Statistically Consistent Saliency Estimation

Shunyan Luo
George Washington University
shine_lsy@gwu.edu

Emre Barut
Amazon Alexa
ebarut@amazon.com

Fang Jin
George Washington University
fangjin@email.gwu.edu

Abstract

The growing use of deep learning for a wide range of data problems has highlighted the need to understand and diagnose these models appropriately, making deep learning interpretation techniques an essential tool for data analysts. The numerous model interpretation methods proposed in recent years are generally based on heuristics, with little or no theoretical guarantees. Here, we present a statistical framework for saliency estimation for black-box computer vision models. Our proposed model-agnostic estimation procedure, which is statistically consistent and capable of passing sanity checks, has polynomial-time computational efficiency since it only requires solving a linear program. An upper bound is established on the number of model evaluations needed to recover regions of importance with high probability through our theoretical analysis. Furthermore, a new perturbation scheme is presented for the estimation of local gradients that is more efficient than commonly used random perturbation schemes. The validity and excellence of our new method are demonstrated experimentally via sensitivity analyses.

1. Introduction

Although deep learning models have achieved exceptional predictive performance for many tasks, these complex and often intractable models can be difficult to interpret and understand. This is a major barrier hindering their wider adoption, especially in domains such as medicine where models need to be qualitatively understood and/or verified for robustness.

In order to address these issues, a number of interpretation approaches have been proposed, many of which are based on visualizations that either quantify the effect of particular neurons or features, or create new images that maximize the target score for specific classes [13, 26, 33]. One popular approach is to build saliency maps by attributing the gradients of the neural network to the input image through various procedures, or by finding perturbations that significantly change the output[2, 4, 12, 15, 19, 24, 25, 28, 29, 30, 34]. Another option is to treat the deep learner as a black-box. Examples in this domain include Baehrens, *et al*. [5] who use a Parzen window classifier to approximate the target classifier locally, and Riberio *et al*. [23], who introduce the LIME procedure that relies on a sparse linear model that is fit to model predictions of perturbed inputs. Lundberg and Lee [17] have proposed SHapley Additive exPlanation (SHAP), which combines the Shapley value from game theory with additive feature attribution methods, highlighting the connections between the SHAP procedure and existing methods such as LRP, LIME and DeepLIFT. Similarly, Chen *et al*. [10] have built L- and C-Shapley procedures that reliably approximate the Shapley values in linear time with respect to the number of features.

The majority of the methods listed above are heuristics constructed according to certain desirable qualities. However, in none of these methods, is it clear what the main estimand is, whether it can be consistently estimated or if (and how) the estimand can be computed more efficiently. In fact, according to recent research by Adebayo *et al*. [1], most methods with good visual inspection lack sensitivity to the model and the data generating process, a theoretical explanation for why guided back-propagation and deconvolutional methods perform image recovery is provided by Nie *et al*. [20]. These findings remind us of the importance of constructing saliency estimation methods that are founded on solid theoretical guarantees. This motivation is not straightforward; recent work by Burns *et al*. [7] propose a saliency estimation technique that includes theoretical guarantees based on the false discovery rate, i.e. FDR control. Although their procedure is very promising from a statistical perspective and theoretically valid under a very general set of assumptions, it requires human input and incurs a significant computational load as it uses a generative model to fill in certain regions of the target image.

In this work, we propose a statistically valid technique for model-agnostic saliency estimation, and prove its consistency under reasonable assumptions. Furthermore, our method passes the sanity checks given by Adebayo *et al*. [1]. Our analysis provides valuable insights into possible ways to improve the accuracy and reliability of our approach. Our main contributions are as follows:

- We introduce a new and innovative saliency estimation framework for CNNs and propose a new local explanation method based on input perturbation. Our procedure only requires solving a linear program, and hence the estimates can be computed very efficiently. Furthermore, the optimization can be recast as a "parametric simplex" problem [31], which allows the computation of the full solution path in an expedient manner.
- We establish the conditions under which the significant pixels in the input can be identified with high probability and present finite-sample convergence rates that can be used to determine the number of necessary model evaluations.
- We determine that the noise distribution for the perturbation has a substantial effect on the convergence rate and propose a new perturbation scheme that uses a highly correlated Gaussian, rather than the widely used independent Gaussian distribution.

We present our notation in the next Section. We define the saliency parameter of interest (i.e. the estimand), the linearly estimated gradient (LEG), and introduce our new statistical framework in Section 3. In Section 4, we propose a regularized estimation procedure for LEG that penalizes the anisotropic total-variation. Our theoretical results are provided in Section 5 and the results of our numerical comparisons are shown in Section 6.

2. Notation

For a matrix B, we use $\mathrm{vec}(B)$ and $\mathrm{vec}^{-1}(B)$ to denote its vectorization and inverse vectorization, respectively. The transpose of a matrix B is given by B^T and we use B^+ for its pseudo-inverse . The largest and smallest eigenvalue of a symmetric matrix B are denoted by $\lambda_{\max}(B)$ and $\lambda_{\min}(B)$. For a set S, we use S^C to denote its complement. For a vector $u \in \mathbb{R}^p$ and a set $S \subseteq [1, \dots, p]$, we use u_S to refer to its components indexed by elements in S. The q-norm for a vector u is given by $\|u\|_q$ and we use $\|B\|_{Fr}$ for the Frobenius norm of a matrix B. The vector of size p whose values are all equal to 1 is denoted by 1_p. Similarly, we use $1_{p_1 \times p_2}$ and $0_{p_1 \times p_2}$ to denote a $p_1 \times p_2$ matrix whose entries are equal to 1 and 0, respectively. Finally, for a continuous distribution F, we use $F + x_0$ to denote a distribution that is mean-shifted by x_0, i.e. $F(z) = G(z - x_0)$ for all z, where $G = F + x_0$.

3. Linearly Estimated Gradient

In gradient-based saliency approaches, the main goal is to recover the gradient of the deep learner with respect to the input. More specifically, let $f(x)$ be a deep learner, $f : \mathcal{X} \to [0, 1]$, where $\mathcal{X}$ is the input space. For instance, for the MNIST dataset that contains 28 by 28 sized images of hand-written digits from 0 to 9, $\mathcal{X} = [0, 255]^{28\times 28}$. We assume that the model output is the probability for a specific class, e.g., $f(x) = P_{model}(x \text{ is a } 9)$. However, this can be modified to check for comparative quantities by setting the output $f(x)$ equal the difference of two class probabilities, that is by writing

$$f(x) := f_9(x) - f_7(x) = P_{model}(x \text{ is a } 9) - P_{model}(x \text{ is a } 7). \tag{1}$$

Then, local saliency is defined as the derivative of $f(\cdot)$ with respect to the input, evaluated at a point of interest $x_0 \in \mathcal{X}$, i.e. $\nabla_x f(x)|_{x=x_0}$. However, in practice, local saliency is often too noisy and one instead uses an average of the gradient around x_0 [25, 29].

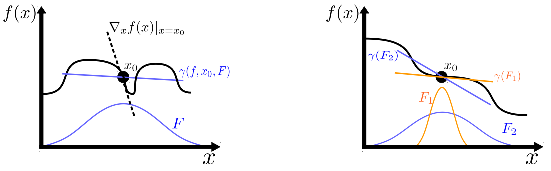
In order to study the saliency procedure from a statistical perspective, we start by defining an estimand, whose definition is motivated by the LIME procedure [23].

Definition 1 (LEG)**.** *For a continuous distribution F, an initial point $x_0 \in \mathcal{X}$ with $\mathcal{X} \subset \mathbb{R}^{p_1\times p_2}$, and a function $f : \mathcal{X} \to [-1, 1]$, the linearly estimated gradient (LEG), $\gamma(f, x_0, F) \in \mathbb{R}^{p_1\times p_2}$ is given by*

$$\gamma(f, x_0, F) = \arg\min_g \mathbb{E}_{x\sim F+x_0} \Big[\big(f(x) - f(x_0) - \mathit{vec}(g)^T \mathit{vec}(x_0 - x)\big)^2\Big]. \tag{2}$$

LEG is based on a first-order Taylor series expansion of the function $f(x)$ around the point of interest x_0. The estimand is a proxy for the local gradient, and is the coefficient that gives the best linear approximation, in terms of the squared error, among all possible choices. The distribution F determines the range of points the analyst wants to consider. We visually demonstrate LEG on two toy examples with a single pixel (i.e. $p_1 = p_2 = 1$) in Figure 1. When the perturbation is taken to be a Gaussian distribution with independent entries, LEG behaves similar to SmoothGrad [29] which uses the average saliency score of multiple images that are generated by adding random perturbations to the initial image. LEG does not rely on saliency scores, which require full knowledge about the underlying deep learner, and instead finds the best linear approximation evaluated for possible perturbations. Thus, if the underlying function, $f(x)$, is linear over the neighborhood around x_0, then the SmoothGrad estimand and LEG would be exactly the same. Yeh *et al.* [32] proposed a generalized metric called infidelity to unify existing explanations including SmoothGrad. In this respective, LEG defines a valid and novel infidelity measure for black box models.

Furthermore, LEG belongs to the class of model interpretation techniques that rely on local smoothing. Interpretation methods in this class are known to be more reliable against adversarial manipulations [11], more faithful to the model [32], tend to pass sanity checks [1], and perform better on



(a) Gradient vs. LEG (b) Effect of F on LEG

Figure 1: Visual demonstrations of LEG for a single input. LEG seeks to find a local linear approximation of $f(x)$ in a neighborhood around x_0; choice of the distribution, F, determines the size of the neighborhood. In Figure 1a, we compare LEG to the gradient, which is very localized. If $f(x)$ is a highly varying function, then the gradient is too noisy, and the saliency score provided by LEG is more meaningful. In Figure 1b, we show LEG for two different distributions. For the distribution with a larger variance, LEG evaluates the input's effect on the output for a larger neighborhood around x_0.

benchmarks that measure the accuracy change after removal of relevant pixels [16].

We note that the variance of F has a large effect on LEG. As F converges to a point mass at 0, if $f(x)$ is twice continuously differentiable in the neighborhood of x_0, then $\gamma \to \nabla_x f(x)$. On the other hand, if F has high variance, then samples from $F + x_0$ are substantially different from x_0 and LEG might no longer be useful for interpreting the model at x_0. However, with some assumption on the distribution F, LEG has an analytical solution as the next lemma shows.

Lemma 1. *Let Z be the random variable with a centered distribution F, i.e., $Z \sim F$ and $\mathbb{E}[Z] = 0_{p_1 \times p_2}$. Assume that covariance of $vec(Z)$ exists, and is positive-definite. Let $\Sigma = Cov(vec(Z))$, then*

$$\gamma(f, x_0, F) = \\ vec^{-1}\left(\Sigma^{-1}\mathbb{E}_{z\sim F}[(f(x_0+z) - f(x_0))\, vec(z)]\right). \quad (3)$$

Proof of the lemma is provided in the Appendix.

Lemma 1 shows that the LEG can be written as an affine transformation of a high dimensional integral where the integrand is $\int (f(x_0+z) - f(x_0))\, z dF(z)$. This analysis also suggests an empirical estimate for the LEG, by replacing the expectation with the empirical mean. The empirical mean can be obtained by sampling x from $F + x_0$, calculating $f(x)$, and then applying Lemma 1. More formally, let $x_1, \ldots, x_n$ be random samples from $F + x_0$, and let $y_1, \ldots, y_n$ be the function evaluations with $y_i = f(x_i)$. Further, let $\tilde{y}_i = f(x_i) - f(x_0)$ and $z_i = x_i - x_0$. Then, the empirical LEG estimate is given by

$$\hat{\gamma}(f, x_0, F) = \text{vec}^{-1}\left(\Sigma^{-1}\left[\frac{1}{n}\sum_{i=1}^{n}\text{vec}\left(\tilde{y}_i z_i\right)\right]\right). \quad (4)$$

As the function $f(x)$ is bounded and F has a positive-definite covariance matrix, then it follows that as $n \to \infty$, $\hat{\gamma} \to \gamma$. However, classical linear model theory [22] shows that rate of the convergence is very slow, on the order of $\frac{1}{\lambda_{\min}(\Sigma)}\sqrt{p_1 p_2 / n}$, where p_1 and p_2 are the dimensions of $\mathcal{X}$. This severely limits the practicality of the empirical approach. In the next section, we propose to use regularization in order to obtain faster convergence rate.

4. Efficient Estimation of LEG

For interpretation of image classifiers, one expects that the saliency scores are located at a certain region, i.e., a contiguous body or a union of such bodies. This idea has lead to various procedures that estimate saliency scores by penalizing the local differences of the solution, often utilizing some form of the total variation (TV) penalty [15]. The approach is very sensible from a practical point of view: Firstly, it produces estimates that are easy to interpret as the important regions can be easily identified; secondly, penalization significantly shrinks the variance of the estimate and helps produce reliable solutions with fewer model evaluations.

In light of the above, we propose to estimate the LEG coefficient with an anisotropic L_1 TV penalty.

Definition 2 (LEG-TV)**.** *For a hyperparameter, $L \geq 0$, the TV-penalized LEG estimate is given as $\tilde{\gamma} = vec^{-1}(g)$ where g is the solution of the following linear program*

$$\min_{g} \|Dg\|_1$$
$$s.t. \left\| D^{+T}\left(\frac{1}{n}\sum_{i=1}^{n} vec\left(\tilde{y}_i z_i\right) - \Sigma g\right)\right\|_{\infty} \leq L, \quad (5)$$

where $D \in \mathbb{R}^{(2p_1p_2 - p_1 - p_2)\times(p_1p_2)}$ is the differencing matrix with $D_{i,j} = 1, D_{i,k} = -1$ if the j^{th} and the k^{th} component of g are connected on the two dimensional grid.

Our method is based on the "high confidence set" approach which has been successful in numerous applications in high dimensional statistics [8, 9, 14]. The set of g that satisfy the constraint in the formulation is our high confidence set; if L is chosen properly, this set contains the true LEG coefficient, $\gamma(f, x_0, F)$, with high probability[1]. This setup ensures that the distance between γ and $\tilde{\gamma}$ is small. When combined with the TV penalty in the objective function, the procedure seeks to find a solution that both belongs to the confidence set and has sparse differences on the grid. Thus,

[1] See Lemma 2 in the Appendix.

the estimator is extremely effective at recovering γ that has small total variation. In Figure 2, we show two resulting estimates of the method with $10k$ model evaluations per channel for a VGG-19 [27] network and LEG-TV estimates do give us a sparser explanation and better visualization[2]. For the distribution F, we use a multivariate Gaussian distribution with the proposed perturbation scheme in Section 5.2. We compute $\tilde{\gamma}$ separately for each channel, and then sum the absolute values of the different channels to obtain the final saliency score.

The proposed method enjoys low computational complexity. The problem in equation 5 is a linear program and can be solved in polynomial time, for instance by using a primal-dual interior-point method for which the time complexity is $O\left((p_1p_2)^{3.5}\right)$ [21]. However, in practice, solutions can be obtained much faster using simplex solvers. In our implementations, we use MOSEK, a commercial grade simplex solver by ApS [3], and are able to obtain a solution in less than 3 seconds on a standard 8-core PC for a problem of size $p_1 = p_2 = 28$. Additionally, the alternative formulation (provided in the Appendix) can be solved using parametric simplex approaches which yield the whole solution path in L [31]. In practice, this approach can save substantial computational costs, when L needs to be tuned for best performance according to specific criteria.

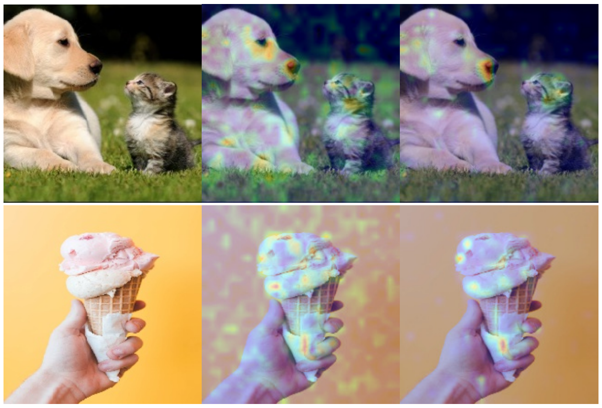
We note that the procedure does not require any knowledge about the underlying neural network and is completely model-agnostic. In fact, in applications where security or privacy could be a concern and returning multiple prediction values needs to be avoided, the term given by $\sum_{i=1}^{n} \text{vec}\left(\tilde{y}_i z_i\right)$ can be computed on the side and supplied alongside the prediction.

5. Theoretical Analysis and Implementation

In this section, we analyze the procedure from a theoretical perspective and derive finite sample convergence rates of the proposed LEG-TV estimator. Our results provides an upper bound on how the error of the estimator changes with respect to the complexity of the true parameter, given by its sparsity in the TV-norm, and the number of input perturbations. These results hold under specific conditions, whose implications we study in Section 5.2. As we noted earlier, this insight is used to derive the ideal perturbation distribution under these conditions.

5.1. Consistency

We first present our condition, which has a major role in the convergence rate of our estimator. The condition is akin to the restricted eigenvalue condition [6] with adjustments specific to our problem.



(a) Origin (b) LEG (c) LEG-TV

Figure 2: LEG estimates for ImageNet images classified by VGG-19. Both approaches select pixels that are critical for the label, such as nose and ear of golden retriever, bottom of cone and scoop of ice-cream. LEG-TV, compared to LEG, provides a more human readable estimate of local saliency.

Assumption 1. *Let D^+ be the pseudo-inverse of the differencing matrix D, and denote the elements of singular value decomposition of D as U, Θ, V where $D = U\Theta V^T$. Furthermore, denote the last $p_1p_2 - p_1 - p_2$ columns of U that correspond to zero singular values as U_2. We define the differencing error as $\Delta = D(\hat{\gamma} - \gamma^\star)$ and Δ_S as the elements of Δ in set S. For the covariance matrix Σ, and any set S with size s, it holds that $\kappa > 0$, where*

$$\kappa = \inf_{\substack{\|\Delta_S\|_1 \geq \|\Delta_{S^C}\|_1 \\ U_2^T\Delta = 0}} \frac{\Delta^T {D^+}^T \Sigma D^+ \Delta}{\|\Delta\|_2^2}. \quad (6)$$

The following theorem is our main result.

Theorem 1. *Let $\gamma^* = \gamma(f, x_0, F)$ and $\Sigma = Cov\left(vec(Z)\right)$, where $Z \sim F$ and $\mathbb{E}[Z] = 0_{p_1 \times p_2}$. Let $\tilde{\gamma}$ be the LEG-TV estimate with $L = \sqrt{2\|D^+\|_1 \log\left(p_1p_2/\epsilon\right)/n}$. If Assumption 1 holds for the covariance matrix Σ with constant κ, then with probability $1 - \epsilon$,*

$$\left\|\gamma^* - \tilde{\gamma} - m 1_{p_1} 1_{p_2}^T\right\|_{Fr}^2 \leq \frac{1}{\kappa}\frac{C_p}{C_d}\sqrt{\frac{s \log p_1p_2/\epsilon}{n}},$$

where $m \in \mathbb{R}$ is a mean shift parameter, s is the number of non-zero elements in $D\gamma^*$, $C_p = 4\sqrt{2\|D^+\|_1} \propto p_1^{1/4}p_2^{1/4}$ and C_d is the minimal positive singular value of D.

The proof uses the "high confidence set" approach of Fan [14]. In the proof, we first establish that, for an appropriately chosen value of L, $\gamma^* = \gamma(f, x_0, F)$ satisfies the constraint in equation 5 with high probability. Then, we make use of TV sparsity of $\tilde{\gamma}$ and γ^* to argue that the two quantities

[2] Please see https://github.com/Paradise1008/LEG for more examples, our source code, and a tutorial on how to create your own LEG estimator

cannot be too far away from each other, since both are in the constraint set. The full proof is provided in the Appendix.

Our theorem has two major implications:

1. We can recover the true parameter as the number of model evaluations increase. That is, TV penalized LEG is a statistically consistent model interpretation scheme. Furthermore, our result states that, ignoring the log terms, one needs $n = O(s\,(p_1 p_2)^{1/2})$ many model evaluations to reliably recover γ^*.

2. Our bound depends on the constant κ, which further depends on the choice of Σ for the perturbation scheme. It is possible to obtain faster rates of convergence with a carefully tuned choice of Σ. As a side note, since γ^* also depends on Σ, the estimand changes when Σ is adjusted. In other words, our result states that certain estimands require fewer samples.

We note that our procedure identifies the LEG coefficient up to a mean shift parameter, m, which is the average of the true LEG coefficient γ. In practice, the average can be consistently estimated (for instance, using the empirical version of LEG in equation 4), and the mean can be subtracted to yield consistent estimates for γ. However, in our numerical studies, we see that this mean shift is almost non-existent: LEG-TV yields solutions that has no mean differences with the LEG coefficient, which we define as the solution of the empirical version as $n \to \infty$.

5.2. Perturbation Scheme

In our main result, we established that the convergence of our estimator depends on the quantity κ which is related to the spectral properties of Σ. In this subsection we explore the ramifications of the assumption.

Our main result in Theorem 1 states that the rate of convergence to the true LEG coefficient is inversely proportional to the term κ. Thus, perturbation schemes for which the restricted eigenvalues are large, as defined in Definition 1, yield saliency maps that require less samples to estimate the LEG. We note that most of the saliency estimation procedures that make use of perturbations take these perturbations to be independent, which results in a covariance matrix that is equal to the identity matrix, $\Sigma = \sigma^2 \mathbb{I}_{(p_1 p_2) \times (p_1 p_2)}$ for some $\sigma^2 > 0$. For LEG estimation without penalization, i.e., using equation 2, this choice is also optimal as the convergence rate for the empirical estimate depends on $1/\lambda_{\min}(\Sigma)$. However, when one seeks to find an estimate for which the solution is sparse in the TV norm, this choice is no longer ideal as demonstrated by our theorem.

In order to choose the covariance matrix of our perturbation scheme in a manner that maximizes the bound in equation 6, one also needs some prior information about the size of S, s. As that requires estimation of s, and a complex optimization procedure, we instead propose a heuristic: we choose Σ so that its eigenvectors match $D^+\Delta$ for vectors Δ with unit-norm and $U_2^T \Delta = 0$. This choice fixes $p_1 p_2 - 1$ many of the eigenvectors of Σ. For the last eigenvector, we use the one vector as it is orthogonal to the rest of the eigenvectors. Our proposed perturbation scheme is as follows:

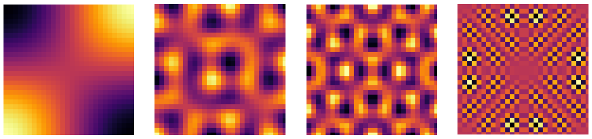


Figure 3: Selected eigenvectors of the proposed Σ. The eigenvectors, which contain the principal directions of the distribution, resemble the basis for 2D Haar wavelets[18].

1. Compute the singular value decomposition of D, and let $D = U\Theta V^T$.

2. Let $\Sigma = \sigma^2 \left(V\Theta^2 V^T + \frac{1}{p_1 p_2} 1_{p_1 p_2} 1_{p_1 p_2}^T \right)$ for some choice of $\sigma^2 > 0$.

As $D^+ = V\Theta^+ U^T$, with the proposed Σ, the numerator in equation 6 reduces to $\sigma^2 \Delta^T \Delta$ and hence $\kappa = \sigma^2$. Without any additional assumptions on S, this is the maximal value for κ. We plot some of the eigenvectors for our proposed Σ with $p_1 = p_2 = 28$ in Figure 3. These eigenvectors are the principal directions of the perturbation distribution F, and the samples drawn from F contain a combination of these directions. We see that samples drawn from this distribution will have sharp contrasts at certain locations. This result is very intuitive: The perturbation scheme is created for a specific problem where boundaries for objects are assumed to exist, and large jumps in the magnitude of the distribution help our method recover these boundaries efficiently. The demonstration of the perturbation scheme using Gaussian noise and its visual comparison with independent perturbation are provided in the Appendix.

5.3. Implementation Details

LEG-TV procedure has two tuning parameters: (i) F, which determines the structure of the perturbation; and (ii) L, which controls the sparsity of the chosen interpretation.

Regarding F, we propose to use a multivariate Gaussian distribution as it is easy to sample from. For Σ, we propose a theoretically driven heuristic for determining the correlation structure of Σ in Section 5.2. However, the choice of the magnitude of Σ, i.e. σ^2, should be chosen discreetly. If this quantity is chosen too low, then the added perturbations are small in magnitude, and the predictions of the neural network do not change, resulting in a LEG near zero. On the other hand, with a very large value of σ^2, the sample images are dominated by extreme pixel intensities which looks like random noise without retaining any information of the original image. Thus it can not be considered belonging to some small neighborhood of the target image. In our implementations, we find that setting σ to be around 0.02 for results in reasonable solutions respectively. We determine this range by computing perturbations of various sizes on numerous images for both experiments. The provided range is found to create perturbations large enough to change the prediction probabilities but small enough to avoid major changes in the image.

For the choice of L, we propose two solutions: The first is the theoretically suggested quantity given in Theorem 1, although this often results in estimates that are too conservative. Our second method is a heuristic based on some of the quantities in the optimization problem and we use this for our demonstrations. We set $L = K_L L_{\max}$ where K is a constant between 0 and 1 and $L_{\max}$ is the smallest value of L for which the solution in equation 5 would result with $g = 0$; i.e. $L_{\max} = n^{-1} \| D^{+T} \left(\sum_{i=1}^{n} \text{vec}(\tilde{y}_i z_i) \right) \|$. We use $K_L = 0.1$ or $K_L = 0.3$ in our implementations. We note that one can obtain solutions for all L by using a parametric simplex solver [31], or by starting with a large initial L, and then using the solution of the program as a warm-start for a smaller choice of L. Both approaches return the solution path for all L, and might be more desirable in practice than relying on heuristics.

6. Experiments

In this section, we compare our procedure with other interpretability techniques. We first present the results of a sensitivity analysis in which the most salient regions according to each interpretation method are masked, and the change in the masked images' classification score is recorded. In this analysis, more effective interpretability techniques are expected to better identify the important regions, and thus images masked according to better approaches should have lower scores. In the second subsection, we run a sanity check, in which we perturb the parameters of the deep learner in a cascading manner starting from the last layer. For this exercise, we follow the setup proposed by Adebayo *et al*. [1], and find that our technique passes the sanity checks - i.e., it fails to provide interpretations for neural networks with randomly chosen parameters.

6.1. Setup

In our analyses, we utilize a pre-trained VGG-19 image classifier trained on ImageNet as the deep learner. We compare our approach with three other popular model agnostic methods, C-Shapley, KernelSHAP, and LIME. In addition to the three techniques, we also provide results for GradCAM as a reference point for a method that requires knowledge about the underlying model. All model agnostic methods are based on 6,000 model evaluations. We take 8x8 patches as the single features for LEG, LEG-TV and C-Shapley, following Yeh *et al*. [32]'s argument that this setting can improve the computational complexity and capture spatial relationship in images. The other two techniques, LIME and KernelSHAP, treat segmentation as "superpixels", and hence cannot utilize large patches.

6.2. Sensitivity analysis

Evaluating explanations is an inescapably subjective task. Sensitivity analyses seek to address this issue by providing a quantitative framework for comparing evaluations and are widely used in contrasting different interpretation techniques. In the sensitivity analysis, first, various interpretation models are used to identify regions of high saliency, and then the identified regions are masked in order of decreasing importance by changing the pixels. Finally, the difference in the score due to the masking is computed via the log-odds which is given as:

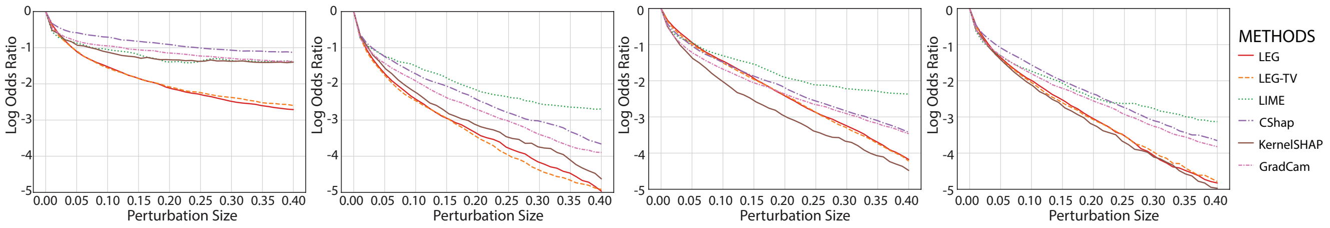
$$LOR = \log \left(\frac{P_c(x')/(1 - P_c(x'))}{P_c(x_0)/(1 - P_c(x_0))} \right),$$

where $P_c(x')$ is the prediction probability of the masked image x' for the class C, and C is the top prediction class of the model for the original image x_0. The previous procedure is repeated by varying the amount of masked regions and finally the change in the log-odds is plotted with respect to the size of the masked region. Methods that can accurately identify salient regions will have a faster reduction in log-odds, and hence an interpretation technique can be said to over-perform one another if the former achieves a faster reduction in log-odds than the latter.

We note that, although the presented sensitivity analysis procedure is commonly employed in the literature, the masking task makes the setup ambiguous: we find that different masking approaches can yield different results. Most masking techniques shift the pixel to a given baseline, although this can mean that inconsistent performance assessments are achieved using different masking techniques. Rather than adopting a single masking technique to carry out the sensitivity analysis for this study, we therefore summarize several meaningful masking techniques, described below, and analyze the results achieved by each in turn to gain a comprehensive understanding of the local explanations.

- Distance-K Masking: Masks the pixel by modifying the intensity by $\pm K$, i.e. $S'_{ij} = S_{ij} \pm K$. The sign of K is determined by the sign of the saliency. If K is large enough, it assigns pixels with positive saliency to black and negative saliency to white.

- Fixed-value Masking: Masks pixels using a fixed color vector such as $(0, 0, 0)$ and $(255, 255, 255)$, which are known named as black-out and white-out, respectively.



(a) Distance-100 (b) Black-out (c) Mean vector (d) Noise masking

Figure 4: Sensitivity results of LEG, LEG-TV, LIME, KernelSHAP, CShap and GradCam with different masking techniques.

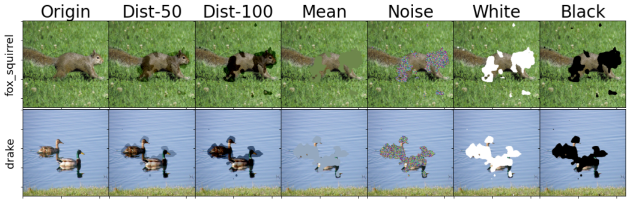


Figure 5: Examples of LEG-TV estimates shown by different masking techniques with 10% masked.

The mean value vector of the image is also applied in most circumstances.

- Noise Masking : Masks the pixel by some random pixel intensity in the range [0,255]. As this randomness may lead to considerable differences between trials, it is best to average these trials to reduce the variance.

Each of these masking techniques can be described as a form of contamination of the original image. Figure 5 presents the contaminated images under various masking techniques. We note that the most commonly utilized masking methods, black-out and white-out will have issues when a large part of the image is black (or white). Distance-K mask darkens the picture with respect to the original pixel intensity, and simulates having different levels of brightness, and thus it can preserve some color and edge patterns. Noise mask eliminates both the color and edge patterns but may not be stable due to the randomization.

We present the results of sensitivity analysis with the four different masking techniques in Figure 4 for 500 images collected randomly from ImageNet's test dataset. LEG and LEG-TV achieve the best performance on Distance-100 and the black-out scheme, while KernelSHAP performs best on mean vector, all deliver excellent performance on noise masking. Note that with Distance-100 masking, as shown in Figure 4(a), KernelSHAP and LIME gain a sharp drop with the initial 2% perturbation, after which they flatten out and are eventually overtaken by LEG. This naturally raises the question of whether it is still possible to compare different explanations when such a crossover occurs. We argue that instead of investigating how fast the prediction for the target class decreases near zero perturbations, an explanation should be able to efficiently discover the minimal amount needed to change the classification result obtained by the method used for supervised learning on computer vision. With that aim, we propose a new metric named the key masking size, denoted by S_{key}, which is defined as the minimum masking size that causes the target model to deliver a different classification result.

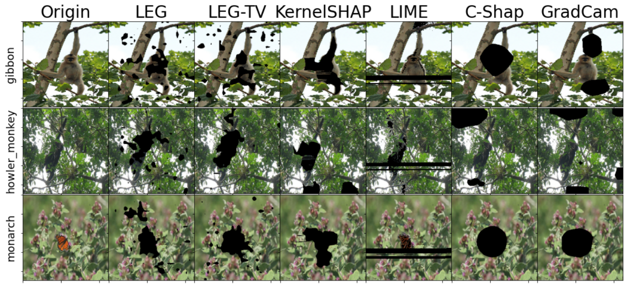


Figure 6: Examples of 10% images masked for all methods.

We compute S_{key} under distance-100, black-out, mean and noise masking settings. The results are summarized in Table 1. We observe that LEG and LEG-TV achieve the lowest key size in Distance-100 and black-out schemes, and under mean and noise masking it performs close to the best performer, KernelSHAP. We note that KernelSHAP relies on a segmentation hyperparameter, and its performance wildly varies with respect to the choice of the hyperparameter - we demonstrate this on an additional study in the Appendix with examples from the MNIST dataset.

The visual analysis also indicates that pixels identified by LEG-TV are visually more meaningful from a human perspective. We demonstrate this in Figure 6, where we plot the top 10% most salient pixels according to different procedures for three randomly selected images in the dataset. In the first image, LEG-TV is able to select different parts of the gibbon. In the second image, LEG and LEG-TV not only figure out the body howler monkey hidden in the background, but also part of the upper limb. However, KernelSHAP and LIME only discover the body part, GradCam and C-

Methods	Masking Techniques			
	Dist-100	Black-out	Mean	Noise
LEG	**0.163**	**0.062**	0.118	**0.069**
	(0.012)	(0.005)	(0.006)	(0.004)
LEG-TV	0.170	**0.056**	0.116	**0.068**
	(0.012)	(0.004)	(0.006)	(0.004)
KernelSHAP	0.251	0.059	**0.074**	**0.060**
	(0.017)	(0.004)	(0.004)	(0.003)
LIME	0.269	0.111	0.161	0.084
	(0.016)	(0.008)	(0.010)	(0.006)
C-Shapley	0.380	0.101	0.131	0.105
	(0.018)	(0.006)	(0.007)	(0.006)
GradCam	0.318	0.086	0.115	0.089
	(0.018)	(0.006)	(0.007)	(0.006)

Table 1: Average minimal perturbation size, $\bar{S}_{key}$, that changes the top prediction class under different masking schemes for VGG-19 using 500 randomly chosen samples in ImageNet. Standard errors are provided in parentheses.

Shapley completely fail. In the last image, LEG and LEG-TV can detect the location of the monarch butterfly, as well as the area of blooms in the background which indicates potential background bias in the models. We also see that KernelSHAP, C-Shapley and GradCam have a tendency to choose compact areas while LEG-TV tends to mask non-contiguous regions despite the smoothness penalty in the formulation.

6.3. Sanity Check

Adebayo *et al*. [1] tests the validity of saliency estimation procedures by varying the weights of the neural network. In a technique named, “cascading randomization”, authors replace the fitted weights of a CNN layer by layer starting from the final layer, and compute the saliency scores with each change. Clearly, a deep learner with randomly chosen weights should have no prediction power, and interpretations based on it should be meaningless. We expect LEG-TV tends to get zero saliency score for all of the pixels through cascading randomization. Small artifacts that might arise in this process, such as positive or negative saliency scores with no spatial structure, should be smoothed over due to the TV penalty in the end.

To verify our intuition, we perform a cascading randomization on the weights of a VGG-19 network. The network weights are replaced by random numbers in a cascading order, starting from the last layer. We generate the dataset for this analysis by randomly selecting 30 images from the web that have matching class categories in ImageNet[3] The results of our experiment for four images are shown in Figure 7. For all of the images in our analysis, LEG-TV estimate loses its pattern gradually after the first or the second perturbation, and the estimate is reduced to either zero or random noise after randomization of the first convolutional layer. That is, after the weights are perturbed, the LEG-TV method fails to detect any signal that could be used for interpretation. These results show that the interpretation provided by our proposed method is both reliable and dependent on the classifier. In order to distinguish whether the effect is due to the formulation of LEG or the total variation penalty, we also repeat the sanity checks using LEG and a very small L coefficient to impose a minimally smooth estimate. The results of this analysis, which are provided in the Appendix, suggest that the robustness of the LEG approach with respect to the sanity checks is due to the reliability of the underlying estimate, and not solely due to the penalties imposed.

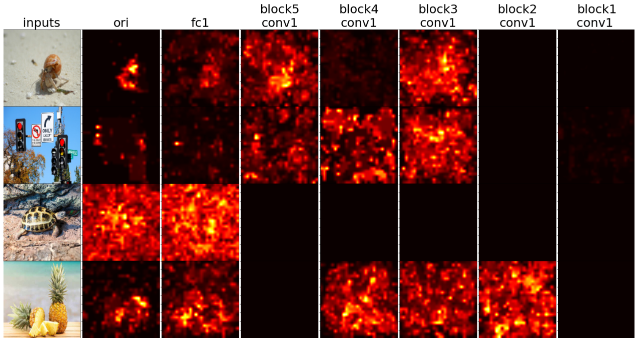


Figure 7: Results of the sanity check with cascading randomization.

7. Conclusion

In this paper, we propose a linearly estimated gradient (LEG) saliency estimation framework for black-box computer vision models that is gradient-weight based and model-agnostic. To the best of our knowledge, this is the first work that aims to address the model agnostic saliency method with statistical consistency. We further propose a new computationally efficient estimator (LEG-TV) using graphical representations of data. In addition to performing a theoretical analysis of the convergence rate, we propose a novel structured Gaussian noise approach that is capable of accelerating the convergence rate substantially. Our experimental results reveal that our proposed models, LEG and LEG-TV, consistently deliver a better performance than other model-specific or model-agnostic methods. In summary, our proposed framework is computationally efficient, requires no prior knowledge about the models, and can guarantee statistical consistency, clearly demonstrating its promise as an essential saliency estimation framework for those working in a wide range of model interpretation fields.

[3] This step ensures that we avoid using any images that might have been used to train the network.

References

[1] Julius Adebayo, Justin Gilmer, and et al. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9505–9515, 2018. 1, 2, 6, 8

[2] Julius Adebayo, Justin Gilmer, Ian Goodfellow, and Been Kim. Local explanation methods for deep neural networks lack sensitivity to parameter values. *arXiv preprint arXiv:1810.03307*, 2018. 1

[3] MOSEK ApS. *MOSEK Optimizer API for Python 9.1.6*, 2020. 4

[4] Sebastian Bach, Alexander Binder, and et al. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015. 1

[5] David Baehrens, Timon Schroeter, Stefan Harmeling, and et al. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010. 1

[6] Peter J Bickel, Ya'acov Ritov, Alexandre B Tsybakov, et al. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009. 4

[7] Collin Burns, Jesse Thomason, and Wesley Tansey. Interpreting black box models with statistical guarantees. *arXiv preprint arXiv:1904.00045*, 2019. 1

[8] Tony Cai, Weidong Liu, and Xi Luo. A constrained 1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011. 3

[9] Emmanuel Candes and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. *The annals of Statistics*, 35(6):2313–2351, 2007. 3

[10] Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. L-shapley and c-shapley: Efficient model interpretation for structured data. In *ICLR*, 2019. 1

[11] Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. In *Advances in Neural Information Processing Systems*, pages 13567–13578, 2019. 2

[12] Maximilian Alber Klaus-Robert Müller Dumitru, Erhan Been Kim Sven Dähne Pieter, Jan Kindermans, and Kristof T Schütt. Learning how to explain neural networks: Patternnet and patternattribution. In *International Conference on Learning Representations*, 2018. 1

[13] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009. 1

[14] Jianqing Fan. Features of big data and sparsest solution in high confidence set. *Past, present, and future of statistical science*, pages 507–523, 2013. 3, 4

[15] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437, 2017. 1, 3

[16] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 9734–9745, 2019. 3

[17] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017. 1

[18] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999. 5

[19] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017. 1

[20] Weili Nie, Yang Zhang, and Ankit Patel. A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. *arXiv preprint arXiv:1805.07039*, 2018. 1

[21] Jorge Nocedal and Stephen J Wright. Numerical optimization second edition. *Numerical optimization*, pages 497–528, 2006. 4

[22] N. Ravishanker and D.K. Dey. *A First Course in Linear Model Theory*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2001. 3

[23] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proc. KDD*, pages 1135–1144. ACM, 2016. 1, 2

[24] Ramprasaath R Selvaraju, Cogswell, and et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017. 1

[25] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proc. ICML*, pages 3145–3153. JMLR. org, 2017. 1, 2

[26] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 1

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[28] Sahil Singla, Eric Wallace, Shi Feng, and Soheil Feizi. Understanding impacts of high-order loss approximations and features in deep learning interpretation. In *International Conference on Machine Learning*, pages 5848–5856, 2019. 1

[29] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 1, 2

[30] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 1

[31] Robert J Vanderbei. *Linear Programming: Foundations and Extensions*. Springer, 2014. 2, 4, 6

[32] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. In *Advances in Neural Information Processing Systems*, pages 10967–10978, 2019. 2, 6

[33] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014. 1

[34] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proc. CVPR*, pages 2921–2929, 2016. 1