of Gradients as Attributions

Attributions measure the contribution of each input feature to the model output decision. An attribution method satisfies the axiom of *sensitivity* if a single feature that differs between a baseline and input - which produce different output predictions - is given a non-zero attribution. Additionally, if model output is not affected by changing a variable, then that variable's attribution shall be zero (Sundararajan, Taly, and Yan 2017). Computing the gradient of the inputs with respect to the output logit is a promising method for computing attributions (Simonyan, Vedaldi, and Zisserman 2014). However, the use of non-linear activation functions causes the sensitivity axiom to be violated (Sundararajan, Taly, and Yan 2017), which is shown in Example 1 below.

Example 1. *Consider a function* $F = 1 - ReLU(1 - x)$*, a baseline* $x' = 0$*, and an input* $x = 2$*. For* $x' = 0$*, the function* F *is equal to* 0*, and for* $x = 2$*, the function* F *is equal to* 1*. Since changing* x *from* 0 *to* 2 *affects the output of* F*, the attribution w.r.t. the feature* x *should be non-zero. However,* $\partial F/\partial x = 0$ *at* $x = 2$*, which results in an attribution of* 0 *(Sundararajan, Taly, and Yan 2017).*

Integrated gradients offers a solution to computing attributions that satisfies the sensitivity axiom.

Integrated Gradients

Integrated Gradients computes attributions by integrating gradients on a straight line between a reference image and an input image (Sundararajan, Taly, and Yan 2017). Let F be the function realizing the output logit of interest. IG_i with input image x is mathematically defined using a path-integral (Sundararajan, Taly, and Yan 2017), as follows:

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

where x' is a black baseline image, $\alpha \in [0, 1]$ parameterizes the straight-line path between x' and x, x_i and x_i' represent a single pixel of their respective images, and IG_i is therefore the attribution of pixel i in the input image.

The IG attribution method is illustrated in Figure 2. The top row shows interpolated inputs, the second row shows the corresponding input gradients, the third row visualizes the output logit with respect to the path. The IG attribution map is equal to the sum of the gradients in the second row. The use of a path-integral ensures that gradients from regions of F where $\partial F/\partial x_i$ is non-zero are computed. In Example 1, IG will compute gradients from the region $[0, 1]$, where $\partial F/\partial x = 1$. The resulting attribution w.r.t. x is 2, i.e., the attribution is non-zero and sensitivity is satisfied. Nevertheless, many attributions computed using IG are still noisy due to saturation effects (Miglani et al. 2020a).

Saturation Effects within Path-Integrals

To introduce and understand the *saturation effect* problem within path-integrals, we examine the performance of the IG attribution method in Figure 2. We study the quality of the computed gradients with respect to the decision and saturated regions of the path integral. It can be observed that (i) gradients from the saturation regions are of low quality and (ii) gradients from the decision region are of high quality. The conclusion is rather straight forward to understand. If the model output does not increase while moving $\triangle\alpha$ along the path, it is intuitive that the corresponding gradients are

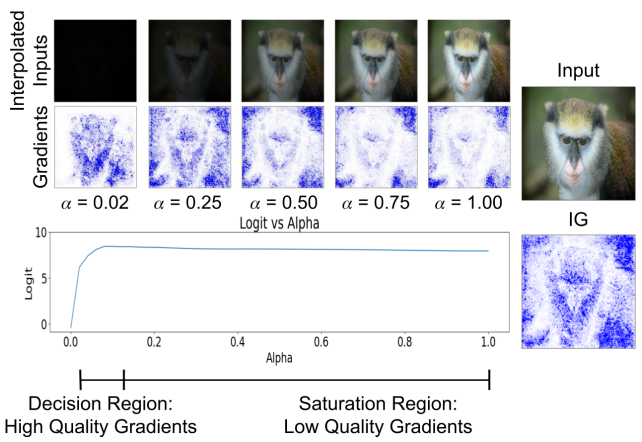

Figure 2: IG and saturation effects within path integrals. The top two rows show interpolated inputs and the corresponding gradients. The IG attribution (shown to the right) is the average of the gradients. The third row shows the logit-α curve that contains the decision and saturation regions which produce higher and lower quality gradients, respectively.

not important to the model decision. Conversely, if the output logit changes rapidly while moving $\triangle\alpha$ along the path, those gradients have a strong impact on the model decision.

This raises the rudimentary question: Is it possible to design a path integral that focuses on computing gradients from the region where the model decision is made and the highly informative gradients are located? It can, for example, be observed in Figure 2 that the gradients computed at $\alpha = 0.02$ alone provide an excellent explanation for the model decision.

Integrated Decision Gradients

In this section, we propose a new attribution method called Integrated Decision Gradients (IDG). We outline the motivation behind IDG, explain the concept of importance factors, and provide IDG's definition and visualization.

Motivation

Path integrals integrate gradients from a reference image to an input target image. A fundamental challenge is to determine the ideal importance of each gradient. Based on the analysis in the previous section, we define a new *sensitivity axiom* for path integrals. Next, we introduce the concept of an importance factor, which is used to construct an attribution algorithm that satisfies the axiom.

Axiom: Sensitivity (path integrals) Let F be the output of a neural network. For every point within a path integral parameterized by α, when $\partial F/\partial\alpha$ is equal to zero, an attribution method satisfies Sensitivity (path integrals) if there is no contribution to the attribution result. If $\partial F/\partial\alpha$ is non-zero, the contribution to the attribution result is non-zero.

None of the existing attribution methods based on path integrals satisfy this axiom (Sundararajan, Taly, and Yan 2017; Miglani et al. 2020a; Kapishnikov et al. 2021b; Pan, Li, and Zhu 2021b). The traditional IG method places an equal weight on all gradients (Sundararajan, Taly, and Yan 2017), even those that occur in the saturation region where $\partial F/\partial\alpha = 0$. The Left-IG attribution attempts to solve this by truncating the path integral after the output logit has reached 90% of its final value (Miglani et al. 2020a). This assigns a weight of zero and one to gradients from the approximate saturation and decision regions respectively, which does not guarantee that the axiom is satisfied. GIG and AGI use non-straight line paths that attempt to avoid integrating gradients from saturated regions (Kapishnikov et al. 2021b; Pan, Li, and Zhu 2021b), which does also not guarantee that the Sensitivity (path integrals) axiom is satisfied.

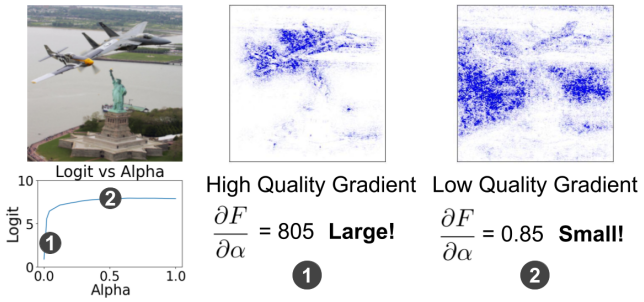

Figure 3: An illustration of the relationship between importance factor magnitude and gradient quality. Higher importance factors are directly related to higher quality gradients.

To satisfy the axiom, we conjecture that the importance of each gradient should be proportional to the impact on the model output, which is conceptually shown in Figure 3. Inspired by this, we define an *importance factor*, as follows:

$$IF(\alpha) = \frac{\partial F(x' + \alpha(x - x'))}{\partial \alpha}, \tag{2}$$

where $IF(\alpha)$ is the importance of the gradient computed at α. Next, we define an attribution method that satisfies the Sensitivity (path integrals) axiom (proof in the Axiomatic Properties section) by scaling each gradient with the importance factor in Eq (2).

Definition of Integrated Decision Gradients

In this subsection, we formally define the IDG attribution algorithm. Given a neural network represented by function $F : R^n \rightarrow [0, 1]$, an input vector x, and given F exists over the range $\alpha \in [0, 1]$, IDG assigns an importance factor to each input feature x_i with respect to the model output, using the following equation:

$$IDG_i(x) = (x_i - x'_i) \times \underbrace{\int_{\alpha=0}^{1} \frac{\partial F(x'_i + \alpha(x_i - x'_i))}{\partial x_i}}_{\text{Traditional IG}} \times \underbrace{\frac{\partial F(x'_i + \alpha(x_i - x'_i))}{\partial \alpha}}_{\text{Importance Factor}} d\alpha. \tag{3}$$

The IDG attribution method is equivalent to IG in Eq (1) but with each gradient scaled with the importance factor in

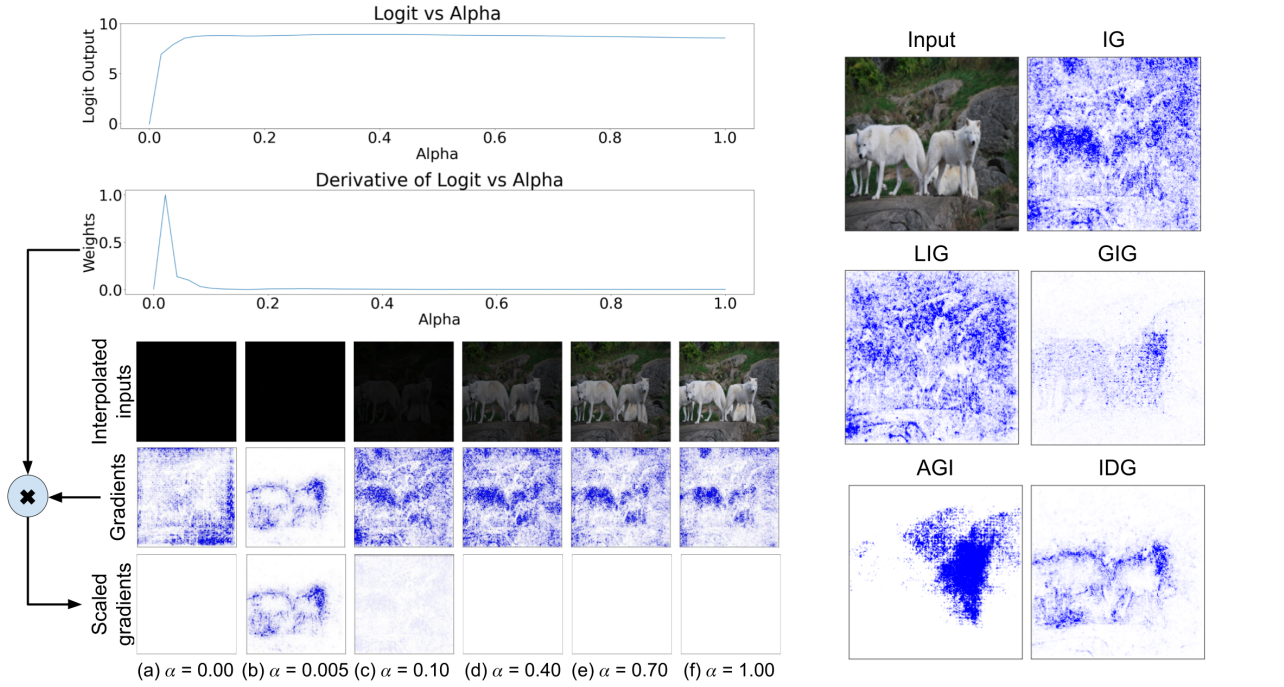

Figure 4: A full visualization of how IDG uses importance factors to eliminate saturation effects. The top row shows the logit-α curve. The next row shows the derivative of the curve, i.e., the importance factors with respect to α. The third row shows the interpolated images, the fourth shows the associated gradients, and the bottom row shows these gradients scaled with the corresponding importance factors. The right side shows the input image, and the attributions computed using IG (Sundararajan, Taly, and Yan 2017), LIG (Miglani et al. 2020a), GIG (Kapishnikov et al. 2021b), AGI (Pan, Li, and Zhu 2021b), and IDG.

Eq (2). The importance factor is equivalent to the derivative of the logit-α curve in the bottom of Figure 2. The importance factors scale-up high quality gradients from the decision region and scale-down low quality gradients from saturated regions, respectively.

The integral is practically computed following IG, using the Riemann sum approximation (Sundararajan, Taly, and Yan 2017), as follows:

$$IDG_i(x) = \frac{(x_i - x_i')}{m} \times \sum_{k=1}^{m} \frac{\partial F(x_i' + \frac{k}{m}(x_i - x_i'))}{\partial x_i} \frac{\partial F}{\partial \alpha} \, d\alpha, \quad (4)$$

where m is the number of steps for approximation. We will further discuss the selection of the step size and its impact on the approximation error in the Adaptive Sampling section.

We illustrate IDG with an example in Figure 4. First, looking at the left side of the figure, the top row shows the logit-α curve associated with the input image. The second row shows the derivative of this curve, i.e., $\partial F/\partial \alpha$ in Eq (2). The third row shows the interpolated inputs for selected alpha values and the fourth row shows the gradients computed by IG for these inputs. The last row visualizes the effect of IDG by scaling the gradients above by the importance factors from the second graph. The importance factors scale up the magnitude of the gradients from the decision region while scaling down the magnitude of the gradients from the saturated regions. In the figure, it can be observed that, in particular, the attributions from $\alpha = 0.005$ are scaled up. On the right of the figure, we show the original image, and the attributions generated by IG, LIG, GIG, AGI, and IDG. The attributions computed using IDG are substantially less noisy than all competitors. We note that GIG has a low amount of noise, but IDG has more focused attributions.

Axiomatic Properties of IDG

In Lundstrom, Huang, and Razaviyayn, it was shown that the axiomatic properties of IG such as Completeness, Sensitivity, Implementation Invariance, and Linearity only hold when assuming a monotonically increasing path and non-decreasing F. We show in the supplementary materials that IDG satisfies the exact same axiomatic properties under the same assumptions. Next, we turn our attention to proving that IDG satisfies Sensitivity (path integrals).

Theorem 1. *IDG is the sole path method to satisfy Sensitivity (path integrals) through the use of the importance factor.*

Proof. Consider the neural network F which is continuous and differentiable over α such that $\partial F/\partial \alpha$ is defined. By definition, an IDG attribution at α along the path is $IDG(\alpha) = \partial F/\partial x \times \partial F/\partial \alpha$. $(x - x')$ is ignored as it is a post-processing factor applied to the attribution which is affected by proper baseline selection.

When $\partial F/\partial \alpha = 0$, the $IDG(\alpha)$ attribution is clearly zero. When $\partial F/\partial \alpha \neq 0$, then $\partial F/\partial x \neq 0$ for at least one feature and the resulting $IDG(\alpha)$ attribution will be non-zero. Therefore it follows, by definition, that IDG satisfies Sensitivity (path integrals). □

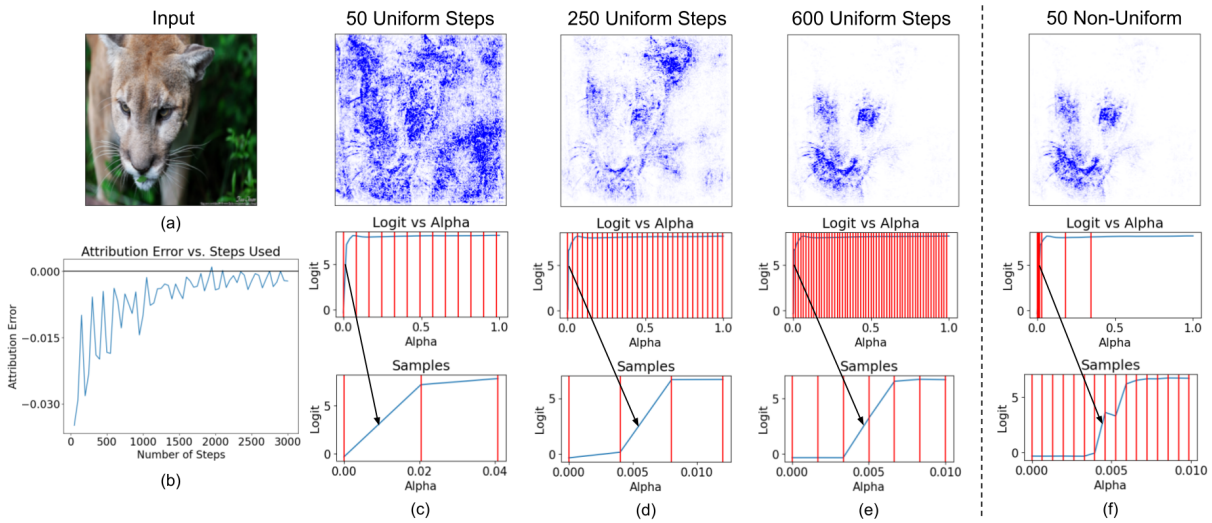

Figure 5: This figure shows the motivation for the adaptive sampling algorithm. The image (a) is the input to the attributions in the figure. The graph (b) demonstrates how the attribution error decreases as step count increases. Columns (c), (d), and (e) of attributions and graphs show the relationship between sample locations and IDG quality as 50, 250, and 600 steps are used respectively. We show that as the number of steps increases, the quality of IDG grows greatly, influencing the adaptive sampling algorithm. Lastly, column (f) shows the equivalent result of column (e) achieved by using adaptive sampling with 50 steps.

Adaptive Sampling Algorithm

In this section, we first analyze the errors within the Riemann sum approximation of the IDG path integral for uniform subdivisions. Next, we propose an adaptive sampling technique to minimize the approximation errors using non-uniform subdivisions. In the supplementary materials, we show that the adaptive sampling only creates improvements with IDG and its impact on regular IG is minor.

Motivation

The errors within the Riemann approximation of the IDG path integral can be calculated, as follows:

$$\epsilon(x, n) = \lim_{m \to \infty} IDG_i(x, m) - IDG_i(x, n), \qquad (5)$$

where $\epsilon(x_i, n)$ is the approximation error for attribution x_i when computing the integral with n uniform subdivisions. n and m are the number of steps used within the Riemann sum approximation in Eq (4).

We analyze the approximation error and the impact on the attributions in Figure 5. The graph (b) shows the average error across all the pixels in the attribution map with respect to the number of used steps n. Since a low step count results in a lack of samples in the decision region, a large number of steps are required for a good approximation. The image (a) is the input for the four columns (c), (d), (e), and (f) of attributions and graphs. The columns show the quality of the attributions with respect to the number of steps and type of subdivision. It is observed from the graphs that taking more samples in the decision region greatly improves IDG attribution quality. Therefore, to obtain high IDG quality without a prohibitive number of steps, we design a new adaptive sampling algorithm - seen in Figure 5 (f) - that uses non-uniform subdivisions concentrated on the decision region.

Adaptive Sampling Methodology

It is desirable to sample the high quality gradients that lie in the decision region to improve the quality of the attained attributions. In Algorithm 1, we show how the adaptive sampling algorithm is used with IDG. Our approach is based on first pre-characterizing the logit-α curve with N uniform subdivisions in lines 3 - 7. Next, M subdivisions are non-uniformly distributed within the N regions based on logit growth and IDG is calculated in lines 8 - 15. Because there are M total samples, line 11 executes $O(N + M)$ times. In practice it is best if $N = M$ (this is shown in the supplementary materials) therefore the algorithm runtime is $O(N)$.

As seen in Figure 5 (e) and (f), combining this adaptive sampling algorithm with IDG creates attributions as strong as IDG with 600 steps while only using 50 steps. Figure 1 provides a high-level overview of this new IDG process. The figure shows that when given an input image and a number of steps, the adaptive sampling algorithm calculates non-uniform subdivisions based on logit growth. These are then used as input for IDG where the gradient at each location is calculated and then weighted, producing the final attribution. In this figure, the IDG sampling graph shows that 31 out of 50 samples are placed in the decision region $\alpha \in [0.0, 0.2]$, where the logit changes from 0 to 7.2.

Experimental Results

In this section, we will evaluate the effectiveness of the proposed method. We perform our experiments in PyTorch using the 2012 validation set of ImageNet (Russakovsky et al. 2015) on NVIDIA A40 GPUs. The attributions computed using Algorithm 1 are called IDG. We compare our method with IG (Sundararajan, Taly, and Yan 2017), LIG (Miglani et al. 2020a), GIG (Kapishnikov et al. 2021b), and AGI (Pan,

Algorithm 1: Computing IDG with Adaptive Sampling

Input: Model F, image x, baseline x', pre-characterization steps N, number of IDG steps M
Output: An attribution map A

$x^0 = x'$
$x^{N-1} = x$
$samples[0] = 0$
// Pre-characterization of logit-α curve
for $i = 0$ to $N - 1$ **do**
  $x^{i+1} = x' + \frac{i}{N} \times (x - x')$
  $samples[i+1] = round(\frac{F(x^{i+1}) - F(x^i)}{F[x^{N-1}] - F[x^0]} \times M)$
end for
// Computation of IDG with non-uniform samples
for $i = 0$ to N **do**
  for $j = 0$ to $samples[i]$ **do**
    $\alpha = \frac{i}{N} + \frac{j}{N \times samples[i]}$
    $x^i = x' + \alpha \times (x - x')$
    $IDG[i] = \frac{\partial F(x^i)}{\partial x} \times \frac{\partial F(x^i)}{\partial \alpha} \times \frac{1}{N \times samples[i]}$
  end for
end for
return $A = mean(IDG)$

Li, and Zhu 2021b). We use Captum for the implementation of IG, whereas LIG, GIG, and AGI are taken from their respective repositories (Kokhlikyan et al. 2020; Miglani et al. 2020b; Kapishnikov et al. 2021a; Pan, Li, and Zhu 2021a). We evaluate the quality of the computed attributions both quantitatively and qualitatively.

In Table 1, we quantitatively evaluate the attributions using standard perturbation testing which measures the importance of the pixels in an attribution via an area under the curve (AUC) score. Four tests are used with three insertion methods and one deletion method from the authors of RISE and XRAI (Petsiuk, Das, and Saenko 2018b; Kapishnikov et al. 2019a) which are described in the next section. The table compares the computed attribution quality for the first 5000 images of the ImageNet dataset such that five images are taken from each of the 1000 classes. The five attribution methods are evaluated with three models trained on ImageNet. We selected ResNet101 (R101), ResNet152 (R152), and ResNeXt (RNXT) as pre-trained models from PyTorch and use the newest ImageNet weights available (V2 for the ResNet models and V1 for ResNeXt) (He et al. 2016; Xie et al. 2017; Paszke et al. 2019).

Qualitatively, we present five selected examples in Figure 6 for the the ResNet101 model generated with the method parameters explained below. We provide a larger, random selection of examples in the supplementary materials.

Inputs are reshaped to (224, 224) for all three presented models. This image processing follows the attribution documentation provided by Captum (Kokhlikyan et al. 2020). The RISE, AIC, and SIC tests use the default parameters found from their respective repositories (Petsiuk, Das, and Saenko 2018a; Kapishnikov et al. 2019b). The IG and LIG attribution methods use 50 steps and a black baseline image. GIG uses the default parameters found at (Kapishnikov et al. 2021a). AGI uses the default parameters found at (Pan, Li, and Zhu 2021a). Lastly, IDG is used with 50 steps and a black baseline image. For all the methods, we use a single baseline only.

Quantitative Evaluation Metrics

The evaluation metrics are built upon the intuition that the highest attribution values should correspond to those features that contribute more to the classification of the target class (Petsiuk, Das, and Saenko 2018b; Kapishnikov et al. 2019a). The process starts from the most important pixels and starts deleting (inserting) them from the original image (to a blurred image for insertion) until only a black (the original) image remains. At each step, the softmax score (or accuracy) is calculated. This gives us an ROC curve from base image to final image, which is used to compute the AUC score for a given attribution. This AUC value is computed for each image and then averaged out over the entire test data selection. For the insertion game, a higher AUC score indicates a better attribution and for the deletion game, a lower AUC score indicates better performance. The two sets of methods presented from Petsiuk, et al and Kapishnikov, et al. take different approaches to the insertion process (Petsiuk, Das, and Saenko 2018b; Kapishnikov et al. 2019a).

In RISE, the insertion (deletion) test which starts (ends) with a Gaussian blurred (black) image (Petsiuk, Das, and Saenko 2018b). In their implementation, pixels are added (deleted) in equal amounts during the test process. Given an NxN image, the test will change the image by N pixels at a time over N steps.

Kapishnikov, et al. present the Accuracy Information Curve (AIC) and Softmax Information Curve (SIC) in their XRAI paper (Kapishnikov et al. 2019a). The AIC test gives each perturbation step a score of 0 or 1 for an incorrect or correct classification and SIC uses softmax as previously discussed. For pixel perturbation, these methods use a schedule that non-linearly removes groups of pixels from the image in increasingly large amounts. The last difference from the RISE insertion test is the blurring method, where the initial image is now blurred in segments, each having its own noise distribution.

Comparison with Previous Work

In Table 1, attribution quality is evaluated using the AIC and SIC insertion metrics and the RISE insertion and deletion metrics. We use an arrow to denote if larger (arrow up) or smaller (arrow down) scores are better. The best score for each model and test type is in bold. Additionally we provide how many times a given method outperforms all other methods in the last row of the table.

It can be observed in Table 1 that IDG achieves a consistent improvement over IG, LIG, GIG, and AGI across all twelve of the tests presented. Comparing IDG to IG and LIG clearly indicates the ability of IDG to mitigate saturation effects in path-based methods while retaining the most important gradient information. When compared to AGI and GIG, the large margin of improvement in the scores shows

Metric	Model	IG (2017)	LIG (2020a)	GIG (2021b)	AGI (2021b)	IDG (ours)
AIC (↑)	R101	0.571	0.589	0.626	0.675	**0.701**
	R152	0.575	0.616	0.646	0.686	**0.718**
	RNXT	0.580	0.611	0.634	0.654	**0.730**
SIC (↑)	R101	0.498	0.522	0.559	0.609	**0.638**
	R152	0.508	0.552	0.582	0.619	**0.659**
	RNXT	0.478	0.518	0.532	0.554	**0.620**
Insertion (↑)	R101	0.498	0.535	0.547	0.561	**0.592**
	R152	0.517	0.562	0.565	0.577	**0.615**
	RNXT	0.276	0.299	0.296	0.307	**0.324**
Deletion (↓)	R101	0.181	0.148	0.155	0.172	**0.108**
	R152	0.202	0.148	0.164	0.190	**0.118**
	RNXT	0.101	0.078	0.082	0.104	**0.068**

Table 1: Comparison of path-based attributions using the AIC, SIC, insertion, and deletion tests

that IDG presents a more complete solution to the saturation problem than these methods. Overall, IDG outperforms all of the path-based attribution methods in the comparison, achieving new state-of-the-art performance.

For qualitative analysis, we compare IG, LIG, GIG, AGI, and IDG in Figure 6. All attributions are computed as previously described. We compare using images of a "Guenon", "Submarine", "Tripod", "African Hunting Dog", and "Warplane" taken from ImageNet (Russakovsky et al. 2015).

Across the five selections, IDG clearly produces attributions with less noise than IG and LIG, further verifying that it solves the saturation problem present in these methods. When compared to GIG, IDG also has superior performance in all of the images. For the Tripod example, even though GIG has relatively low noise, IDG has stronger attributions on the tripod in the foreground and the one in the background as well. Lastly, when comparing to AGI, it can be seen AGI generally has low extraneous noise in the attributions. However, IDG provides tighter, and lower noise attributions on the class subject in the images, therefore the results are better.

The images clearly show that IDG improves visual quality over the other path-based methods. IDG generates attributions with less random noise, showing its ability to solve the saturation problem. Additionally, it shows its ability to outperform the methods which use non-straight-line paths. We provide an additional 50 visual comparisons in the supplementary materials.

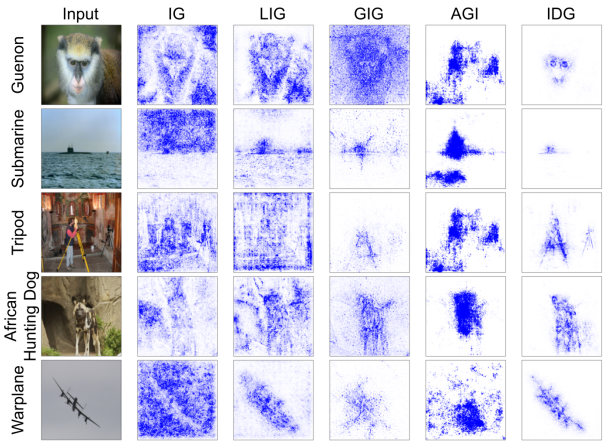

Figure 6: Qualitative comparison of attributions computed using the IG (Sundararajan, Taly, and Yan 2017), LIG (Miglani et al. 2020a), GIG (Kapishnikov et al. 2021b), and AGI (Pan, Li, and Zhu 2021b), and IDG methods. It is seen in the selected examples that IDG solves the saturation problem and outperforms the state-of-the-art path-based attribution methods in visual quality.

Discussion

In this paper, we propose a new attribution method called Integrated Decision Gradients (IDG). The key idea of IDG is to perform the path integral while weighting sampled gradients by their associated logit growth. This amplifies gradients from the decision region, and negates those from the saturation region, solving the saturation issue, and satisfying the Sensitivity (path integrals) axiom. In contrast, traditional IG integrates gradients between the same images while giving all gradients equal weight, saturated or not, causing the majority of saturated gradients to dominate the output. Additionally, we provide evidence that the decision region of the path integral is where the best gradients lie. With this, we present an adaptive sampling algorithm which densely samples the decision region without runtime penalty, improving IDG performance. We show qualitatively and quantitatively that IDG reaches state-of-the-art performance in the path-based attribution field. In our future work, we plan to apply IDG concepts to other attribution methods to further enhance attribution quality. We also plan to employ IDG within practical real-world applications. Both our code and extended technical report including supplementary materials are publicly available via https://github.com/chasewalker26/Integrated-Decision-Gradients.

Acknowledgements

The authors were in part supported by Lockheed Martin Corp., the Florida High Tech Corridor, DARPA Co-operative Agreements #HR00112020002, #HR00112420004, and #FA8750-23-2-0501, and DOE grants #DE-SC0023494 and #DE-SC0024576. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of those providing support for this work.

References

Das, A.; and Rad, P. 2020. Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey. arXiv:2006.11371.

Fatima, M.; Pasha, M.; et al. 2017. Survey of machine learning algorithms for disease diagnostic. *Journal of Intelligent Learning Systems and Applications*, 9(01): 1.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Jha, S.; Ewetz, R.; Velasquez, A.; and Jha, S. 2021. On smoother attributions using neural stochastic differential equations. In *30th International Joint Conference on Artificial Intelligence (IJCAI), 2021*.

Jha, S. K.; Ewetz, R.; Velasquez, A.; Ramanathan, A.; and Jha, S. 2022. Shaping noise for robust attributions in neural stochastic differential equations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 9567–9574.

Kapishnikov, A.; Bolukbasi, T.; Viegas, F.; and Terry, M. 2019a. XRAI: Better Attributions Through Regions. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 4947–4956. Los Alamitos, CA, USA: IEEE Computer Society.

Kapishnikov, A.; Venugopalan, S.; Avci, B.; Wedin, B.; Terry, M.; and Bolukbasi, T. 2019b. AIC code repository. https://github.com/PAIR-code/saliency/tree/master/saliency/metrics. Accessed: 2022-10-15.

Kapishnikov, A.; Venugopalan, S.; Avci, B.; Wedin, B.; Terry, M.; and Bolukbasi, T. 2021a. GIG Code Repository. https://github.com/PAIR-code/saliency/tree/master/saliency/core. Accessed: 2022-10-15.

Kapishnikov, A.; Venugopalan, S.; Avci, B.; Wedin, B.; Terry, M.; and Bolukbasi, T. 2021b. Guided Integrated Gradients: an Adaptive Path Method for Removing Noise. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5048–5056. Los Alamitos, CA, USA: IEEE Computer Society.

Kokhlikyan, N.; Miglani, V.; Martin, M.; Wang, E.; Alsallakh, B.; Reynolds, J.; Melnikov, A.; Kliushkina, N.; Araya, C.; Yan, S.; and Reblitz-Richardson, O. 2020. Captum: A unified and generic model interpretability library for PyTorch. arXiv:2009.07896.

Lundstrom, D. D.; Huang, T.; and Razaviyayn, M. 2022. A Rigorous Study of Integrated Gradients Method and Extensions to Internal Neuron Attributions. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and Sabato, S., eds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 14485–14508. PMLR.

Miglani, V.; Kokhlikyan, N.; Alsallakh, B.; Martin, M.; and Reblitz-Richardson, O. 2020a. Investigating Saturation Effects in Integrated Gradients. arXiv:2010.12697.

Miglani, V.; Kokhlikyan, N.; Alsallakh, B.; Martin, M.; and Reblitz-Richardson, O. L. 2020b. Left-IG Code Repository. https://github.com/vivekmig/captum-1/tree/ExpandedIG. Accessed: 2022-10-01.

Mirdita, M.; Schütze, K.; Moriwaki, Y.; Heo, L.; Ovchinnikov, S.; and Steinegger, M. 2022. ColabFold: making protein folding accessible to all. *Nature Methods*, 1–4.

Pan, D.; Li, X.; and Zhu, D. 2021a. AGI Code Repository. https://github.com/pd90506/AGI. Accessed: 2022-11-01.

Pan, D.; Li, X.; and Zhu, D. 2021b. Explaining Deep Neural Network Models with Adversarial Gradient Integration. In Zhou, Z.-H., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2876–2883. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc.

Petsiuk, V.; Das, A.; and Saenko, K. 2018a. RISE Code Repository. https://github.com/eclique/RISE. Accessed: 2022-11-01.

Petsiuk, V.; Das, A.; and Saenko, K. 2018b. RISE: Randomized Input Sampling for Explanation of Black-box Models. In *Proceedings of the British Machine Vision Conference (BMVC)*.

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

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252.

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

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. arXiv:1312.6034.

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

Springenberg, J. T.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. 2015. Striving for Simplicity: The All Convolutional Net. arXiv:1412.6806.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, 3319–3328. JMLR.org.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1492–1500.

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