Initialization Noise in Image Gradients and Saliency Maps

Ann-Christin Woerl, Jan Disselhoff, Michael Wand
Institute of Computer Science
Johannes Gutenberg University Mainz, Germany
{awoerl, jadissel, wandm}@uni-mainz.de

Abstract

In this paper, we examine gradients of logits of image classification CNNs by input pixel values. We observe that these fluctuate considerably with training randomness, such as the random initialization of the networks. We extend our study to gradients of intermediate layers, obtained via GradCAM, as well as popular network saliency estimators such as DeepLIFT, SHAP, LIME, Integrated Gradients, and SmoothGrad. While empirical noise levels vary, qualitatively different attributions to image features are still possible with all of these, which comes with implications for interpreting such attributions, in particular when seeking data-driven explanations of the phenomenon generating the data. Finally, we demonstrate that the observed artefacts can be removed by marginalization over the initialization distribution by simple stochastic integration.

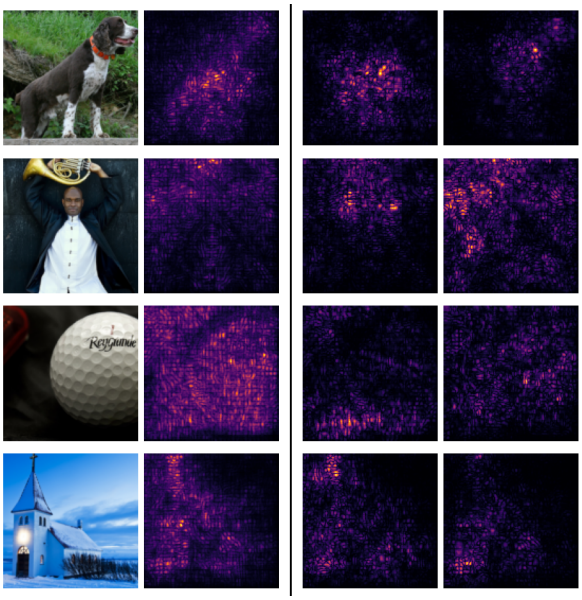


Figure 1. Logit-by-image gradients (ResNet18 on "ImageNette" [12]). First column: reference image; second column: mean over 50 models, column 3-4: single models with random initialization.

1. Introduction

Deep neural networks have revolutionized pattern recognition, detecting complex structures at accuracies unheard of just a few years back. Unsurprisingly, the newly gained ability to model complex phenomena comes at costs in terms of interpretability — it is usually not obvious how nonlinear, multi-layer networks reach their conclusions. Correspondingly, a lot of research has focused on developing *interpretation* methods for explaining how deep networks make decisions [11], and this often takes the form of *attributing* decisions to subsets of the data. In the case of image classification, this usually leads to *saliency maps* highlighting the image area containing decisive information [25, 26, 29, 30, 32, 35].

Strong classifiers trained from example data combined with suitable attribution methods have opened up a new approach to empirical research: *understanding phenomena* by *interpreting learned models* [9]. We often know of posterior outcomes (for example, tumor growth rates or treatability with certain medication) but do not understand how these are related to prior data (say, findings from histological tissue samples). If we are able to train a strong classifier that can predict posterior outcomes from prior data, an attribution method could potentially explain which aspects of the data predict this outcome (for example, which visual features in the histology indicate a negative or positive therapeutic prognosis [38]), thereby providing new insight into the phenomenon at hand.

For these kinds of research approaches, the classifier might only be an auxiliary tool: In terms of attribution, we are not primarily interested in explaining how *the classifier* reaches its decision (which, of course, would be highly relevant when studying potential data leakage or the fairness of decisions [4, 27]), but our actual goal is to accurately characterize which *features in the data* are related to *the phenomenon* to be explained. Ultimately, it is of course impossible to be sure whether an ad-hoc classifier (even with

great statistical performance and hypothetical perfect attribution) actually does exploit all relevant information (and only this), but we would of course in such cases make an effort to avoid wrong or incomplete information or misattributions that we are already aware of.

The main insight and contribution of this paper is to point out one such source of fluctuations in attributions, the impact of which, to the best of our knowledge, has not yet been documented in literature so far: *In nonlinear CNNs, image gradients of network outputs can contain significant training* noise, i.e., noise from (in particular) random weight initialization and stochastic batch selection (Fig. 1). The level of such noise, i.e., information unrelated to the data itself, *often exceeds the level of the attribution signal*, and variability includes coarse-scale variations that could suggest qualitatively varying attributions. Surprisingly, this still holds (and can even be worse) for more sophisticated attribution techniques, including popular approaches such as SHAP [20], LIME [25], DeepLIFT [29], or Integrated Gradients [35]. Even class activation maps (including top- and intermediate-level GradCAMs [26], the former to a lesser degree) can be affected by noise to an extent that could plausibly alter coarse-scale attribution.

Exploring the phenomenon further, we observe that gradient noise grows with depth, is rather weak in simple convex architectures (linear softmax regression), and dampened stochastically for wide networks (as suggested by the known convexity in the infinite-width limit [7, 19]). This indicates that nonlinearity and nonconvexity might play an important role in causing the problem by either amplifying numerical noise or convergence to different local minima.

We further show that training noise artifacts can be removed by marginalization [37], which in practice can be implemented with simple stochastic integration: By averaging the results of tens of independently initialized and trained networks, signal-to-noise-levels can be brought to acceptable levels. We also demonstrate that the same stochastic ensembling technique also improves the visual quality of feature visualization by optimization [23]. While marginalization incurs non-trivial additional computational efforts, it can remove a significant source of uncertainty when explaining how data features are related to outcomes in previously unknown ways.

2. Related Work

In this section, we discuss related work on attribution methods as well as findings on shortcomings and methods for quality improvements.

Gradient-based interpretation: Taking the gradient of the output of a deep neural network w.r.t. input data converts a complex nonlinear model into a local linear approximation, which can be interpreted as a saliency map [30]. Closely related representations are obtained by deconvolution [40] and "guided" gradients [34], which both only differ in the masking of negative intermediate values during backpropagation.

Noise artifacts: Image gradients are typically very noisy. SmoothGrad [24, 32], performs local stochastic integration in image space to denoise saliency maps. While very effective in reducing apparent noise, our experiments show that SmoothGrad results can still suffer from training variability. Suppressing negative [34] or small activations [14] also leads to denoising. However, even just suppressing negative contributions in "guided" gradient methods appears to already decouple results from training [1]. Computing gradients only at higher level network layers (via GradCAM [26]) also improves the quality of saliency maps (typically at lower resolutions due to pooling layers) but our experiments show that training noise still causes fluctuations.

Non-local perturbation: A fundamental problem of gradients is that they only reflect local changes and thus plausibly overlook relevant features that are already present in full saturation. This can be addressed by non-local comparisons against baselines, for example by integration [35] or feature selection [29]. High-level GradCAM also appears to avoid the issue by examining how complex composite features are classified by the final layer(s) [13]. Linear models can still suffer from complexity issues [18]. Careful feature selection (by fitting regularized surrogate models in "LIME" [25] or selecting the most informative sets in "SHAP" [20]) can also improve results substantially — LIME and SHAP are currently the most popular attribution packages on GitHub [2].

Alternative attribution methods: Model interpretation techniques include withholding data, for example by blank overlays [40], blurring [8], or maximizing contributions from a suitable background distribution [13]. Self-attention can also be used as an information source [3, 6]. Our paper focuses on gradient and perturbation methods. An initial experiment following the method of [13] indicates less susceptibility to training noise, but a comprehensive study of alternative approaches is left for future work.

Limitations of attribution: Explaining phenomena by data-driven learning remains challenging [9, 22]. Aside from the central conceptual problem of linking classification and attribution to the structure of data, the statistical nature of available classifiers has been identified as a problem, too: Recent work [21, 22] cautions the reader to treat statistically initialized and trained models as random variables and to subject them to marginalization, as we do in this paper. While these papers describe general strategies for error analysis, we specifically study variability in popular gradient and perturbation-based methods and find evidence of significant issues with initialization noise that were not previously reported (and which one might not intuitively

expect to encounter in approaches such as SmoothGrad, GradCAM, integrated gradients, or SHAP/LIME that visually clean up saliency maps).

Further limitations of saliency maps have also been discovered by Adebayo et al. [1], who show that many popular methods can become independent of higher layer parameters and invariant under random relabeling of data. As a result, they primarily yield information on prior knowledge encoded in the architecture, rather than offering data-specific insights. In their study, simple gradients and GradCAM show the best behavior. Our findings are orthogonal: training noise affects all of the tested methods. Saliency maps are also sensitive to adversarial perturbations, where only minimal augmentations are required to create arbitrary results [5].

3. Method

3.1. Saliency Maps

We begin by giving some formal definitions. We denote the *input* of the network as vector $x \in \mathbb{R}^d$. A neural network computes a map $f_\theta : \mathbb{R}^d \to \mathbb{R}^C$, where C is the dimension of the output; in our experiments we consider only classification, so C also denotes the number of classes. $\theta \in \mathbb{R}^{d_p}$ describes the parameters of the network. In practical settings, the values for θ are the result of a training process. Formally, the result is dependent on the initialization weights θ_0, the data set D, and training hyperparameters, such as *optimizer, batchsize, learning rate*, as well as the random choices such as the ordering of batches. We use T to denote the function that transforms initial into final parameters using data D, $\theta = T(\theta_0, D)$. In this view, both θ_0 and T are random variables, drawn from distributions $\theta_0 \sim p(\theta_0), T \sim p(T)$.

We now define a *saliency method* as a deterministic map

$$S_{f_\theta} : \mathbb{R}^d \to \mathbb{R}^d, \tag{1}$$

that takes an input vector x and returns an output of the same shape, with the goal of "explaining" the behavior of f_θ by highlighting salient entries in x [1]. Taking the influence of training and initialization into account, the salience mapping becomes

$$S_{f_\theta}(x) = S_{f_{T(\theta_0, D)}}(x), \tag{2}$$

i.e., f_θ is completely determined given training randomness $T \sim p(T)$, initialization $\theta_0 \sim p(\theta_0)$, and dataset D.

In this formulation, the result of the saliency map depends not only on the input x, but on the training scheme, initialization, dataset and architecture encoded in f (Fig. 2).

Most of these values describe unwanted influences on the saliency map if we are looking for patterns belonging to the problem that is modeled. In other words, the choice of architecture or initialization does not influence the underlying patterns of the phenomenon of interest, but it still has an influence on the saliency map. In the following, we will call factors that can influence the saliency map but are independent of the phenomenon **extrinsic**, while factors that belong solely to the phenomenon are **intrinsic**. Depending on the specific problem, even the dataset and its sampling can contain extrinsic information we might want to remove; however, in this paper, we treat D as fixed, non-random information.

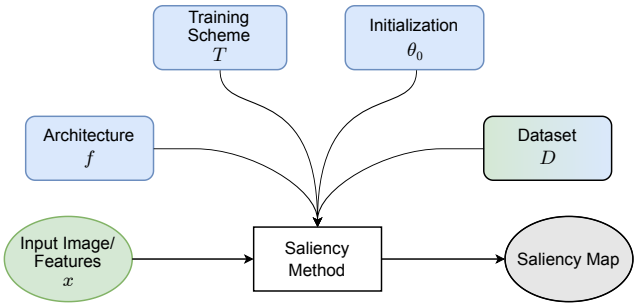


Figure 2. Typical Influences on Saliency Methods. When such methods are used to explain phenomena, the influence of factors extrinsic to the problem (here in blue) should be minimized. While biases in a dataset can be seen as extrinsic factors, the underlying true distribution is intrinsic to the phenomenon.

Note that saliency methods that are implementation agnostic – or even treat the model as a blackbox – can still be influenced by such extrinsic factors, as architecture and initialization fundamentally influence which function is learned by the model. In general, we want to minimize the effect of these extrinsic factors in order to produce a clear signal of the phenomenon of interest.

3.2. Bayesian Marginalization

In a Bayesian view, irrelevant random factors can be removed by marginalization [9, 37], i.e., integrating the joint distribution over all variants. For noise introduced from initialization and random training choices, this is simple. We just estimate the mean saliency map as

$$\mu = \mathbb{E}_{\theta_0 \sim p(\theta_0), T \sim p(T)} \left[S(f_{T(\theta_0)}, x) \right] \tag{3}$$

Note that $\mu \in \mathbb{R}^d$ is again vector in input shape (i.e., in our experiments, an image). In order to quantify the level of noise, we also compute the pixel-wise variance $\sigma \in \mathbb{R}^d$ of the saliency maps:

$$\sigma^2 = \mathbb{E}_{\theta_0 \sim p(\theta_0), T \sim p(T)} \left[S(f_{T(\theta_0)}, x) - \mu \right]^2 \tag{4}$$

Remark: An ideal approach should also remove the extrinsic influence of the choice of architecture. However, marginalization over architecture is difficult conceptually

(what would be an appropriate general set of architectures?) and computationally. For this reason, we restrict ourselves to removing initialization noise (and gain the insight that this already has substantial influence). Our computation does, however, capture training randomness due to stochastic batch gradient descent. In our discussion, we will simply subsume this under the notion of "initialization noise" for simplicity. Hyperparameters are also still treated as non-random constants.

3.3. Stochastic Integration

The integrals in Eq. (3) and Eq. (4) are extremely high-dimensional; thus we resort to stochastic integration. As most saliency methods provide bounded results (either implicitly or explicitly) the per-pixel variance is also bounded [28]. Thus, the central limit theorem applies and we obtain stochastic convergence of the mean (Eq. (3)) at a rate of $\mathcal{O}(1/\sqrt{n})$ when averaging over n independently initialized and trained models, as visually confirmed by experiments (Fig. 3). More efficient integration methods [37] exist, but we restrict our experiments to simple averaging to demonstrate the effect while minimizing potential error sources.

3.4. Signal to Noise Ratio

In order to quantify variability due to initialization noise, we compute a *signal to noise ratio* (SNR), given by.

$$SNR = \frac{||\mu||_2}{||\sigma||_2}, \tag{5}$$

where $||x||_2 := \sqrt{\sum_i x_i^2}$ denotes the standard ℓ_2-norm for vectors, i.e., we divide the norm of μ by the total standard deviation over all pixels, respectively.

The SNR does not reveal the structure of the noise; a high SNR could hypothetically also be caused by high-frequency fluctuations that do not lead to qualitatively different judgments. Therefore, we verifying our results using a second metric, the established *Structural Similarity Index (SSI)* [36], which yields qualitatively similar results. Since there is (still) no generic metric to capture human perception comprehensively [41], we also perform a visual examination to confirm the potential for variability in structure.

4. Experiments

We now assess the influence of training randomness on saliency maps experimentally. All experiments have been conducted on a single commodity PC equipped with an AMD Ryzen 9 5900X CPU, 128GB of RAM and an Nvidia GeForce RTX 3090 GPU. Deep networks have been implemented using PyTorch.

As *data sets*, we consider *CIFAR10* [16], *FashionMNIST* [39] and "*Imagenette*" [12] at a resolution of 128 × 128.

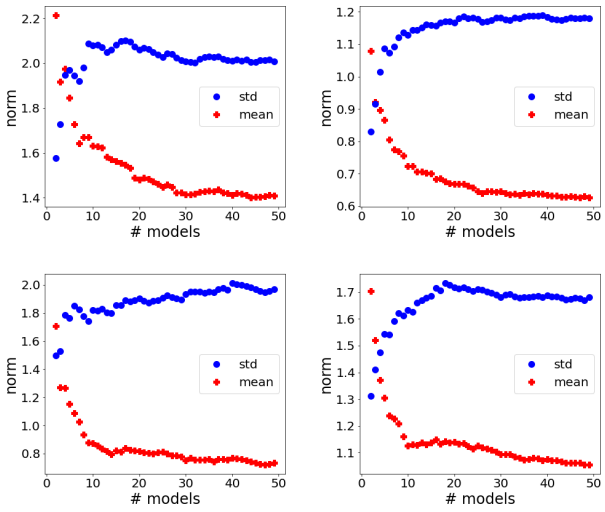


Figure 3. Examples for experimentally calculated approximations for μ and σ (gradient maps from Fig. 1). Increasing the number of models quickly leads to a stable estimate for both.

The latter is a subset of 10 distinct classes from the Imagenet dataset. As our experiments require a large number of models to be trained, we chose Imagenette as a compromise between complexity and computational cost.

In terms of networks, we restrict ourselves to *VGG19* [31], *ResNet18* and *ResNet101* [10] as popular feed-forward CNN architectures, as well as a simple custom CNN-design for testing the influence of width and depth. In all three cases, we use the standard implementation from torchvision; for VGG19 on Imagenette we replace ReLUs with softplus to examine the effect of alternative activation functions.

4.1. Analysis of Input Gradients

First, we investigate the behavior of input gradients, i.e.,

$$S_{f_\theta}(x) := \nabla_x f_\theta(x) \tag{6}$$

As function f, we consider the full network up to the logits; we omit the final softmax-layer, as it can be easily seen that the input gradients vanish at class-label output probabilities of zero or one, which renders the probability gradients even less suitable for saliency detection. To prove this, let $\sigma(z)_i = {e^{z_i}}/{\sum_j e^{z_j}}$ be the softmax-output with $z_i := f_\theta(x)_i$ denoting the i-th logit. Following the chain rule, the input gradient for the class-label output is then given by

$$\nabla_x \sigma(z)_i = \nabla_x \left[\frac{e^{z_i}}{\sum_j e^{z_j}} \right] \tag{7}$$

$$= \sigma(z)_i \left[\nabla_x z_i - \sum_j \sigma(z)_j \nabla_x z_j \right]. \tag{8}$$

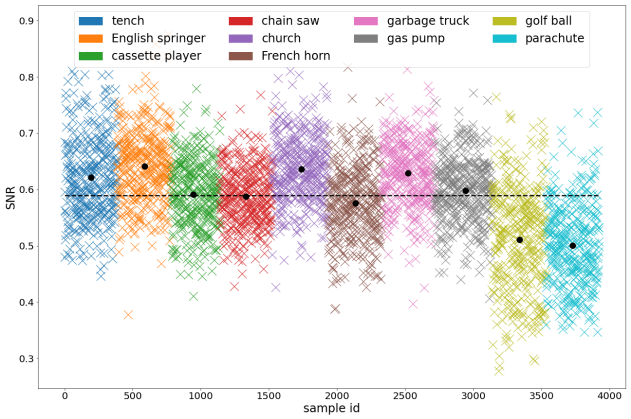


Figure 4. SNR of Imagenette classes. The values are produced using ResNet18. Each colored dot corresponds to a single sample. The dotted line describes the overall average SNR.

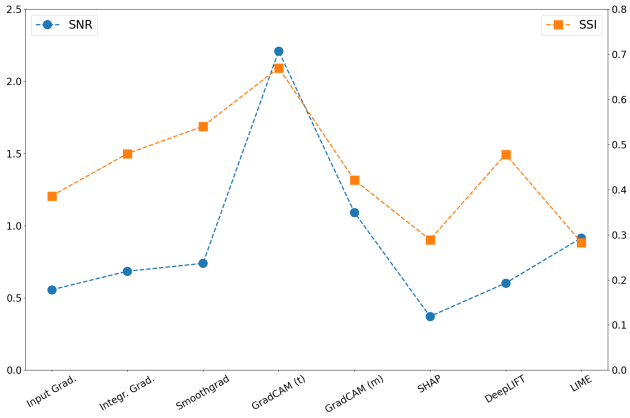


Figure 5. SSI values show a qualitatively similar behavior to SNR, with LIME deviating a bit to the worse (computed for the four example images from Fig. 1.)

As the class-label output probability approaches zero or one, Eq. (8) converges towards zero. We compute the logit-gradient analytically using the built-in automatic differentiation of pytorch.

Gradients are not only an early saliency method but are also often used for empirical studies of deep networks as well as as a basis for more elaborate methods [2]. Therefore, their behavior is of particular interest.

We train 30 ResNet18 models with random He initialization up to a validation accuracy of 87.4 - 89.1 %. For training, we use ADAM-Optimizer [15] and One-Cycle-LR-Scheduler [33] with a maximum learning rate of 0.01. We chose this combination for its fast convergence and good generalization performance, exceeding for example simple step-decay+SGD by a substantial margin at roughly one-tenth of the cost.

Signal-to-noise ratios: Tab. 1 provides mean SNR-values over the validation part of the datasets based on 30 models. We consistently obtain values below one (in the range of $0.44...0.59$), i.e., the noise is more prominent than the actual saliency signal, roughly a factor of two.

Fig. 3 displays the norm of the mean ($||\mu||_2$, red/lower curve) and standard deviation ($||\sigma||_2$, blue/upper curve) over an increasing number of models. The four plots correspond to the samples in Fig. 1. Both the norms of mean and standard deviation appear to converge already after averaging a small number of models. The typic drop in the norm of the mean indicates the presence of uncorrelated noise.

Fig. 4 illustrates the SNR for all images of the validation set of Imagenette over 30 ResNet18-models. The 10 different classes are plotted in different colors. The mean SNR over all images is approximately 0.59, with noise being 70% larger than the signal (indicated by the black dashed line). The SNR varies for different classes; however, an analysis of the softmax predictions shows that there is no correlation between the quality of the prediction and the SNR. Repeating this experiment with VGG19 and ResNet101 shows a similar pattern of the SNR for different classes but at an even worse (lower) level, which might already suggest that deeper architectures tend to be noisier.

Qualitative comparison: Fig. 1 shows input gradients for four random images. The first column displays the input image, the second column shows the mean gradient, and the last two columns show the gradients of two single models, picked manually for variability. As one can see, the specific input gradients for different models differ significantly from each other and from the mean; in some cases (e.g., golf ball in row three), the attributed area has almost no overlap and differs drastically from the mean.

Fig. 5 compare SSI and SNR for different saliency methods calculated on the four examples of Fig. 1. The ranking by SNR and SSI are qualitatively similar (with a deviation to the worse for LIME) so we confine ourselves to SNR in following experiments.

Discussion: Marginalization removes the variability, but it is important to stress that the value of input gradient as a saliency method is still limited due to its well-known conceptual issue of often not detecting key features of a class [2, 35]). Our results, however, caution against the treatment of raw gradients as properties of the data examined. A substantial amount of the information depicted originates from the initialization, not the data.

Training protocol: Having trained the networks with "superconvergent" one-cycle might raise the concern that gradient noise might be an artifact of accelerated training. To eliminate the concern of the results being caused by numerical issues of the chosen training scheme, we repeat the experiment, training additional 30 instances of a ResNet18 on Imagenette with stochastic gradient descent with a learning rate of 0.01 for 30 epochs and a step decay of 0.5 after

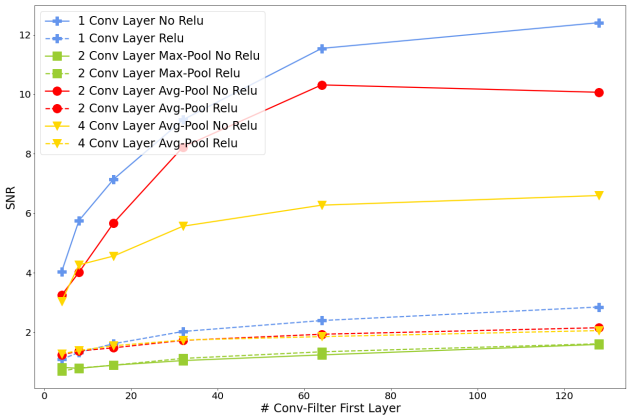


Figure 6. SNR for different architectures. Increasing the width tends to increase the SNR, while adding depth and sources of non-linearity tend to decrease SNR.

epochs 15 and 25. The resulting networks perform a little worse than networks trained with "superconvergence" but we obtain a similar SNR value (0.79), so it is unlikely that the seen results are due to numerical issues specific to ADAM or one-cycle.

Optimization vs. initialization: Training noise originates from random initialization and random choices (batch selection) during optimization. In order to measure their contribution, we train an ensemble of ResNet18 models with fixed initialization for each instance. This results in an improved SNR compared to varying initialization (average SNR varying: 0.694 vs. average SNR fixed: 0.830). However, a visual examination of the saliency maps reveals that even with fixed initialization, there are variations in feature localization. This suggest that both noise in initialization and optimization contributes to the variability we observe.

4.2. Influence of Architectural Parameters

Next, we study the influence of depth, width and amount of nonlinearity on the noise level. Previous experiments have already suggested that depth increases initialization noise. For a systematic study, we run experiments on the FashionMNIST dataset as its low complexity allows comparing many architectures at an acceptable computational cost. The dataset contains 48.000 training images and 12.000 validation images divided into 10 different classes of cloth.

We vary width and depth, and toggle sources of nonlinearity: Starting from a baseline model of a single 3×3 convolution layer followed by a linear classifier, we add more layers and/or increase the number of feature channels in the convolutional filters. We also compare networks with and without max-pooling, and with and without ReLU layers.

Width is varied from 4 to 128 filters, while depth varies from a single to 4 convolutional layers. For every combination of these parameters, we train 20 networks with random initialization and calculate the mean SNR over the complete validation set. Results are shown in Fig. 6. The number of channels (width) used in the first layer is plotted on the x-axis, while the y-axis indicates the mean SNR over all validation samples. The solid lines refer to linear networks without ReLU activation, while the dashed lines correspond to networks with ReLU activation.

	VGG19	ResNet18	ResNet101
Input Gradient	0.436	0.555	0.435
Integrated Gradient*	0.612	0.742	0.744
Smoothgrad**	0.627	0.789	0.878
GradCAM (top)	0.816	1.593	1.373
GradCAM (interm.)	0.612	1.112	0.680
SHAP	0.252	0.365	0.308
DeepLIFT	0.453	0.628	0.554
LIME**	0.609	0.772	0.826

(a) CIFAR10

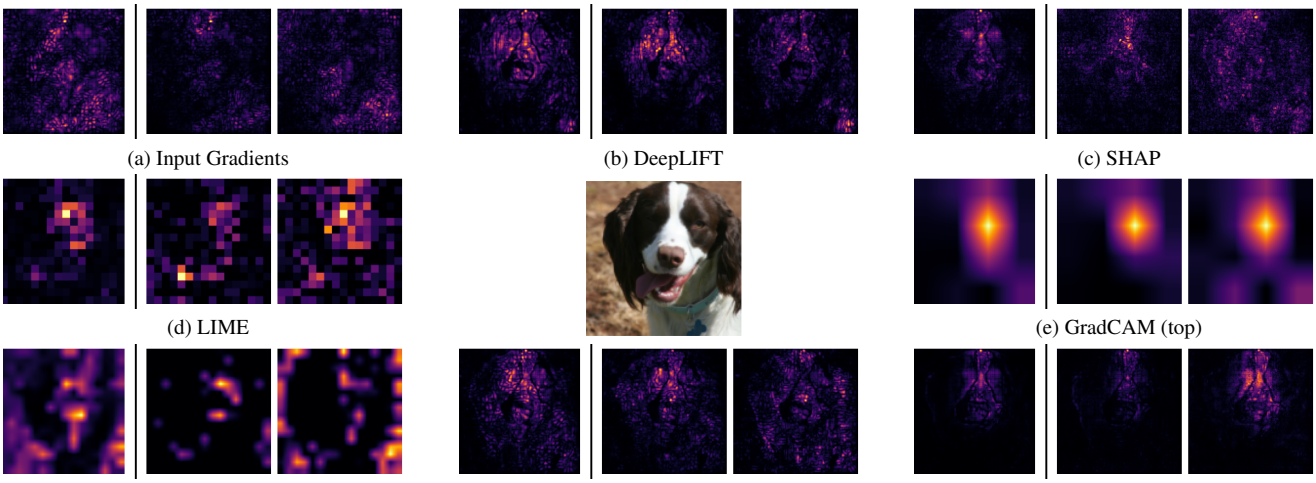
	VGG19	ResNet18	ResNet101
Input Gradient	0.585	0.589	0.444
Integrated Gradient*	0.725	0.695	0.577
Smoothgrad**	0.724	0.681	0.470
GradCAM (top)	1.370	2.363	3.171
GradCAM (interm.)	0.748	1.171	1.253
SHAP	0.418	0.386	0.327
DeepLIFT	0.978	0.650	0.587
LIME**	0.919	0.997	0.982

(b) Imagenette

Table 1. Mean SNR over validation set for different saliency methods and model architectures. Due to increased computational costs we use only a subset for methods marked with "*" (10% of the validation set) and "**" (50 random samples of validation set).

Width and depth: In each architecture, increasing the width w improves SNR, visually consistent with a $\sqrt{w}$-increase due to the averaging of multiple computation paths. Depth decreases SNR consistently, as already expected from the previous experiments.

Linear vs. nonlinear: The most prominent result is the significant drop in SNR once nonlinearities are introduced (with max-pooling and ReLUs both showing similar effects). The convex 1-layer and 2-layer with averaged pooling without non-linearity show the best SNR by a large margin. Interestingly, improvements are also visible for the convex networks, indicating that some initialization noise due to imperfect optimization might still be present, although at a much lower level than in nonlinear architectures. Our observations show that both nonconvexity and nonlinearity increase training noise (including non-convex [17] linear multi-layer networks, solid-yellow-triangle curve). The results are in principle consistent with both an amplification of numerical noise as well as convergence to different local minima as the main source of noise.



(a) Input Gradients (b) DeepLIFT (c) SHAP

(d) LIME (e) GradCAM (top)

(f) GradCAM (intermediate) (g) Integrated Gradients (h) SmoothGrad

Figure 7. Comparison of Saliency Maps for multiple Saliency Methods on differently initialized models. All methods show variability in the produced results. First column: mean over 30 models, column 2-3: single model with random initialization.

Activation functions: One might conjecture that non-smooth activations might be a source of noise. For this reason, we employ softmax in the VGG19-results on Imagenette in Tab. 1, with results not differing qualitatively from ReLU, indicating that results hold for alternative, sufficiently nonlinear activation functions.

Normalization methods: Normalization layers are crucial for stabilizing generalization performance in deep learning. To assess their impact on noise levels, we conduct experiments using two ensembles of ResNet18 models: one with batch normalization (BN) and the other with instance normalization (IN). The results show that the alternative normalization method have only minimal differences in performance (average SNR of BN: 0.694 vs. IN: 0.702).

4.3. Further Saliency Methods

So far, we have only studied plain image gradients of logits. While conceptually important, gradients are not very suitable for attributing network decisions. We thus extend our experiments to several popular saliency approaches that operate by perturbations that resemble augmented gradient or finite difference computations. Specifically, we run *DeepLIFT*, *SHAP*, *LIME*, *Integrated Gradients*, *GradCAM* and *SmoothGrad* on top of *Integrated Gradients*. For this experiment, we use ensembles of 30 models for estimating mean and variance.

Signal-to-noise ratios: Tab. 1 shows the resulting SNR values on *Imagenette* and *CIFAR10* for different architectures. Strikingly, only GradCAM was able to achieve an SNR above 1. Lower layer GradCAMs (tested on 2nd lowest ResBlock-layer of four) fare, expectedly, worse than top-level visualizations.

On CIFAR10, Residual Networks seem to consistently generate better SNR values than VGG19, but this pattern does not hold on Imagenette. Notably SHAP, which has a strong theoretical justification, performs the worst. This might not be surprising, as our results do not imply that any of these methods is a bad saliency method, but that saliency methods in general tend to capture information about the model, and not the phenomenon modelled. A high specificity to the most relevant features exploited by the model might therefore plausibly increase the visibility of the influence of initialization.

Qualitative comparison: Again, in order to understand the practical implications, we search through example images manually for particularly large variations (as these might lead to misinterpretations). Fig. 7 shows examples of varying saliency maps for different methods and initialization, applied to the dog ("English Springer") image in the center. The first column of every subfigure shows the mean saliency map over 30 models, while the second and third columns display the saliency map of two manually chosen networks (an exhaustive account is provided in the supplementary material). The reduced variability of top-level GradCAM is clearly visible, but even here, different initializations might lead to attributions distinct enough that they might conduct to inconsistent interpretations (in particular, taking into account the low resolution of the output). Intermediate-level GradCAM and the popular SHAP and LIME show strong structural differences. For all of the methods, viewing several different saliency maps in progression (as shown in the video accompanying this paper), suggests that variability is in part due to the classifier focusing on different features associated with the class at hand, combined with background noise in varying degrees.

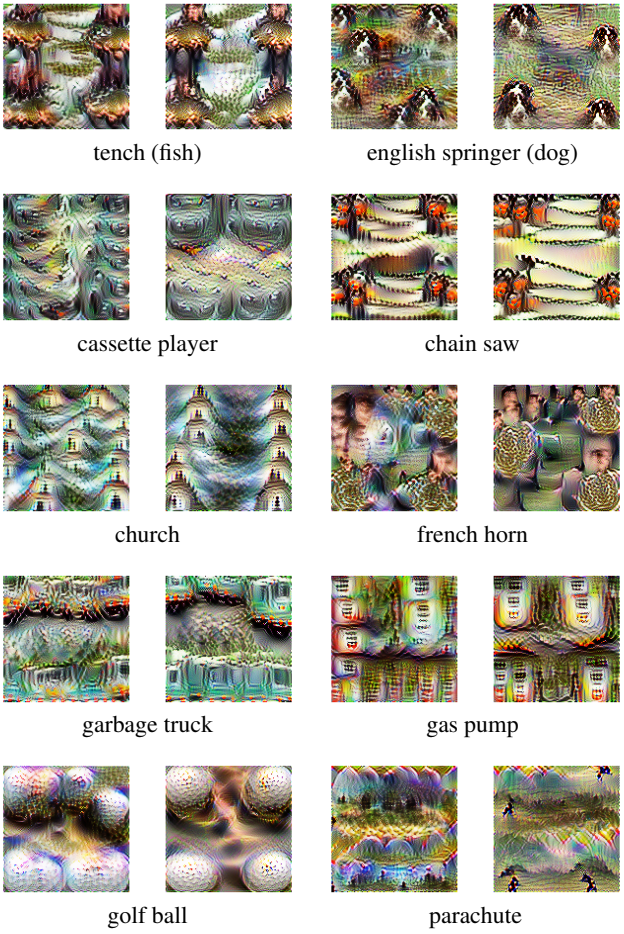


Figure 8. Ensemble Dreams: Images optimized for maximizing class-logit activations appear to capture typical features of the class in a more complete way when using marginalized model gradients (right image) than single model results (left image).

4.4. Feature Visualization by Optimization

In a final experiment, we apply our method to feature visualization by optimization. The well-known "deep dream" / "inceptionism" approach [23] can be used to create typical input images that trigger the activation of neurons in a network. We augment this approach towards an "Ensemble Dream" method that uses gradients from an ensemble of independently trained networks. Specifically, we maximize output logits of an ensemble of 20 ResNet18 models and average their gradients during optimization. We employ the lucent library with all baseline parameters. Results are shown in Fig. 8: While ensembling does not lead to qualitatively different visualizations, the results are often cleaner and appear to show more complete features specific to the class detected by the network.

5. Discussion

Assessing our overall findings, it is very important to state that this paper at its core shows a negative result: Saliency maps based on gradients as well as popular more sophisticated attribution methods vary substantially with training randomness (initialization, random batch choices). Consequently, it cautions the reader to draw conclusions about the nature of the phenomenon observed from single model instances only, as some of the structures obtained originate in initialization randomness rather than uniquely reflecting training data properties. Also, the observed variability does not imply that attribution results are wrong, but we can be sure that they must be at least incomplete at times.

Importantly, one should also not make the converse conclusion: While marginalization is able to reliably dampen training noise to obtain clearer signals[1] this eliminates only one but not all potential extrinsic factors (such as architectural limitations and hyperparameters), and cannot address conceptual limitations such as the limited sensitivity of gradient-based saliency maps.

In a non-rigorous sense, the results obtained from ensemble dreams (Fig. 8) provide an intuitive analogy of our main results: single networks appear to model features in a less complete way than a set of models drawn from different initializations. However, the visualization of the marginal model still shows only a limited understanding of the phenomenon at hand.

If marginalization is not possible (for example due to the increased training costs), the (popular) top-level Grad-CAM results appear to be least affected by training noise. Nonetheless, the lower-level variability we have observed might still be problematic in certain critical applications.

Limitations & Future work: Our paper studies only a limited range of data sets and architectures, and counter-examples in Fig. 1 and 6 are manually curated. As these serve primarily as counter-examples, we consider this nonetheless sufficient to show the presence of a potential problem. However, a study of a broader class of attribution methods (such as masking-based and attention-based methods) as well as strongly divergent architectures (such as vision transformers [6]) is still subject to future work.

The employed marginalization technique is rather inefficient; more sophisticated schemes for sampling network ensembles exist already in the literature [37]. Our approach has been motivated by simplicity and elimination of hidden dependency, not practical efficiency.

A broader study of external randomness due to choices of architectures and hyperparameters would also be an interesting direction for future research, aiming at drawing attribution conclusions in a more automated fashion.

[1] Averaging 50 networks yields a 7-fold and 30 networks a 5.5-fold increase in SNR by the $\sqrt{n}^{-1}$-law, pushing all SNR values well above 1.0.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018. 2, 3

[2] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. *Gradient-Based Attribution Methods*, pages 169–191. Springer International Publishing, Cham, 2019. 2, 5

[3] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 782–791, 2021. 2

[4] Roberto Confalonieri, Ludovik Coba, Benedikt Wagner, and Tarek R. Besold. A historical perspective of explainable artificial intelligence. *WIREs Data Mining and Knowledge Discovery*, 11(1):e1391, 2021. 1

[5] Ann-Kathrin Dombrowski, Maximilian Alber, Christopher J. Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame, 2019. 3

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2, 8

[7] Cong Fang, Yihong Gu, Weizhong Zhang, and Tong Zhang. Convex formulation of overparameterized deep neural networks. *IEEE Transactions on Information Theory*, 68(8):5340–5352, 2022. 2

[8] Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3449–3457, 2017. 2

[9] Timo Freiesleben, Gunnar König, Christoph Molnar, and Alvaro Tejero-Cantero. Scientific inference with interpretable machine learning: Analyzing models to learn about real-world phenomena. *arXiv preprint arXiv:2206.05487*, 2022. 1, 2, 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 4

[11] Andreas Holzinger, Anna Saranti, Christoph Molnar, Przemyslaw Biecek, and Wojciech Samek. *Explainable AI Methods - A Brief Overview*, pages 13–38. Springer International Publishing, Cham, 2022. 1

[12] Jeremy Howard. Imagenette, https://github.com/fastai/imagenette/. 1, 4

[13] Saeed Khorram, Tyler Lawson, and Li Fuxin. Igos++: Integrated gradient optimized saliency by bilateral perturbations. In *Proceedings of the Conference on Health, Inference, and Learning*, CHIL '21, page 174–182, New York, NY, USA, 2021. Association for Computing Machinery. 2

[14] Beomsu Kim, Junghoon Seo, Seunghyun Jeon, Jamyoung Koo, Jeongyeol Choe, and Taegyun Jeon. Why are saliency maps noisy? cause of and solution to noisy saliency maps. *CoRR*, abs/1902.04893, 2019. 2

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. 5

[16] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. 4

[17] Thomas Laurent and James von Brecht. Deep linear networks with arbitrary loss: All local minima are global. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2902–2907. PMLR, 10–15 Jul 2018. 6

[18] Zachary C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, jun 2018. 2

[19] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. On the linearity of large non-linear models: when and why the tangent kernel is constant, 2020. 2

[20] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017. 2

[21] Christoph Molnar, Timo Freiesleben, Gunnar König, Giuseppe Casalicchio, Marvin N. Wright, and Bernd Bischl. Relating the partial dependence plot and permutation feature importance to the data generating process, 2021. 2

[22] Christoph Molnar, Gunnar König, Julia Herbinger, Timo Freiesleben, Susanne Dandl, Christian A. Scholbeck, Giuseppe Casalicchio, Moritz Grosse-Wentrup, and Bernd Bischl. *General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models*, pages 39–68. Springer International Publishing, Cham, 2022. 2

[23] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization. 2, 8

[24] Daniel Omeiza, Skyler Speakman, Celia Cintas, and Komminist Weldemariam. Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. *CoRR*, abs/1908.01224, 2019. 2

[25] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016. 1, 2

[26] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, oct 2019. 1, 2

[27] Arash Shaban-Nejad, Martin Michalowski, John S. Brownstein, and David L. Buckeridge. Guest editorial explainable ai: Towards fairness, accountability, transparency and trust in healthcare. *IEEE Journal of Biomedical and Health Informatics*, 25(7):2374–2375, 2021. 1

[28] Rajesh Sharma, M. Gupta, and G. Kapoor. Some better bounds on the variance with applications. *Journal of Mathematical Inequalities*, (3):355–363, 2010. 4

[29] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. 2017. 1, 2

[30] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In Yoshua Bengio

and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014. 1, 2

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. 4

[32] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017. 1, 2

[33] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates, 2017. 5

[34] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015. 2

[35] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017. 1, 2, 5

[36] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 4

[37] Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NeurIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. 2, 3, 4, 8

[38] Ann-Christin Woerl, Markus Eckstein, Josephine Geiger, Daniel-Christoph Wagner, Tamas Daher, Philipp Stenzel, Aurélie Fernandez, Arndt Hartmann, Michael Wand, Wilfried Roth, and Sebastian Foersch. Deep learning predicts molecular subtype of muscle-invasive bladder cancer from conventional histopathological slides. *European Urology*, 78, 04 2020. 1

[39] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. 4

[40] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing. 2

[41] Guangtao Zhai and Xiongkuo Min. Perceptual image quality assessment: a survey. *Science China Information Sciences*, 63(11):1–52, 2020. 4