Distilled Gradient Aggregation: Purify Features for Input Attribution in the Deep Neural Network

Giyoung Jeon\textsuperscript{1,3}\textsuperscript{*}, Haedong Jeong\textsuperscript{1,2}\textsuperscript{*}, Jaesik Choi\textsuperscript{2,3}
\textsuperscript{1}UNIST \textsuperscript{2}KAIST \textsuperscript{3}INNEJI
giyoung@unist.ac.kr haedong.jeong@kaist.ac.kr jaesik.choi@kaist.ac.kr

Abstract

Measuring the attribution of input features toward the model output is one of the popular post-hoc explanations on the Deep Neural Networks (DNNs). Among various approaches to compute the attribution, the gradient-based methods are widely used to generate attributions, because of its ease of implementation and the model-agnostic characteristic. However, existing gradient integration methods such as Integrated Gradients (IG) suffer from (1) the noisy attributions which cause the unreliability of the explanation, and (2) the selection for the integration path which determines the quality of explanations. FullGrad (FG) is an another approach to construct the reliable attributions by focusing the locality of piece-wise linear network with the bias gradient. Although FG has shown reasonable performance for the given input, as the shortage of the global property, FG is vulnerable to the small perturbation, while IG which includes the exploration over the input space is robust.

In this work, we design a new input attribution method which adopt the strengths of both local and global attributions. In particular, we propose a novel approach to distill input features using weak and extremely positive contributor masks. We aggregate the intermediate local attributions obtained from the distillation sequence to provide reliable attribution. We perform the quantitative evaluation compared to various attribution methods and show that our method outperforms others. We also provide the qualitative result that our method obtains object-aligned and sharp attribution heatmap.

1 Introduction

Deep Neural Networks (DNNs) are increasingly applied to many fields in human-life such as self-driving, medical predictions and time-series forecasts. Along with these improvements, the recent models get bigger and more complicated that humans cannot investigate and understand the internal decision mechanism of them. Identifying and analyzing the reasons for the model predictions are important because the malfunction or the groundless decision of the model can cause the critical problems. As an effort to provide the evidences on the decisions, the input attribution has been well-studied, especially in visual tasks. Input attribution method aims to measure how much each input feature contributes to the model prediction. Because the output of this method takes the form of heatmap to provide relative importance among input features, it helps to locate the appearance of human-level semantics in the input. Previous work also applies the attribution to debug the model by wiping out Clever Hans of DNNs \cite{Lapuschkin2019}. However, obtaining the trustful input attribution is still challenging because (1) the highly nonlinear structure of modern DNNs makes it difficult to correctly track the relationship between the input and the output, and (2) quantifying the reliability of attribution methods is non-trivial because the ground-truth is not available.

\textsuperscript{*}Equal Contribution
Gradient-based input attribution is one of the main techniques to derive the relationship between the model decision and the input features. The partial derivative of the output with respect to the input provides the measure of sensitivity, which is easily computed in DNNs. Integrated Gradients (IG) \cite{Sundararajan et al., 2017} is a commonly used gradient-based method, which provides the axiomatic properties to support the reliability of attributions. However, IG inheres the problem of noisy attribution, which originates from the gradient integrating path, and several variants of IG have been proposed to alleviate this issue \cite{Smilkov et al., 2017, Kapishnikov et al., 2019, 2021, Pan et al., 2021}. FullGrad (FG) \cite{Srinivas and Fleuret, 2019} also raises the counter-intuitive behaviors of IG. FG avoids this problem by considering only the local gradients instead of the path integration and proposes to use the bias gradient. But FG is vulnerable to the small perturbation in the inputs due to its locality.

In this work, we provide the analysis on the weakness of (1) FG method which is unavoidable if it considers only single anchor point, and (2) IG method which the continuous path based gradient integration may fail to quantify the intuitive attribution. To complement the shortcomings of the two methods, we propose to aggregate the attribution from the multiple anchors. For the selection of anchor points, we devise an algorithm to sequentially distill the irrelevant features to generate the reliable attribution. The main contributions of our work are,

- Propose a novel feature distillation algorithm based on the intermediate local attribution to generate the sequence of meaningful anchor points.
- Devise Distilled Gradient Aggregation (DGA), an attribution method by aggregating the intermediate local attributions from the distilled input sequence for the reliable attributions.
- Qualitative and quantitative evaluations to validate the proposed method outperforms existing gradient-based attribution methods.

## 2 Related Work

Input attribution is one of the post-hoc explanation methods, which aims to identify the influence of each input feature to the model output on the trained model. There exists a considerable variety of techniques to derive the input attribution. With the property that the feature map obtained by the convolutional layers includes the spatial information, Class Activation Mapping (CAM) methods compute the attribution by the weighted sum of the feature maps \cite{Zhou et al., 2016, Selvaraju et al., 2017}. Layer-wise Relevance Propagation (LRP) method propagates the model output backward to the input \cite{Bach et al., 2015, Nam et al., 2020}. LRP extends the Taylor decomposition to the DNNs and distributes the relevance in layer-wise sense.
There are another approach by measuring the behavior of the model by perturbing the input features. Optimizing the input by gradient ascent gives an example which would maximally activates the target neuron [Erhan et al., 2010; Nguyen et al., 2016; Olah et al., 2018]. Instead of maximizing the target neuron activation, Extremal Perturbation optimizes the mask which removes or reveals the part of the input to localize the attributed part of input [Fong et al., 2019]. It is extended by using Integrated Gradients [Sundararajan et al., 2017] for the optimization to make the optimization more stable [Qi et al., 2020]. By collecting the pair of partially removed inputs and corresponding model outputs, training a linear model to resembles the mapping would give the feature importance in terms of linear weights [Ribeiro et al., 2016]. Rather than training a new model, Randomized Input Sampling for Explanation (RISE) computes the attribution by aggregating multiple randomly masked inputs, weighted by the model outputs [Petsiuk et al., 2018].

Based on Aumann-Shapley value [Aumann and Shapley, 2015], which is one solution of the fair distribution in the cooperative game theory, Integrated Gradients (IG) has been proposed [Sundararajan et al., 2017]. IG is equipped with axiomatic properties which are desirable for the attribution methods. IG is computed by integrating gradients over the straight path from the predefined baseline to the input. As the attribution is corrupted by the noisy information along the path, alternatives for the different paths have been proposed [Smilkov et al., 2017; Kapishnikov et al., 2019, 2021]. FullGrad (FG) [Srinivas and Fleuret, 2019], which utilizes the bias-gradient, is proposed to suppress the counter-intuitive behavior of IG which is caused by the weak-dependency between the local linear regions.

3 Distilled Gradient Aggregation Method

In this section, we propose our gradient aggregation method, Distilled Gradient Aggregation (DGA). We first provide an example and analysis about the inconsistency observed by FullGrad, which uses a single anchor point to compute attributions. We also provide the counter-intuitive behavior of IG caused by the continuous gradient integration path. To complement both shortcomings, we propose an aggregation method, which ensembles the local attribution from the sequence of inputs. To reduce computational cost and reinforce the features that are in charge of the model decision, we suggest a sequential feature distillation algorithm, which distills irrelevant features from the input.

3.1 Analyzing FG and IG on Simple Models

Assume we have the input vector $x \in \mathbb{R}^2$ and a simple neural network $f$ equipped with partial linear activation (e.g., ReLU). This network $f$ can be regarded as the combination of piece-wise linear functions [Montufar et al., 2014]. Each piece-wise linear function is only defined and feasible in
Figure 3: (a) A contour of logit values for trained \( f \). (b) The linear regions which the trained network comprises. Each colored linear region corresponds to each piece-wise linear function. (c) Selected two linear regions (A and B) and the zero baseline. The dotted lines indicate the perturbations for \( x_1 \) axis in the same linear region. (d) Attribution of IG and FG for each linear region. We identify that for linear region A (include the baseline), the global attribution (IG) is same as the local ones (FG). However, for linear region B, the global and local attribution has different attributions for input samples.

The implementation details are described in the Appendix J.

Vulnerability of FullGrad

FullGrad suggests that the attribution should be same inside the same linear region \( R^{(k)} \), and this reduces the dependency between the attribution and the input \( x \). This property is introduced as weak dependency. However, such weak dependency derives the attribution to be vulnerable to the model perturbation. For example, let we have two inputs, \( x \) and \( x' = x + \epsilon \), where \( \epsilon \) be any small enough random perturbation. If we find any \( x \) and \( x' \), such that the model output is same, \( f(x) = f(x') \), but the region is different, then the attribution on each input should be different. This can be visualized by the simple experiment by generating noise perturbed image \( x + \epsilon \) and measuring the attribution, where \( \epsilon \sim \mathcal{N}(0, \sigma I) \). Figure 2 shows an example where the FullGrad generates inconsistent attribution along with the simple Gaussian noise is added.

Counter-intuitive behavior of IG

To visualize the counter-intuitive behavior of IG, we select two linear regions (A, B) in Figure 3c and calculate the attribution in each region. In particular, we select a sequence of data from \( a \) (white dot) to \( b \) (green dot), which is only shifted in \( x_1 \) dimension. Figure 4a illustrates corresponding attribution for two selected linear regions. We observe that only attribution of \( x_1 \) changes for the sequence of region A in both IG and FG methods. However, for the sequence of region B, the IG attribution of both \( x_1 \) and \( x_2 \) changes at the same time, while FG attribution shows attribution change only in \( x_1 \). We conjecture this counter-intuitive behavior of IG is caused by the baseline selection. With the zero-baseline (\( x = 0 \)), the integration paths of samples in region A traverse only a single region to compute IG. On the other hand, paths of samples in region B traverse through multiple regions. When traversing multiple regions, the counter-intuitive behavior is induced due to passing the undesirable linear regions. From this observation, we can identify that the selection of baseline determines (1) which linear regions are traversed by the
Figure 4: Distillation sequence $\tilde{X}$ with WC mask and EPC mask for the target class French horn in the pre-trained VGG-16. The bottom row of (a) and (b) indicates the local attribution $\phi_{UFG}(\tilde{x}(n))$ for each colored box of first row.

Based on these insights, we raise the weakness of the local and global attribution methods: (1) vulnerability and (2) counter-intuitive behavior. To alleviate these issues, we desire an attribution method to combine the strengths of each type of attribution methods. To resolve such desire step by step, we start from the local attribution which is free from the baseline selection. Then, the remained problem is how to select the meaningful linear regions to generate the reliable attributions.

3.2 Sequential Feature Distillation

For the strategy to select linear regions, RISE [Petsiuk et al., 2018] suggests the random perturbation-based approach. RISE explores the multiple linear regions with randomly ablated masks to measure the importance of each ablated features. However, the randomized ablation includes the stochastic process which requires expensive computational cost to achieve reliable attributions. We are also inspired from the adaptive selection for the perturbed inputs in Guided IG (GIG) [Kapishnikov et al., 2021], which can improve the final attribution. Thus, we hypothesize that the adaptive exploration of linear regions based on the intermediate local attribution can reduce the cost of randomized exploration and follow the spirit of the adaptive exploration. Finally, we propose the sequential feature distillation algorithm to obtain a sequence of ablated inputs, the sequence $\tilde{X} = [\tilde{x}(0), \tilde{x}(1), \tilde{x}(2), \ldots]$, where the irrelevant features are distilled. Motivated by previous work that masking out the irrelevant features using IG [Fong et al., 2019; Qi et al., 2020], we propose to distill the impurities by using the intermediate local attribution obtained along the sequence.

For reliable local attribution, FG proposes the bias gradient with post-processing $\Psi(\cdot)$ which includes the normalization and upsampling. As $\Psi(\cdot)$ suggested by FG is usually over-estimated by the bias gradient in deeper layers [Grabska-Barwinska et al., 2021], we redefine $\Psi(\cdot)$ as uniformly distributing function for the bias gradient to alleviate the over-estimation problem.

$$\Psi_{UFG}(\cdot) = \frac{\mathbf{1}^T \frac{\dim(x)}{\dim(x)} \mathbf{1}}{\dim(x)}$$

where $\mathbf{1}^d$ denotes a $d$-dimensional all-ones vector. We call FG with redefined post-processing $\Psi(\cdot)$ as Uniform FullGrad (UFG), $\phi_{UFG}(\cdot)$. Then we use UFG as the intermediate local attribution method throughout the remained paper. We note that different local attribution, such as Grad*Input [Shrikumar et al., 2016], can be used in our method. We provide the comparison of selecting different local attribution in Appendix G.

To distill off the uninformative features, we build a mask to zero out the features with low magnitude of local attribution. In the sequence $\tilde{X}$, the relation between $n$ and $n+1$-th ablated input is formalized as,

$$\tilde{x}(n + 1) = \mathcal{M}(\tilde{x}(n)) \odot \tilde{x}(0)$$

where $\tilde{x}(0) = x$ and $\mathcal{M}(\cdot)$ is a mask extractor. We define this mask extractor as the Weak Contributor (WC) mask, $\mathcal{M}^{WC}$. The level of the WC mask increases along the distillation sequence $\tilde{X}$ with the
Algorithm 1 Distilled Gradient Aggregation

Input: Model \( f \), Input \( x \)

Parameter: \# of steps \( N \), EPC threshold \( q \), Negative scale \( \beta \)

Output: Attribution \( \phi^{DGA}(x) \)

1: Let \( \tilde{x}(0) = x \), \( \Phi = \emptyset \)
2: for \( n \) in \( \{0 \ldots N\} \) do
3: \( \Phi = \Phi \cup \{\phi^{UFG}(\tilde{x}(n))\} \)
4: \( M = \frac{n}{N} M^{WC}(\tilde{x}(n), n; N) + (1 - \frac{n}{N}) M^{EPC}(\tilde{x}(n); q) \)
5: \( \tilde{x}(n+1) = \tilde{x}(0) \odot M \)
6: end for
7: \( \phi^{DGA}(x) = \sum_{\phi \in \Phi} \text{ReLU}(\phi) \)
8: return \( \phi^{DGA}(x) \)

We define an additional mask to reduce the saturation by filtering out features with extremely strong attribution. We call this mask as Extreme Positive Contributor (EPC) mask \( M^{EPC}(\cdot) \) and it is formulated as,

\[
S_j^{EPC}(x) = \left\{ k \mid \phi_k^{UFG}(x) \leq \phi_j^{UFG}(x) \right\}
\]

\[
M_j^{EPC}(x; q) = \begin{cases} 
1 & \text{if } \frac{\|S_j^{EPC}(x)\|}{\text{dim}(x)} \leq q \\
0 & \text{otherwise}
\end{cases}
\]

(9)
(10)

where \( q \) is EPC threshold to control the ratio of ablation. Finally, we combine two masks for our distillation algorithm with relative weights w.r.t. the current step \( n \) as,
Table 1: Comparison of various attribution methods with LeRF and MoRF on three models.

<table>
<thead>
<tr>
<th></th>
<th>G*I</th>
<th>GBP</th>
<th>IG</th>
<th>FG</th>
<th>GIG</th>
<th>DGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeRF (↑ is better)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VGG-16</td>
<td>0.078</td>
<td>0.113</td>
<td>0.096</td>
<td>0.415</td>
<td>0.110</td>
<td><strong>0.434</strong></td>
</tr>
<tr>
<td>ResNet-18</td>
<td>0.114</td>
<td>0.145</td>
<td>0.158</td>
<td>0.448</td>
<td>0.185</td>
<td><strong>0.533</strong></td>
</tr>
<tr>
<td>Inception-V3</td>
<td>0.171</td>
<td>0.162</td>
<td>0.243</td>
<td>0.558</td>
<td>0.255</td>
<td><strong>0.691</strong></td>
</tr>
<tr>
<td>MoRF (↓ is better)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VGG-16</td>
<td>0.045</td>
<td>0.094</td>
<td>0.036</td>
<td>0.110</td>
<td>0.029</td>
<td><strong>0.023</strong></td>
</tr>
<tr>
<td>ResNet-18</td>
<td>0.050</td>
<td>0.124</td>
<td>0.038</td>
<td>0.131</td>
<td>0.029</td>
<td><strong>0.019</strong></td>
</tr>
<tr>
<td>Inception-V3</td>
<td>0.105</td>
<td>0.145</td>
<td>0.066</td>
<td>0.175</td>
<td>0.061</td>
<td><strong>0.041</strong></td>
</tr>
</tbody>
</table>

\[
\tilde{x}(n + 1) = \left( \frac{n}{N} \mathcal{M}^{WC}(\tilde{x}(n), n; N) + \left(1 - \frac{n}{N}\right)\mathcal{M}^{EPC}(\tilde{x}(n); q) \right) \odot \tilde{x}(0)
\]  

We note that in early distillation step, EPC mask takes high weight to reduce saturation at too highly attributed feature. In the late stage, WC mask gains high weight to remain the relevant features. The distillation sequence \( \tilde{x} \) with WC and EPC masks is shown in Figure 4 (b). We identify that using both WC and EPC masks removes irrelevant features (red box) and iteratively assign the attribution to relevant features (blue box).

### 3.3 Attribution Aggregation

With \( N \) distillation steps, we obtain \( N \) local attributions. The remaining question is how to aggregate these local attributions to acquire the final attribution. Likewise previous studies [Selvaraju et al., 2017; Kindermans et al., 2018; Bach et al., 2015], we desire to take the positive contribution from each local attribution. Thus, we take the ReLU before the aggregation.

\[
\phi^{DGA}(x) = \frac{1}{N} \sum_{n=1}^{N} \text{ReLU}(\phi^{UFG}(\tilde{x}(n)));
\]

Finally, we call the unification of preceding modules as Distilled Gradient Aggregation (DGA) method. DGA method consists of the distillation algorithm with WC and EPC masks to generate the ablated inputs to achieve the local attribution, and the aggregation process considering features with the positive attributions. The illustration of the overall structure is depicted in Figure 4 and pseudo code is provided in Algorithm 1. To analyze the influence of each module (WC/EPC mask and ReLU), we perform the ablation study for each module and the results are available in Appendix A and H.

### 4 Experimental Results

In this section, we validate the effectiveness of DGA by both quantitative and qualitative comparison. We note that evaluating the attribution method is still challenging due to the absence of the ground truth. With this difficulty, we verify if the proposed method appropriately reflects the behavior of the model prediction by providing the quantitative comparison using three metrics: (1) pixel perturbation [Samek et al., 2016], (2) sensitivity-\( n \) [Ancona et al., 2018] and (3) RemOve-And-Retrain (ROAR) [Hooker et al., 2019]. Then we also provide the qualitative comparison among different attribution methods. In the following experiments, we set the hyperparameters for DGA as \( N = 30 \) and \( q = 0.9 \) with simple grid search. The details for the hyperparameter exploration is available in Appendix K. We select various gradient-based attribution methods as the baselines: Gradient*Input (G*I), Guided BackPropagation (GBP), Integrated Gradients (IG), FullGrad (FG), and GuidedIG (GIG).

#### 4.1 Pixel Perturbation

Pixel perturbation is widely used method to benchmark the attribution methods if they correctly capture the relevance between the input features and the model output. To quantify the relevance
Table 2: Comparison of various attribution methods with sensitivity-$n$ on ResNet-18

<table>
<thead>
<tr>
<th>q%</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
</tr>
</thead>
<tbody>
<tr>
<td>G*I</td>
<td>-0.006</td>
<td>-0.009</td>
<td>-0.009</td>
<td>-0.013</td>
<td>-0.007</td>
<td>-0.014</td>
<td>-0.016</td>
<td>-0.022</td>
<td>-0.037</td>
</tr>
<tr>
<td>GBP</td>
<td>0.022</td>
<td>0.032</td>
<td>0.025</td>
<td>0.024</td>
<td>0.022</td>
<td>0.027</td>
<td>0.024</td>
<td>0.024</td>
<td>0.020</td>
</tr>
<tr>
<td>IG</td>
<td>0.013</td>
<td>0.022</td>
<td>0.020</td>
<td>0.027</td>
<td>0.032</td>
<td>0.037</td>
<td>0.039</td>
<td>0.047</td>
<td>0.67</td>
</tr>
<tr>
<td>GIG</td>
<td>0.006</td>
<td>0.007</td>
<td>0.004</td>
<td>0.002</td>
<td>0.003</td>
<td>0.002</td>
<td>0.002</td>
<td>0.002</td>
<td>-0.001</td>
</tr>
<tr>
<td>FG</td>
<td>-0.045</td>
<td>-0.017</td>
<td>-0.005</td>
<td>0.001</td>
<td>0.003</td>
<td>0.003</td>
<td>0.008</td>
<td>0.008</td>
<td>0.004</td>
</tr>
<tr>
<td>DGA</td>
<td>0.095</td>
<td>0.096</td>
<td>0.101</td>
<td>0.098</td>
<td>0.095</td>
<td>0.089</td>
<td>0.083</td>
<td>0.080</td>
<td>0.079</td>
</tr>
</tbody>
</table>

between the input features and the model output, pixel perturbation method removes the pixel values in order of relevance obtained by attribution methods. Then it measures the AUC of the softmax output change for the target class with the perturbation. There are two orders of removal, Most-Relevant-First (MoRF) to remove the pixels with top $k\%$ relevance and Least-Relevant-First (LeRF) to remove the pixels with bottom $k\%$ relevance. If input feature is actually highly related to the model prediction, the softmax output should decrease steeply when it is removed. Thus, MoRF is better if it is lower. In the same manner, higher LeRF is better.

We use 50k images of the validation set provided by ImageNet [Russakovsky et al., 2015]. We use three publicly available pre-trained models: VGG-16 [Simonyan and Zisserman, 2015], Inception-v3 [Szegedy et al., 2016], ResNet-18 [He et al., 2016]. Table 1 indicates MoRF and LeRF results for the various attribution methods and model architectures. We identify that DGA shows the best performance in both MoRF and LeRF measure on entire architectures. We also provide additional comparison with different attribution methods (e.g., LRP) in Appendix I.

4.2 Sensitivity-$n$

Sensitivity-$n$ [Ancona et al., 2018] has been suggested to quantify the generalization of previously suggested properties, Completeness [Sundararajan et al., 2017] and Summation to Delta [Shrikumar et al., 2017]. In general, since not all deep learning models can satisfy Sensitivity-$n$, we use empirical approximations to determine whether there is an algorithmic bias in the process of calculating attribution. To measure how much the attribution method satisfies this properties, the metric quantifies the correlation between the sum of the attributions ($\sum_{i \in S} \phi_i(x)$) for any subset of features ($S$), and the change of the model output when the input subset is ablated ($x_{S^c}$). The metric can be represented as,

$$\text{corr}_S \left[ \sum_{i \in S} \phi_i(x), f(x) - f(x_{S^c}) \right] \approx \text{corr}_{M_q} \left[ \langle \Psi(M_q), \phi(x) \rangle, f(x) - f((1 - \Psi(M_q)) \odot x) \right]$$

where the subset $S$ is uniform randomly sampled with its cardinality $|S| = n$ and $S^c$ denotes its complement. The higher correlation value indicates that the attribution method empirically satisfies the sensitivity-$n$.

For computational efficiency, instead of ablating the entire individual input features ($n$), we perform patch-wise ablation to compute sensitivity-$n$ with randomly sampled binary masks $M_q \in \{0, 1\}^{14 \times 14}$ (100 masks for each input) where the portion of 1 in each mask is constrained be $q$. After the mask selection, we multiply the upsampled mask ($\Psi(M_q)$) to the attribution and the input to compute the correlation. The Table 2 shows the correlation values for various percentage of selection ($q$) over each attribution method for ResNet-18. We identify that DGA has the highest correlation values in entire cases.

4.3 RemOve-And-Retrain (ROAR)

ROAR is another metric to evaluate how well the attribution method captures the relevance of feature in the perspective of the model training. ROAR is performed by measuring the performance of the re-trained model with inputs modified according to relative ordering of the attribution. Each input in the dataset is modified by removing pixels with top $k\%$ attribution and replacing them with the average pixel value of the input. We perform ROAR experiment with simple CNN (6 Conv + 3
Figure 5: Comparison of ROAR experiment results on CIFAR-10 dataset among various attribution methods. The test accuracy for corresponding the percentage of removal.

Linear) trained on 50k images of training set provided by CIFAR-10 dataset [Krizhevsky, 2009] using Adam optimizer with learning rate 3e-4 and 100 epochs. After training, the performance of the model is quantified using the standard test dataset with 10k images. We note that the attribution method captures more relevant features if the test accuracy is lower. We provide the average performance over 10 trials for each attribution method, where the parameters are random initialized at each trial and fixed between attribution methods. Figure 5 shows the test accuracy measure in the ROAR experiment for each attribution method. The result indicates that the model trained on the modified dataset with DGA steeply decreases the test accuracy even with 10% removed. We conclude that DGA can extract the features which are relevant to training procedure in DNNs.

4.4 Qualitative comparison

We qualitatively compare the various attribution methods by visualizing the attribution heatmap and top 10% most relevant features at the same time. In Figure 6, we provide the result of randomly selected images from the validation set of ImageNet with the pre-trained VGG-16. We can identify that the attributions are more aligned with the object comparing other methods. For example, in the right-top row, DGA focuses the person who grabs a paddle while almost methods distribute the relevant pixels to sky and ocean. Although FG concentrates to the person, the relevant patch has less sharp than patch of DGA. We provide more examples and results for different models in Appendix B-D.

5 Discussion

In this paper, we propose a novel gradient-based attribution method, Distilled Gradient Aggregation (DGA). We provide the vulnerability of FG against the input perturbation and the counter-intuitive behavior of IG. To complement the weakness of both methods, we propose the gradient aggregation method along the distillation sequence that generates the inputs which the impurities are distilled. Our method obtains high quality attributions with its sharpness and object-alignment, and we verify the method through pixel perturbation, sensitivity-\(n\), and ROAR evaluation metrics. We believe that our DGA method can be broadly applied to explain a decision of various DNNs.

Broader Impact Transparency of deep models is a matter of the highest priority for the application of such models in the real world, e.g., medical diagnosis [Caruana et al., 2015] and autonomous driving [Yurtsever et al., 2020]. We believe that providing the evidence which is well-aligned with the model decision would help the users of such applications to place great trust and the developers to improve or fix the model for better performance. Discovering unintended biases in the model is another issue [Stock and Cisse, 2018]. Such biases may occur from the dataset [Kim et al., 2018] or the model itself. Identifying the root cause and removing such biases would be another expected future work, beyond the explanation on the input features.

Limitation Although our method has empirically outperformed in qualitative comparison and various quantitative experiments compared to previous work, the notion of better input attribution method is still vague. In this work, we adaptively find the sequence of inputs by using local attribution, but there would exist better justification of the sequence or the set of inputs that are essential clues for identifying the core features in the input.
Figure 6: Qualitative comparison among various attribution methods for VGG-16 in the validation dataset of ImageNet. Upper rows describe the heatmaps obtained by each methods and lower rows shows top 10% most relevant input features. DGA generates sharp and object-oriented attribution heatmap in the almost examples. See more examples in Appendix B-D.

Acknowledgement

This work was conducted by Center for Applied Research in Artificial Intelligence (CARAI) grant funded by Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD) (UD190031RD) and partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2022-0-00984, Artificial Intelligence, Explainability, Personalization, Plug and Play, Universal Explanation Platform), (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)), (No. 2022-0-00184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics).
References


Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default [TODO] to [Yes], [No], or [N/A]. You are strongly encouraged to include a justification to your answer, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section ??.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Figure 5 includes the error range obtained by multiple runs
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]