Understanding Integrated Gradients with SmoothTaylor for Deep Neural Network Attribution

Gary S. W. Goh[†], Sebastian Lapuschkin[‡], Leander Weber[‡], Wojciech Samek[‡] and Alexander Binder[†]
[†]ISTD Pillar, Singapore University of Technology and Design, Singapore 487372
[‡]Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany
gary_goh@mymail.sutd.edu.sg, alexander_binder@sutd.edu.sg
{sebastian.lapuschkin, leander.weber, wojciech.samek}@hhi.fraunhofer.de

***Abstract*—*Integrated Gradients* as an attribution method for deep neural network models offers simple implementability. However, it suffers from noisiness of explanations which affects the ease of interpretability. The *SmoothGrad* technique is proposed to solve the noisiness issue and smoothen the attribution maps of any gradient-based attribution method. In this paper, we present *SmoothTaylor* as a novel theoretical concept bridging *Integrated Gradients* and *SmoothGrad*, from the Taylor's theorem perspective. We apply the methods to the image classification problem, using the ILSVRC2012 ImageNet object recognition dataset, and a couple of pretrained image models to generate attribution maps. These attribution maps are empirically evaluated using quantitative measures for sensitivity and noise level. We further propose adaptive noising to optimize for the noise scale hyperparameter value. From our experiments, we find that the *SmoothTaylor* approach together with adaptive noising is able to generate better quality saliency maps with lesser noise and higher sensitivity to the relevant points in the input space as compared to *Integrated Gradients*.**

I. Introduction

Deep neural networks have displayed remarkable success in various large-scale, real-world and complex artificial intelligence tasks in computer vision [1]–[3] and natural language processing [4], [5]. However, these high performing non-linear neural models, unlike traditional machine learning models, act like a *black box* which suffers from poor input-to-output inference and interpretability. Due to the nature of how deep neural network algorithms are designed, it is difficult to explain *what* or why an *individual* input result in the model arriving at a particular output [6]. This major disadvantage hinders human experts to fully understand the basis and the reasoning of every prediction a deep neural model makes for each input, limiting the extent of its application in practice.

With the aim to better understand the complex input-to-output behavior of a deep neural network, a number of previous work [7]–[18] focus on the problem of attribution. Attributions measure the contribution of the model's output explained in terms of its input variables. For instance, for image classification systems, an attribution method assigns a relevance score to every pixel of the input image that explains for the model's predicted class. There are many applications where such an ability to "explain" for a complex model's decision is crucial. Attributions act as supporting evidence to explain the rationale of a model's decision. This helps to facilitate the building of trust between humans and automated systems [19], and encourage higher adoption of deep neural networks in practice, especially in high-risk application areas. The importance of attribution is more apparent in view of the recent vulnerability discoveries in deep neural networks against malicious and yet unnoticeable to-the-human-eye adversarial attacks [20], [21].

Sundararajan et al. [15] proposed *Integrated Gradients* (*IG*) as an attribution method for deep neural networks, which unlike other methods [9]–[11], [13], [14], [16], [17], is fully independent of the composition of the model's structure, and can be easily implemented with access to just the input's gradients after back-propagation. As such, it is computationally efficient to compute, and can be widely applied to various deep neural networks architectures and tasks.

However, *IG* require a selected baseline as a benchmark, which raises the question on how such a baseline is to be chosen. In addition, just as with other gradient-based methods [7], [8], *IG* often create attribution maps that are noisy which affects the ease of its interpretability. For example, compare the saliency maps (attribution maps visualized by a 2D image) of *IG* (center two) with other methods [8], [13], [22] in Figure 1, which is based on a DenseNet [1] with 121 layers pretrained for the ImageNet image classification task. The noisiness of its explanations is visually striking.

Those noise pixels seemingly scattered at random across the maps as shown in Figure 1 may indeed reflect the true behavior of the gradients of the deep neural model: as the networks get deeper, the gradients across the input space fluctuate more sharply, resembling white noise, which is described as the shattering gradient problem [23]. To tackle the noisiness issue, Smilkov et al. [22] proposed the *SmoothGrad* technique, which uses a random sampling strategy around the input with averaging of the obtained attributions to produce visually sharper attribution maps.

In this paper, our contributions are as follows:

- We present *SmoothTaylor* as a theoretical concept bridge between *IG* and *SmoothGrad*. Unlike *IG*, it does not require a selected fixed baseline. Under additional assumptions, *SmoothTaylor* is an instance of *SmoothGrad*. Regarding novelty, *SmoothTaylor* is derived from the Taylor's theorem. Experimental results show that *SmoothTaylor* is able to produce higher quality attribution maps that are more sensitive and less noisy as compared to *IG*.

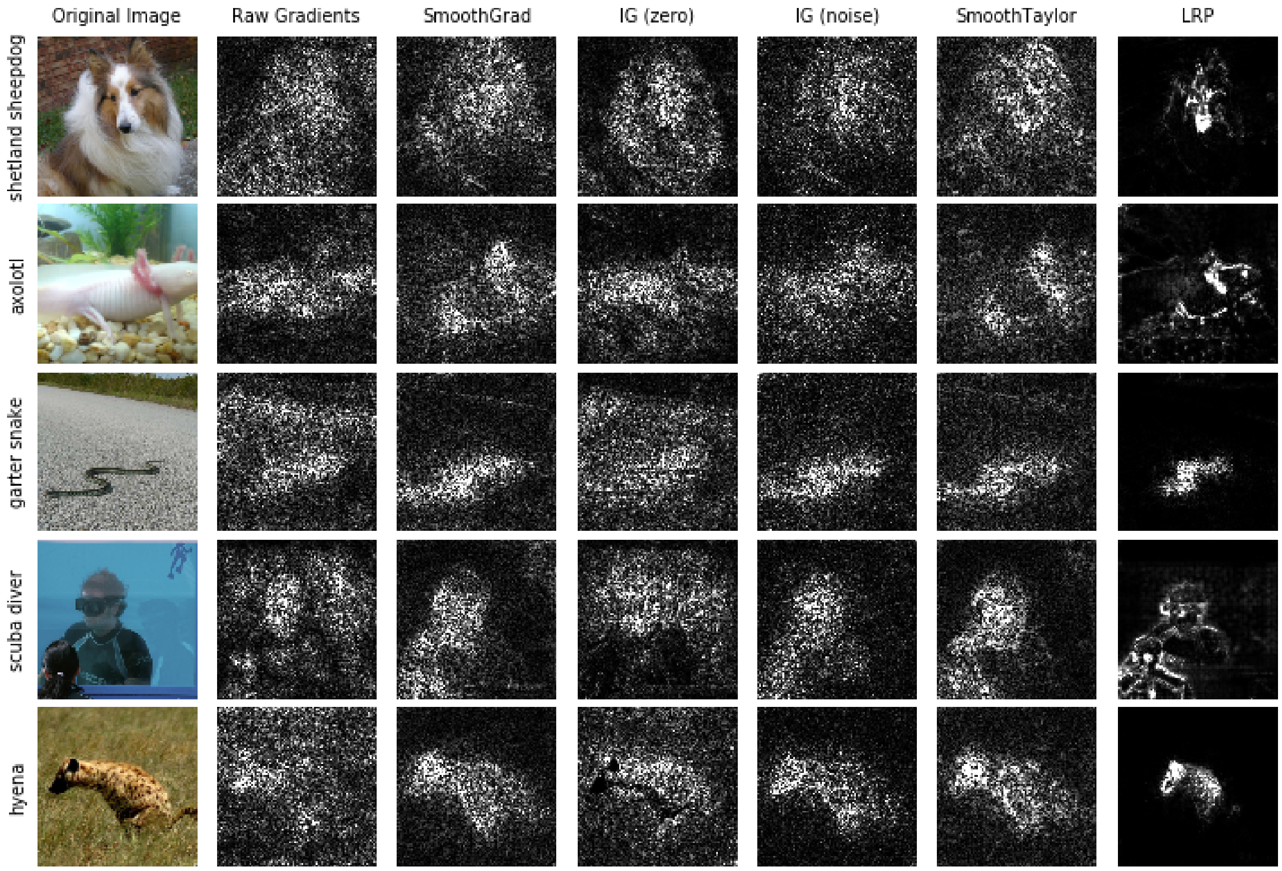

Fig. 1. Comparison of saliency maps computed by different attribution methods. These saliency maps show the relative contributions of each input pixel that explains for the model's prediction. Columns from the left: original input image; raw gradients; *SmoothGrad*; *IG* with zero as the baseline ($M = 50$); *IG* with noise as the baseline ($N = 1$); *SmoothTaylor* ($\sigma =$5e$-$1, $R = 150$); *Layer-wise Relevance Propagation*. Setup: DenseNet121 image classifier pretrained for ImageNet. Normalized absolute values are used to visualize the attribution maps and values above 99th percentile are clipped.

- From the perspective of gradient shattering, we explain why *SmoothGrad* and *SmoothTaylor* deteriorate with too small amount of added noise.
- We emphasize smoothness as a second quality measure for attribution and introduce multi-scaled average total variation as a new evaluation measure for smoothness of the attribution maps.
- We further propose adaptive noising for individual input samples to optimize for either predictor sensitivity of the generated attribution map or the noisiness of it. We show that it results in large improvements in performance compared to constant noise levels.
- This paper aims at a better understanding of existing gradient-based attribution methods.

The rest of the paper is organized as follows. Section II briefly describes *IG* and *SmoothGrad*. In Section III, we derive *SmoothTaylor* as a theoretical bridging concept. Next, in Section IV, we conduct experiments by applying the attribution methods on a large-scale image classification problem to generate attribution maps. These attribution maps are quantitatively evaluated and compared. Adaptive noising is discussed in Section V.

II. Preliminaries

A. Integrated Gradients

Suppose one aims to explain the prediction of a deep neural network represented by a function f for input x. The integrated gradient [15] for the i^{th} dimension of the input is defined as follows:

$$IG_i(x,z) := (x_i - z_i) \times \int_{\alpha=0}^{1} \frac{\partial f(z + \alpha \times (x-z))}{\partial x_i} d\alpha \quad (1)$$

The gradient of f in the i^{th} dimension is denoted by $\frac{\partial f(x)}{\partial x_i}$, and z is a selected input baseline. In practice, the path integral is usually approximated by a summation across discrete small intervals m with M steps along the straightline path from input x to baseline z, as follows:

$$IG_i(x,z) \approx (x_i - z_i) \times \frac{1}{M} \sum_{m=1}^{M} \frac{\partial f(z + \frac{m}{M} \times (x-z))}{\partial x_i} \quad (2)$$

Note that the attributions of the *IG* method satisfy some desirable properties. First, it satisfies *implementation invariance* since the computations are only based on the gradients of f, and are fully independent on any aspects of the models.

It also fulfils the *completeness* axiom, which ensures that the attributions add up to the output difference between input x and baseline z (i.e. $\sum_i IG_i(x, z) = f(x) - f(z)$).

Thus, it is recommended to choose baseline z to be zero (with a near-zero score, i.e. $f(z) \approx 0$) to represent the absence of input features. This acts as a basis for comparison and thus allows for the interpretation of the attributions to be a function of solely the individual input features. For images, this is a fully black image, which is argued to be a natural and intuitive choice. However, a black image is usually a statistical outlier to most pretrained models, which makes explanations relative to implausible outlier points seem irrelevant. Another disadvantage of using zero as the baseline is that input features that are zero or near-zero will never appear on the attribution maps since multiplier $x_i - z_i$ will be almost close to zero. For example in Figure 1, saliency maps of *IG* with zero as the baseline mostly fail to highlight objects of interests represented by dark-colored pixels.

An alternative baseline with the same near-zero score property is also proposed – uniform random noise. To address the issue of which random noise baseline to be chosen, a valid approach is to draw different noise baselines $z^{(n)}$ to compute N *IG* mappings, and average over them[1]:

$$\overline{IG}_{noise}(x) = \frac{1}{N}\sum_{n=1}^{N} IG(x, z^{(n)}) \tag{3}$$

This slight extension does seem to improve *IG* and result in more sensitive attribution maps with less noise, though there is still much room for improvement. Moreover, it should be noted that uniform random noise is also an unseen outlier, thus it guides to generate explanations that are no more meaningful than the zero baseline. Perhaps, the need for this method to fix a baseline that is consistent enough for all inputs, and at the same time does not deviate too far from the points in the dataset, is a fundamental flaw in its design, as such a baseline may not exist.

B. SmoothGrad

While the original *SmoothGrad* technique [22] smooths the raw gradients over the input space, it can be viewed as a general procedure which computes an attribution map by averaging over multiple attribution maps of an arbitrary gradient-based attribution method (denoted as $\mathcal{M}$) with multiple N' noised inputs:

$$SmoothGrad(x) = \frac{1}{N'}\sum_{n=1}^{N'} \mathcal{M}(x+\epsilon),\ \epsilon \sim \mathcal{N}(0, \sigma'^2) \tag{4}$$

Gaussian noise with parameter σ' is used to smoothen the input space of the attribution method and construct visually sharper attribution maps. It is briefly discussed in their paper that σ' needs to be carefully selected to get the best result. If too small, the attribution maps are still noisy; if too large, the maps become irrelevant.

III. SmoothTaylor

In this section, we explain the derivation of *SmoothTaylor*. Firstly, we discuss the motivation of our proposed improvement from the Taylor's theorem approximation perspective. Any arbitrary differentiable function f can be approximated by Taylor's theorem with the first order term while ignoring all other higher order terms:

$$f(x) \approx f(z) + \sum_i (x_i - z_i)\frac{\partial f(z)}{\partial x_i} \tag{5}$$

This yields an explanation, which describes how the output of the model $f(\cdot)$ in point x is different from the output of the same model in point z. Notably, it is an explanation for x relative to z. This raises the valid issue on how the point z should be chosen.

Secondly, in statistics, a valid method to deal with uncertainty is to compute an average over an uncertain quantity. In the case of uncertainty about which point z should be chosen, the proper approach is to draw several roots $z^{(r)}$ (according to some method which we defer the discussion till later) and average over them, so as to improve the power of the approximation:

$$f(x) \approx \frac{1}{R}\sum_{r=1}^{R}\left[f(z^{(r)}) + \sum_i (x_i - z_i^{(r)})\frac{\partial f(z^{(r)}))}{\partial x_i}\right] \tag{6}$$

Equation (6), in turn, is a discrete approximation for the integral (with S which has to be a measurable set):

$$f(x) \approx \int_{z \in S} f(z) + \sum_i (x_i - z_i)\frac{\partial f(z)}{\partial x_i} dz \tag{7}$$

We are now ready to outline our method. Based on the concepts described above, the smooth integrated gradient in the i^{th} dimension of an input x within a set of roots $z \in S$ is defined as follows:

$$SmoothTaylor_i(x) := \int_{z \in S} (x_i - z_i)\frac{\partial f(z)}{\partial x_i} dz \tag{8}$$

Equation (8) has two salient differences to *IG* from Equation (1). First, the explanation point z_i in the inner product $(x_i - z_i)$ is part of the integral, whereas in *IG*, it is outside of it. Second, the integration set S is not a path from x to some point z as it was in *IG*.

Similarly, for the reason of efficient computation, the integral can also be approximated using a discrete summation over R multiple roots $z^{(r)}$:

$$SmoothTaylor_i(x) \approx \frac{1}{R}\sum_{r=1}^{R}(x_i - z_i^{(r)})\frac{\partial f(z^{(r)})}{\partial x_i}, z^{(r)} \sim S \tag{9}$$

Equation (9) is derived from the averaged Taylor's theorem approximation in Equation (6) by choosing a set of roots such that the model output score difference between each root $z^{(r)} \in S$ and input x is almost close to zero (i.e. $\forall r : f(x) - f(z^{(r)}) \approx 0$). As a result, the inner summation term $f(z^{(r)})$ is canceled out with $f(x)$, and the remaining

[1] https://github.com/ankurtaly/Integrated-Gradients/

terms can be explained as the sum of the smooth integrated gradients across all dimensions. Note that this loosely satisfies the *completeness* axiom just like the *IG* method. It also fulfils the *implementation invariance* property.

The next issue is to decide on a suitable method to generate the roots $z^{(r)}$. If one is interested in classification or segmentation as pixel-wise classification, then one would want to choose the set S to be a set of points where the prediction output class switches. However searching these points on the training dataset might result in roots which are too far away from the input x to be explained, which will impact the quality of the Taylor approximation. One alternative is to seek for a random set of points sufficiently close to x, so that the quality of the Taylor approximation is acceptable, and also sufficiently far away, so that the noise from the gradient shattering effect in deep networks [23] can be canceled out by averaging over many z from many different linearity regions. A simple approach, inspired by *SmoothGrad*, is to add a random variable ϵ to input x, where ϵ can be drawn from a Gaussian distribution with standard deviation σ being the noise scaling factor:

$$z^{(r)} = x + \epsilon, \text{where } \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{10}$$

The choice of the σ value should be carefully selected, and it is further discussed in Section V. This follows the principle of choosing $z^{(r)}$ to be close to x and also sufficiently far away, so that the need for a good Taylor approximation and averaging effect of the noise in the gradients can be balanced.

Theorem: If the roots in *SmoothTaylor* are chosen as per Equation (10), then the discrete version of *SmoothTaylor* as given in Equation (9) is a special case of *SmoothGrad* with $\mathcal{M} = \nabla f(x + \epsilon) \cdot \epsilon$.

This theorem does not hold for other choices of the set S in Equation (9), thus *SmoothTaylor* defines an algorithm class of its own.

SmoothTaylor offers an alternative formulation to *IG*, where the selection of a fixed baseline is not required. The above theorem establishes *SmoothTaylor* with a choice of roots as in Equation (10) as a theoretical bridging concept between *IG* and *SmoothGrad*.

IV. Experiments

We apply *SmoothTaylor* and *IG* [15] attribution techniques, and compare their results. We choose to analyze them on the image classification task. The goal is to compare the quality of the attribution maps computed by these two methods. To encourage reproducibility, we publicly release our source code[2]. Here, we describe our experiment setup and evaluation metrics.

A. Setup

We use the first 1000 images from the ILSVRC2012 ImageNet object recognition dataset [24] validation subset as the scope of our experiment. It is a 1000 multi-class image classification task, with each image preprocessed to be the size of 224×224 pixels. We choose two deep neural image classifier models, DenseNet121 [1] and ResNet152 [25], that are both pretrained on the ImageNet dataset to apply the attribution methods. We compute the attributions with respect to the function of the predicted class for each input image regardless of the ground truth label. Therefore, the attribution process is entirely unsupervised.

B. Hyperparameters

For the *SmoothTaylor* method, we vary the parameter values for the number of roots R to be 100, 150, and 200, and the noise scaling factor σ to be 3e$-$1, 5e$-$1, and 7e$-$1. The magnitudes of the noise scaling factor are decided to be roughly in the range of the average values of the inputs after normalization. For *IG*, we choose total steps M to be 50, and vary the type of baselines used. We use the zero (black image) baseline, and random uniform noise baselines with different samples sizes N to be 1, 5, 10, and 20.

C. Evaluation Metrics

Sundararajan et al. [15] argued against empirical methods for evaluating attribution methods, and thus decided to rely on an axiomatic approach to determine the quality of an attribution method. However, axiom sets might be incomplete, and for a data-driven science, a quantitative evaluation is often aligned with the goals. Furthermore, there are limitations to qualitative evaluation of attribution maps due to biases in human intuition towards simplicity whereas deep neural models which might be over-parametrized and thus of high complexity. Therefore, in this paper, we use the following two quantitative metrics:

1) Perturbation Approach: One such metric suggested by Samek et al. [26] relies on selecting the top salient regions of pixels in the input image by attribution and successively replacing them with random noise (also known as pixel perturbation), and then measuring the drop in model output scores. A higher score drop signifies a more sensitive attribution method, since the attributions are able to better identify the salient parts of the input that explain the model's output.

We describe our pixel perturbation evaluation procedure formally as follows. First, we use a sliding local window of kernel size $k \times k$ in the input image space to find an ordered sequence $\mathcal{O} = (r_1, r_2, ..., r_L)$ that contains the top-L most salient non-overlapping regions. The sorting of the regions is based on the average absolute attribution values of the pixels' location within each kernel window, from the highest to lowest (most relevant first). A high average absolute attribution value in a region r_l denotes a high presence of evidence that supports the model's prediction.

Second, we follow the sequence of ordered regions in $\mathcal{O}$ to apply the perturbations on. Let $g(x, r)$ be a function which performs the perturbation on some input image x at region r, where information in that region is removed by the replacement of the value of its pixels with random values

[2] https://github.com/garygsw/smooth-taylor

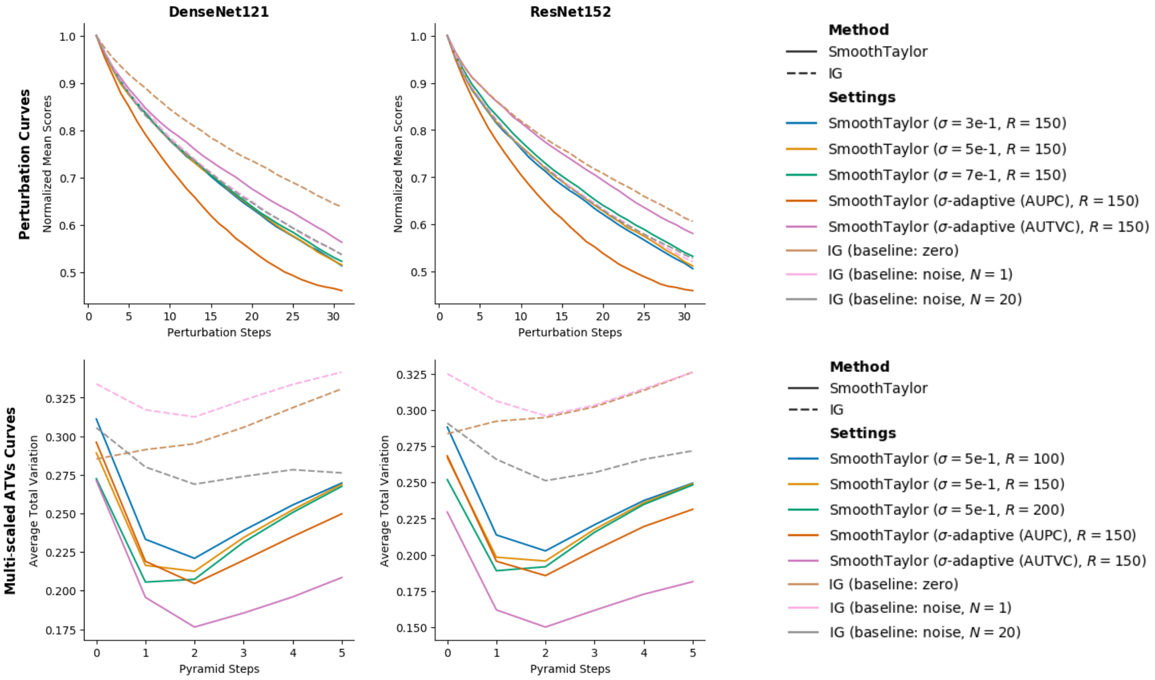

Fig. 2. Evaluation metrics curves; the lower the curve the better. Right: Legends. Top row: Perturbations curves. Bottom row: Multi-scaled TV curves. Left column: Based on DenseNet121. Right column: Based on ResNet152.

drawn from a uniform distribution across the valid input value range. The function g is then successively applied starting with the original input image $x^{(0)} = x$. The input image for the next step $x^{(l)}$ is iteratively updated after perturbation at step l for L times:

$$\forall\, 1 \leq l \leq L: \;\; x^{(l)} = g(x^{(l-1)}, r_l) \tag{11}$$

At each step l, we consider P number of different random perturbation samples and compute the mean score $\bar{y}^{(l)}$:

$$\bar{y}^{(l)} = \frac{1}{P} \sum_{p=1}^{P} f(x^{(l-1)^{(p)}}) \tag{12}$$

The perturbation with the median output score is selected as the actual perturbation to update. To quantitatively measure the strength of an attribution method, we look at how much these mean output scores drop with steps l. That can be quantified by taking the area under the perturbation curve (AUPC) (see Figure 2 (top)) after normalizing each mean score $\bar{y}^{(l)}$ at each step l with the original score $f(x)$, and averaged over all images in the dataset. Throughout our experiments, we use kernel size $k = 15$, number of perturbations $L = 30$, and perturbation sample size $P = 50$.

2) Average Total Variation: We use average total variation (ATV) as the second evaluation metric to measure the smoothness or the total amount of noise of each pixel with its local neighbors. We consider a saliency map $\mathcal{S}$ as vector of size $h \times w$ to represent every pixel. Taking only absolute values, a min-max normalization (with values above 99th percentile clipped off) is applied on an attribution map to construct a saliency map. The ATV of $\mathcal{S}$ is computed as follows:

$$ATV(\mathcal{S}) = \frac{1}{h \times w} \sum_{i,j \in \mathcal{N}} \|\mathcal{S}_i - \mathcal{S}_j\|_p \tag{13}$$

Here, $\mathcal{N}$ defines the set of pixel neighbourhoods (adjacent horizontal and vertical pixels) and $\|\cdot\|$ is the ℓ_p norm. We use the established ℓ_1-norm in our experiments.

In addition, we construct Gaussian pyramids [27] on the saliency maps by repeatedly scaling their dimensions down by 1.5 and applying a Gaussian smoothing filter to remove information. This process is repeated for each saliency map until the size of the map is smaller than 30×30 pixels. We then compute the ATV of the scaled and blurred saliency maps at each step – we call them multi-scaled ATVs. Subsequently, after averaged over all images, we take the area under the multi-scaled ATVs curve (AUTVC) (see Figure 2 (bottom)) as the measure quantity to evaluate the quality of an attribution method.

D. Results

We compute the attribution maps using a few different attribution methods based on two pretrained image classifiers on the ImageNet dataset. Examples of these attribution maps are visualized as saliency maps in Figure 1.

Qualitatively, we can observe that *SmoothTaylor* produces visually sharper saliency maps as compared to *IG*. In addition, they are better at highlighting distinctive regions that explain the model's prediction. While it is not the best method that produces the least noise or the most sensitivity (see saliency

TABLE I
AREA UNDER THE CURVES RESULTS.
NOTE: LOWER AUPC AND AUTVC IS BETTER.

Attribution Method		**Image Classifier Model**			
		DenseNet121		ResNet152	
IG					
baseline	N	AUPC	AUTVC	AUPC	AUTVC
zero	-	23.63	1.52	22.87	1.51
noise	1	21.51	1.62	21.05	1.54
	5	21.54	1.52	20.99	1.43
	10	21.46	1.45	**21.02**	1.37
	20	**21.43**	**1.39**	**21.02**	**1.32**
SmoothTaylor		DenseNet121		ResNet152	
σ	R	AUPC	AUTVC	AUPC	AUTVC
3e−1	100	21.24	1.28	20.83	1.20
	150	21.19	1.24	20.79	1.16
	200	**21.13**	1.22	**20.78**	1.14
5e−1	100	21.25	1.23	21.00	1.14
	150	21.20	1.19	20.95	1.10
	200	**21.13**	1.16	20.86	1.07
7e−1	100	21.39	1.20	21.37	1.08
	150	21.30	1.15	21.32	1.04
	200	21.30	**1.12**	21.14	**1.01**

TABLE II
AREA UNDER THE CURVES RESULTS FOR *SmoothTaylor* WITH EXTREME HYPERPARAMETER VALUES.
NOTE: LOWER AUPC AND AUTVC IS BETTER.

SmoothTaylor		**Image Classifier Model**			
Hyperparameters		DenseNet121		ResNet152	
σ	R	AUPC	AUTVC	AUPC	AUTVC
5e−1	10	21.74	1.55	21.43	1.43
1e−4	100	23.45	1.79	23.00	1.55
1e−3	100	23.60	1.53	23.14	1.48
1e−2	100	23.90	1.57	23.46	1.23
1e−1	100	22.03	1.43	**21.44**	1.22
1	100	**21.88**	**1.17**	22.16	**1.04**
2	100	23.54	1.19	24.48	1.27

TABLE III
AREA UNDER THE CURVES RESULTS WITH ADAPTIVE NOISING.
NOTE: LOWER AUPC AND AUTVC IS BETTER.

SmoothTaylor		**Image Classifier Model**			
Hyperparameters		DenseNet121		ResNet152	
σ	R	AUPC	AUTVC	AUPC	AUTVC
Adaptive-AUPC	150	**19.55**	1.14	**19.30**	1.05
Adaptive-AUTVC	150	22.14	**0.99**	22.52	**0.85**

maps produced by Layer-wise Relevance Propagation [13]), *SmoothTaylor* offers ease of implementation and fulfils the two current fundamental axioms of an attribution method.

Next, we discuss the results using quantitative evaluation measures. A summary of the experimental results is shown in Table III with the AUPC and AUTVC values for each experiment run. The Simpson's rule is used to compute the area under the curves. We analyze the results based on two objectives – sensitivity and noise level, and also compare the results based on two different classifier models.

1) Sensitivity: As observed in Figure 2 (top), when compared to *IG*, the attribution maps of *SmoothTaylor* are able to cause a larger classification score drop as perturbation step increases. Expectedly, the AUPC values for *SmoothTaylor* are also lower, showing that *SmoothTaylor* is more sensitive to relevant explanations points in the input space than *IG*. The averaged *IG* with noise baselines are shown to have large improvements; almost close to the performance of *SmoothTaylor* at our chosen hyperparameters, though still a little worse. Their improvements also produce diminishing marginal returns as N increases beyond more than 5. On closer inspection with Table III, it shows that our choice for σ values did not produce any significant effect on the AUPC values, which is worth investigating further in Section IV-E. However, the AUPC values clearly decrease as R increases. This is expected as the "smoothing" effect is greater when we draw more roots, resulting in a statistically better representation of z which improves the power of the Taylor approximation.

2) Noise level: The *SmoothTaylor* method clearly generates attribution maps that are much less noisy than *IG*. As seen in multi-scaled ATV curves in Figure 2 (bottom), all the curves for *SmoothTaylor* are lower that the curves for *IG*. We also compare the effect of σ and R on the noisiness of the attribution maps of *SmoothTaylor*. First, the AUTVC values decrease as R increases. This is also expected due to the increase "smoothing" effect. Second, the AUTVC values seem to increase as σ increases. However, we believe that this relationship is not monotonically true, as the selection of our σ values may be too low across all images in the dataset. We discuss this further in Section IV-E.

3) DenseNet121 vs. ResNet152: The sensitivity improvements in the perturbation curves by *SmoothTaylor* over *IG* is noticeably lesser for ResNet152 as compared to DenseNet121. One hypothesis is that the gradients from ResNet152 are less noisy to begin with, since residual networks are shown to have reduced shattering gradients effect. Thus, with more reliable gradients to explain for the model's prediction, the effectiveness of smoothing is also reduced.

E. Noise Hyperparameter Sensitivity Analysis

We choose a range of σ values as high as 2 and as low as 1e−4, while fixing R to be 100. The effects of different values of the noise scale parameter for *SmoothTaylor* are displayed in Figure 3, and its results are summarized in Table II.

We can observe that for too small noise choices such as 1e−4 or 1e−3, the AUPC sensitivity is lower than for choices in the order of 1e−1. This can be explained from the effect of gradient shattering in deep networks: when the gradient has a large component resembling white noise, as observed in [23], then using averages is a statistically reasonable attempt to remove the white noise component. Rectified Linear Units (ReLu) networks consist of zones with locally linear predic-

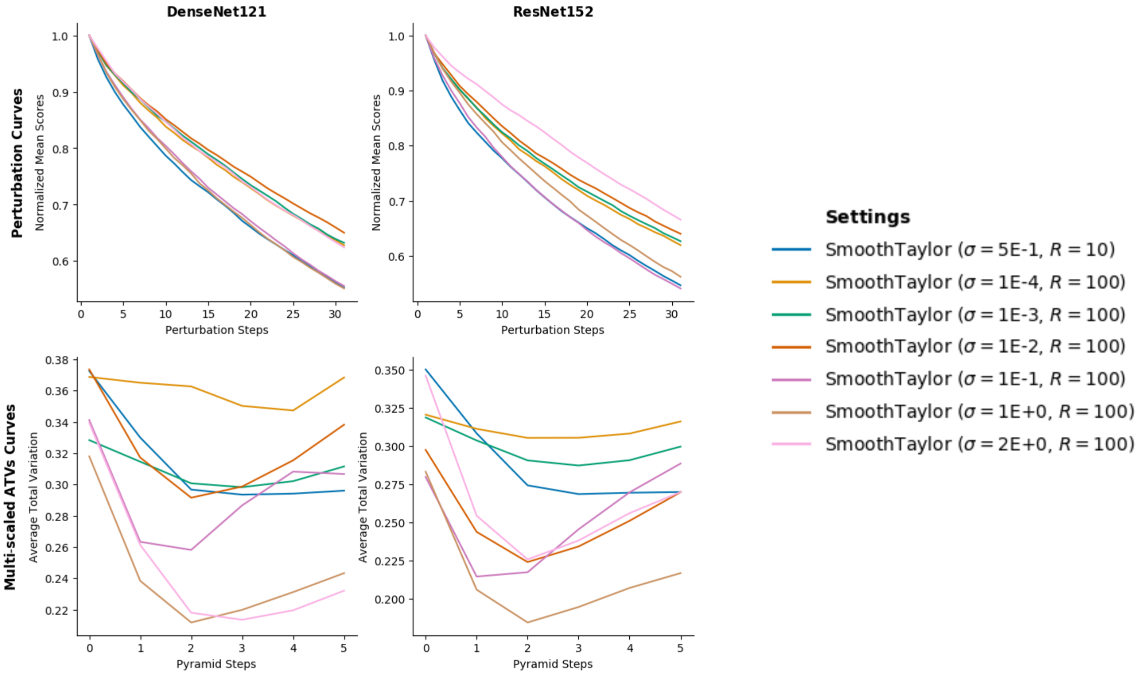

Fig. 3. Evaluation metrics curves for the study of the impact of varying the noise hyperparameter; the lower the curve the better. Top row: Perturbation curves. Bottom row: Multi-scaled TV curves. Left column: Based on DenseNet121. Right column: Based on ResNet152.

tions – see Figure 3 in [28] for a clear illustration of this effect.

The gradient is constant within each such zone. Above averaging requires to sample the gradient at many different local linearity zones around the sample of interest x. In particular averaging requires z_i to be outside of the linearity zone in which x is in. This explains why a very small amount of noise will not result in an effective averaging of white noise, as most of the samples z_i would just stay in the local linearity zone of x and fail to sample different gradient values.

The size of the local linearity zone is sample-dependent [28]. This observation supports the claim that the noise scale σ needs to be carefully calibrated within a certain range (i.e. it cannot be too small or too big) for every individual sample x in order for the attribution maps of *SmoothTaylor* to be of high quality. Therefore, based on this observation, we go further and propose an adaptive improvement to *SmoothTaylor* in the next section.

V. Adaptive Noising

Ideally, the value of noise scale σ should depend on each individual input, and not generally fixed to all inputs. Thus, we propose an adaptive noising technique to search for an optimal noise scale value for each input, so as to optimize the *SmoothTaylor* method.

Algorithm 1: Adaptive Noising

```
Parameters: Max. iterations $i_{max}$, learning rate $\alpha$, learning decay $\gamma$, max.
            stop count $s_{max}$
Input      : Input $x$, root size $R$, model $f$
Output     : Optimal $\sigma^*$ value
begin
    $\sigma \leftarrow \frac{1}{N} \sum |x|$;
    AUC $\leftarrow$ ComputeAUC($x, R, f, \sigma$);
    $i \leftarrow 1$; $s \leftarrow 0$; $\sigma^* \leftarrow \sigma$; AUC$^* \leftarrow$ AUC;
    while $i \leq i_{max}$ do
        AUC$_s \leftarrow$ ComputeAUC($x, R, f, |\sigma + \alpha|$);
        if AUC$_s >$ AUC then
            $\sigma \leftarrow |\sigma - \alpha|$;
            AUC$_s \leftarrow$ ComputeAUC($x, R, f, \sigma$);
        else
            $\sigma \leftarrow |\sigma + \alpha|$;
        end
        if AUC$_s >$ AUC then
            if $s \leq s_{max}$ then
                $\alpha \leftarrow \alpha * \gamma$; $s \leftarrow s + 1$;
            else
                break
            end
        else
            $s \leftarrow 0$;
            if AUC$_s <$ AUC$^*$ then
                AUC$^* \leftarrow$ AUC$_s$; $\sigma^* \leftarrow \sigma$;
            end
        end
        AUC $\leftarrow$ AUC$_s$; $i \leftarrow i + 1$;
    end
end
```

We adopt an iterative heuristic line search approach to design our algorithm. The goal is to find an optimal value for σ such that the attribution maps can be the most sensitive or least noise (quantified by AUPC or AUTVC respectively). As such, while fixing R, we search for σ^* for each input such that the AUPC or AUTVC of its attribution map is minimized. We describe our algorithm in Algorithm 1.

In our proposed iterative optimization procedure, we search for σ^* within maximum iterations of i_{max}. We include an early stopping mechanism with maximum stop count s_{max}. At each iteration, σ is updated with learning rate α which direction depends on a line search. The learning rate is reduced by a factor learning decay $\gamma < 1$ whenever the current iteration's AUC is greater than the previous one. In our experiment, we use $R = 150$ and set maximum iterations $i_{max} = 20$, maximum stop count $s_{max} = 3$, learning rate $\alpha = 0.1$, learning decay $\gamma = 0.9$, and use the same setup from the AUC computation in our earlier experiments.

We report the results from using adaptive noising in Table III and compare with the results from previous experiment runs. With adaptive noising, we are able to obtain the best AUPC or AUTVC values among all runs. However, it is to be noted that computing AUPC is computationally expensive and slow while computing AUTVC is much faster. The results conclusively show that *SmoothTaylor* with adaptive noising is preferable over constant noise injection.

VI. Conclusion

Explaining for all deep neural model decisions is a huge challenge given the vast taxonomy of model types and scope of problems. Thus it is crucial to find a simple attribution method that is easily applied to various model architectures so as to encourage widespread usage. In this paper, we bridge *IG* and *SmoothGrad* and proposed *SmoothTaylor* from the Taylor's theorem perspective. In our experiments, we also introduce multi-scaled average total variation as a new measure for noisiness of saliency maps. We further proposed adaptive noising as a hyperparameter tuning technique to optimize our proposed method's performance. From the experimental results, *SmoothTaylor* is able to produce attribution maps that are more relevance-sensitive and with much less noise as compared to *IG*.

Acknowledgment

The research is supported by the National Research Foundation, Prime Minister's Office, Singapore, under its CREATE programme, Singapore-MIT Alliance for Research and Technology (SMART) Future Urban Mobility (FM) IRG.

References

[1] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2261–2269.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[3] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, 2019, pp. 10 691–10 700.

[4] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding Back-Translation at Scale," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 489–500.

[5] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, 2017.

[6] F. Fan, J. Xiong, and G. Wang, "On Interpretability of Artificial Neural Networks," 2020. [Online]. Available: http://arxiv.org/abs/2001.02522

[7] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K. R. Müller, "How to explain individual classification decisions," *Journal of Machine Learning Research*, vol. 11, pp. 1803–1831, 2010.

[8] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside Convolutional Networks: Visualising image classification models and saliency maps," in *International Conference on Learning Representations, Workshop Track Proceedings*, 2014, pp. 1–8.

[9] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," *European Conference on Computer Vision*, pp. 818–833, 2014.

[10] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for Simplicity: The All Convolutional Net," in *International Conference on Learning Representations, Workshop Track Proceedings*, 2015, pp. 1–14.

[11] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning Deep Features for Discriminative Localization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929.

[12] L. M. Zintgraf, T. S. Cohen, and M. Welling, "A New Method to Visualize Deep Neural Networks," in *Workshop on Visualization for Deep Learning, International Conference on Machine Learning*, 2016.

[13] A. Binder, G. Montavon, S. Lapuschkin, K. R. Müller, and W. Samek, "Layer-wise relevance propagation for neural networks with local renormalization layers," in *International Conference on Artificial Neural Networks*, 2016, pp. 63–71.

[14] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International Conference on Machine Learning*, vol. 7, 2017, pp. 4844–4866.

[15] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic Attribution for Deep Networks," in *International Conference on Machine Learning*, vol. 7, Sydney, Australia, 2017, pp. 5109–5118.

[16] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K. R. Müller, "Explaining nonlinear classification decisions with deep Taylor decomposition," *Pattern Recognition*, vol. 65, pp. 211–222, 2017.

[17] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 336–359, 2020.

[18] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, ser. LNCS. Springer, 2019, vol. 11700.

[19] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An Overview of Interpretability of Machine Learning," in *IEEE International Conference on Data Science and Advanced Analytics*, 2018, pp. 80–89.

[20] A. Nguyen and C. L. Date, "Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images," *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[21] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 86–94.

[22] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "SmoothGrad: Removing noise by adding noise," in *Workshop on Visualization for Deep Learning, ICML*, 2017. [Online]. Available: http://arxiv.org/abs/1706.03825

[23] D. Balduzzi, M. Frean, L. Leary, J. Lewis, K. Wan-Duo Ma, and B. Mcwilliams, "The Shattered Gradients Problem: If resnets are the answer, then what is the question?" in *International Conference on Machine Learning*, 2017.

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 770–778.

[26] W. Samek, A. Binder, G. Montavon, S. Bach, and Klaus-Robert Muller, "Evaluating the Visualization of What a Deep Neural Network Has Learned," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 8, no. 11, pp. 2660 – 2673, 2017.

[27] P. J. Burt and E. H. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communications*, vol. COM-3, no. 4, pp. 532–540, 1983.

[28] R. Novak, Y. Bahri, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein, "Sensitivity and Generalization in Neural Networks: an Empirical Study," in *International Conference on Learning Representations*, 2018.