New Definitions and Evaluations for Saliency Methods: Staying Intrinsic, Complete and Sound

Arushi Gupta[*,1], **Nikunj Saunshi**[*,1], **Dingli Yu**[*,1], **Kaifeng Lyu**[1], **Sanjeev Arora**[1]

[1]Princeton University
{arushig,nsaunshi,dingliy,klyu,arora}@cs.princeton.edu

[*]Denotes equal contribution

Abstract

Saliency methods compute heat maps that highlight portions of an input that were most *important* for the label assigned to it by a deep net. Evaluations of saliency methods convert this heat map into a new *masked input* by retaining the k highest-ranked pixels of the original input and replacing the rest with "uninformative" pixels, and checking if the net's output is mostly unchanged. This is usually seen as an *explanation* of the output, but the current paper highlights reasons why this inference of causality may be suspect. Inspired by logic concepts of *completeness & soundness*, it observes that the above type of evaluation focuses on completeness of the explanation, but ignores soundness. New evaluation metrics are introduced to capture both notions, while staying in an *intrinsic* framework—i.e., using the dataset and the net, but no separately trained nets, human evaluations, etc. A simple saliency method is described that matches or outperforms prior methods in the evaluations. Experiments also suggest new intrinsic justifications, based on soundness, for popular heuristic tricks such as TV regularization and upsampling.

1 Introduction

Saliency methods try to understand why the deep net gave a certain answer on a particular input, and are an important component of explainability, fairness, robustness, etc., in deep learning. This paper restricts attention to the (large) class of saliency methods that return an importance score for each coordinate of the input — often visualized as a heat map — which captures the coordinate's importance to the final decision.[1] Early saliency methods used an axiomatic system of "credit attribution" to individual input coordinates, using backpropagation-like methods [5, 36] and cooperative game theory, such as Shapley values [27, 45]. See Section 2 and Samek et al. [34] for discussion of strengths and limitations of such methods. More recent methods try to find heat maps that are largely concentrated on a smaller set of pixels/coordinates and are discussed further below.

There exists an ecosystem of *evaluation metrics* to evaluate saliency methods. Evaluations can be *extrinsic*, involving human evaluation [1], and comparison to certain ground truth explanations [47]. We restrict attention to *intrinsic* evaluations, which use computations involving the net itself and the heat/saliency map — but no human evaluations or evaluations that involve training new deep nets. Popular intrinsic evaluations include *saliency metric* [9] and insertion & deletion metrics [30].

A frequent idea in intrinsic evaluations (see Section 3 for references) is to create a new composite input — or sequence of such inputs — using the heat map and the original input, and to evaluate the original net on this composite input. For example, if M is a binary vector with 1's in the k coordinates with

[1]Heat maps suffice for recognition/classification tasks; other tasks may require more complex explanations.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Figure 1: Masked CIFAR-10 images generated by our procedure with TV regularization. "Artifacts" exist for masks generated for incorrect labels; more examples can be found in Figure 11 in Appendix. The base model outputs the correct label on the original image (ship and bird resp.) with probability at least 0.99, and assigns probability at most 10^{-5} for the incorrect label (cat and frog resp.). With the generated masks, the AUC metric (defined in Section 3.1) for the correct label remains high (around 0.94 and 0.90), which corresponds to *completeness*, but AUC metric for the incorrect label rises tremendously (around 0.18 for cat mask, 0.71 for frog mask.) This suggests violation of *soundness*.

the highest values in the heat map, then $x \odot M$ (with $\odot$ denoting coordinate-wise multiplication) can be viewed as a masked input where only k of the original coordinates of x remain. It is customary to replace the zeroes in this masked input with "uninformative" values, which we refer to as *gray* pixels. This masked input is fed into the original net[2] to check if the net outputs the same label as on the full input — and, if so, one should conclude that the unmasked coordinates of x were salient to the net's output. Not surprisingly, recent methods [11, 9, 31] for finding heatmaps use an objective that tries to directly optimize performance on such evaluations, greatly outperforming older axiom-based methods.

Should we trust such mask-based explanations? At first sight, this question may appear naive. In the above-described masked input $x \odot M$, some k pixels were copied from the original image and the remaining pixels were made "gray." If the original net outputs essentially the same label as it did on the full image, does this not *prove* that the portion that came from the mask M must have *caused* the net's output on the full image? The answer is "No", because the *positions* of the gray pixels can carry some signal.

Figure 1 shows that in standard datasets, it is possible to create masks out of most inputs such that the masked input can cause the net to assign high probability for a *different* label than in its original prediction. Prior works referred to this phenomenon as "mask artifacts" and suggested using total variation (TV) regularization to mitigate this issue. But Figure 1 shows TV regularization is not a full solution to masks that cause the net to (incorrectly) flip the answer.

Soundness needed. The masked input method is trying to *prove* that a certain portion of the image "caused" the net's output. Usually logical reasoning must display both *completeness* (i.e., all correct statements are provable) and *soundness* (incorrect statements cannot be proved). Checking that the net's output is essentially unchanged with the masked input is akin to verifying *completeness*.

But this by itself should not be convincing because, as mentioned, it is possible to use the method to find a different mask to justify another (i.e., wrong) label. *Soundness* would require verifying that the same saliency method cannot be used to produce masked inputs that make the net output a different label. Together, completeness and soundness ensure that the evaluation of the masked inputs produced using various labels approximately tracks the model's probability of assigning the label (see Section 3.2).

Main Contributions of this paper:

- Section 3.2 draws connections to logical proof systems and formalizes completeness and soundness in the context of saliency methods. These are intrinsic evaluations of saliency without appealing to human judgements (e.g., whether or not machine-generated masks contain artifacts). Our notion of completeness and soundness requires saliency methods to output a map for *every label* and not just the model prediction.
- Section 4 revisits methods that learn masks through optimization. Incorporating soundness, we propose a simple optimization method that maximizes the probability for any given label[3], rather than only the label deemed most likely by the net. A key design choice of this method is the

[2]This raises a potential issue of distribution shift because the net never trained on masked inputs. In practice, especially for image data, trained nets continue to work fine on masked inputs.

[3]Computation cost proportional to the number of labels is only incurred while designing/evaluating the saliency method but not deployment time. Another alternative is to verify soundness condition for top k labels.

pixel replacement strategy during training: non-salient pixels are replaced with pixels of a random image, instead of using gray pixels, blurring or counterfactual models. [4]

- Experiments (in Section 6) estimate completeness and soundness scores for various methods. Our simple method does well on completeness and soundness as one would expect, but the slight surprise is that it performs comparably to prior masked-based methods on earlier saliency evaluations as well, suggesting that soundness can be achieved without paying much in completeness. Heuristic choices in existing mask-based methods such as TV regularization and upsampling, which were hitherto justified extrinsically (i.e., they make masks look natural to humans), receive an *intrinsic* and precise justification: they improve soundness. A theoretical result in Section 5 complements this experimental finding by showing that TV regularization helps saliency methods even in a simple linear classification setting, not just for vision data.

2 Prior approaches

We relegate a more thorough description of prior work to Appendix E, but mention some common saliency and saliency evaluation methods here. Saliency methods aim to explain a model's decision about an input. Saliency evaluation methods aim to evaluate the goodness of a saliency method.

Saliency methods include *backpropagation based approaches* such as Gradient $\odot$ Input [38], iGOS++ [20], Robust Perturbations [22], LRP [5], GradCAM [36], Smooth-Grad [39]. Another line of work is *masking methods* which include techniques based on averaging over randomly sampled masks [30], optimizing over meaningful mask perturbations [11], and real time image saliency using a masking network [9]. Pixels that have been removed from the image by the mask may be replaced by greying out, by Gaussian blurring as in [11], or with infillers such as CA-GAN [46] used in [8, 31], or DFNet [14]. The pixel replacement strategy we used is closely related to hot deck imputation [32], where features may be replaced either by using the mean feature value (analogous to replacing with grey) or sampling from the marginal feature distribution (analogous to replacing with image pixels sampled from other training images). Some prior work [7] has found that mean imputation does not significantly affect model output on the beer aroma review dataset. On Imagenette, by contrast, we found that replacement strategy can matter (See Figure 4). De Cao et al. [10] find masks using differentiable masking. *Boolean logic* is another approach for saliency methods [18, 17, 28, 29, 49].

Arguments about saliency. Discussions about the methods and the meaning of saliency appear, among others, in [37, 12, 13, 40]. Some reveal situations where Shapley axioms work against feature selection or where Shapley values may be calculated in conflicting ways [12, 40]. Others question efficacy of saliency methods that add noise [37], making explanations non class discriminative [13].

Saliency evaluation methods. Extrinsic evaluation metrics include the **WSOL** metric, and **Pointing Game** metric proposed by Zhang et al. [47] and **ROAR** [15]. Other more intrinsic methods include early saliency evaluation techniques like **MorF** and **LerF** [33], **Insertion and Deletion game** proposed by Petsiuk et al. [30], which involve either inserting pixels in order of most importance or deleting pixels in order of most importance. **BAM** [44] creates saliency maps by pasting object pixels from MSCOCO [25]. The **Saliency Metric** proposed by Dabkowski and Gal [9] thresholds saliency values above some α chosen on a holdout set, finds the smallest bounding box containing these pixels, upsamples and measures the ratio of bounding box area to model accuracy on the cropped image, $s(a, p) = \log(\max(a, 0.05)) - \log(p)$ where a is the area of the bounding box and p is the class probability of the upsampled image. Adebayo et al. [1] proposed "sanity checks" for saliency maps. They note that if a model's weights are randomized, it has not learned anything, and therefore the saliency map should not look coherent. They also randomize the labels of the dataset and argue that the saliency maps for a model trained on this scrambled data should be different than the saliency maps for the model trained on the original data. We study the effect of model layer randomization on our method in Appendix section D.4 and find that randomizing the model weights does cause our saliency maps to look incoherent. Tomsett et al. [43] also discover sanity checks for saliency metrics, finding that saliency evaluation methods can yield inconsistent results. They evaluate saliency maps on reliability, i.e. how consistent the saliency maps are. To measure a method's reliability, because access to ground truth saliency maps are not available, they use three proxies 1) inter-rater reliability, i.e. how whether a saliency evaluation metric is able to consistently rank some saliency methods

[4] Code for computing completeness, soundness, and our masking method available at `https://github.com/agup/soundness_saliency.git`

above others, 2) inter-method reliability, which indicates whether a saliency evaluation metric agrees across different saliency methods, and 3)internal consistency reliability, which measure whether different saliency methods are measuring the same underlying concept.

Discussions. Brunke et al. [6] show that perturbation methods are sensitive to baseline and Petsiuk et al. [30] point out that human centric explanations (based on bounding boxes) may not reveal why the model made a certain decision. Our notion of intrinsic saliency method outputs the mask but introduces soundness to validate it.

3 Masking explanations and completeness/soundness

The goal of a saliency method is to produce explanations for the outputs of a given deep net f. We think of a saliency method (Definition 3.1) producing explanations — a heat map in this case — as providing a *proof* of the statement f *outputs the label* a *for input* x. Inspired by logical proof systems, we propose that intrinsic evaluations for saliency methods should test for both completeness and soundness, and the rest of the section provides definitions for these notions.

3.1 Saliency methods and AUC metric

For a net f, let $f(x, a)$ denote the (post-softmax) output probability of the net on input x for label a. We are interested in saliency methods that output a heatmap with a scalar per coordinate of the input.

Definition 3.1 (Saliency Method)**.** A saliency method $\mathcal{M}$ is an algorithm that takes 1) a deep net f, 2) an input x, and 3) a label a and produces a heatmap $\boldsymbol{m} = \mathcal{M}(f, x, a) \in \mathbb{R}^{\dim(x)}$.

Note that in the above definition, $\mathcal{M}$ can access any part of f including intermediate layers and gradients. Examples of methods that can be interpreted as returning heatmaps are mask-based methods [11, 9, 8, 2, 31] that output heatmaps in $[0, 1]^{\dim(x)}$, backpropagation-like methods [5, 36], Shapely values [27, 45], and Gradient $\odot$ Input.

Given such a saliency method, an important question is how to evaluate the quality of the "explanations" or heat maps it produces. A common evaluation idea is to use the behavior of the net f on a "masked" input — only the top few pixels based on the heat map in question are retained, while the others are hidden appropriately. Inspired by this idea we define evaluation metrics for saliency methods. We abstract the step of retaining the top few pixels into an *input modification process*, defined below.

Definition 3.2 (Input Modification Process Γ)**.** Let Γ be a potentially randomized procedure that takes an input x and a heatmap $\boldsymbol{m}$ and generates a modified input $\tilde{x}$. i.e. $\tilde{x} \sim \Gamma(x, \boldsymbol{m})$.

This input modification process defines a *base metric* that is the starting point of the completeness and soundness metrics that will be defined later.

Definition 3.3 (Base Metric)**.** Let x be an input (image), a a label, $\boldsymbol{m}$ a heatmap and Γ an input modification process. Then, a base metric is a function $g(x, a, \boldsymbol{m}) = \mathbb{E}_{\tilde{x}\sim\Gamma(x,\boldsymbol{m})}[f(\tilde{x}, a)]$, where f is the neural net of interest[5].

The base metric measures the expected output of f acting on *modified* inputs $\tilde{x}$ and label a and different choices for Γ lead to different base metrics. Some examples of Γ that stay in the *intrinsic* framework are: graying out the pixels outside some subset S of $\boldsymbol{m}$, replacing them by pixels from a Gaussian blurring of x [11]. Input modification through Γ also shows up in saliency methods, including our proposed method in Section 4; we discuss prior choices of Γ in that section. For our evaluation metrics, however, we choose Γ that grays out remaining pixels.

The popular Area-Under-the-Curve (AUC) evaluation metrics [30] of saliency methods can, in fact, be reinterpreted in terms of our base metric. We state the *insertion game* and its AUC below, since it forms the basis of our subsequent completeness and soundness metrics, and then define the AUC metric as an instantiation of Definition 3.3.

Definition 3.4 (AUC of insertion game)**.** For $s = 1$ to $\dim(x)$ take the top s pixels as per $\boldsymbol{m}$, and plot the probability $f(x, a)$ given by model to label a on the input x, where the top s pixels of x are retained and remaining pixels are assigned a default gray value. Return the area under the curve.

[5] In general we can replace f with any target function h that depends on f

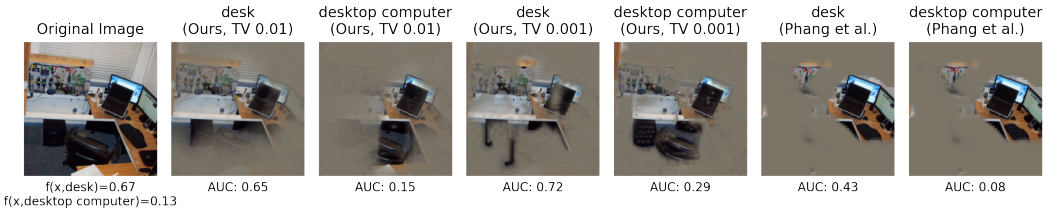


Figure 2: ImageNet image and masked versions for labels desk and desktop computer. Heatmaps generated by our method with $\lambda_{TV} = 0.01$, $\lambda_{TV} = 0.001$ and Phang et al. [31]'s method, respectively.

Definition 3.5 (AUC metric)**.** We denote by $g_{\text{AUC}}(x, a, \boldsymbol{m})$ the base metric when a modified input $\tilde{x} \sim \Gamma(x, \boldsymbol{m})$ is obtained as follows:

- Sample $s \sim \{1, \ldots, \dim(x)\}$ and pick S to be top s pixels as per $\boldsymbol{m}$.
- Set $\tilde{x}[S] = x[S]$ and $\tilde{x}[\bar{S}] = x_{\text{gray}}[\bar{S}]$ where x_{gray} is a "default" input with gray values.

We note that g_{AUC} is indeed equivalent to the AUC insertion game from Definition 3.4. While many choices of base metrics can be used, we use this AUC metric for subsequent definitions since it confers two benefits: 1) it allows for fair comparison between methods since all methods are evaluated at the same sparsity level; 2) it considers the effect of multiple sparsity levels instead of picking an arbitrary one. We are now ready to present the formalization of completeness and soundness.

3.2 Completeness and soundness

Kaifeng: I changed the ordering of paragraphs in this section. Can someone check?

Just as for proof systems, we would like a saliency method to be logically complete and sound, i.e., it should be able to justify the label output by the net, but no other label[6].

The statements to be "proved" are of the form *"does f think x has label a?"*. We use the model output probability $f(x, a)$ as fractional truth value of the "statement" (x, a). For a saliency map $\boldsymbol{m}$ that a saliency method $\mathcal{M}$ generates, we interpret the AUC metric $g_{\text{AUC}}(x, a, \boldsymbol{m})$ as the evaluation of truth value of the "proof" $\boldsymbol{m}$, as adjudged by the base metric g_{AUC}. Thus in this context, completeness for $\mathcal{M}$ would entail that whenever $f(x, a)$ is large (i.e., the statement is sufficiently true), $g_{\text{AUC}}(x, a, \boldsymbol{m})$ should be large (a sufficient good proof can, and is, generated). Soundness, on the other hand, would entail that $g_{\text{AUC}}(x, a, \boldsymbol{m})$ should be small whenever $f(x, a)$ is small. Thus we would like $g_{\text{AUC}}(x, a, \boldsymbol{m})$ to track the value of $f(x, a)$ through a lower bound and upper bound as follows:

Definition 3.6 (Completeness and Soundness)**.** Consider a saliency method $\mathcal{M}$. For $\alpha, \beta \leq 1$, the method $\mathcal{M}$ is *α-complete on f, x, a* if the heatmap $\boldsymbol{m} = \mathcal{M}(f, x, a)$ it produces satisfies $\boxed{g_{\text{AUC}}(x, a, \boldsymbol{m}) \geq \alpha\, f(x, a)}$, while $\mathcal{M}$ is *β-sound on f, x, a* if $\boxed{g_{\text{AUC}}(x, a, \boldsymbol{m}) \leq \frac{1}{\beta} f(x, a)}$. Conversely the completeness and soundness scores for a heap map $\boldsymbol{m}$ on (x, a) are defined as:

$$\alpha(x, a) = \min\left\{\frac{\max\{g_{\text{AUC}}(x, a, \boldsymbol{m}), \epsilon_1\}}{f(x, a)}, 1\right\}, \quad \beta(x, a) = \min\left\{\frac{\max\{f(x, a), \epsilon_2\}}{g_{\text{AUC}}(x, a, \boldsymbol{m})}, 1\right\}. \tag{1}$$

Definition 3.7 (Worst case completeness and soundness)**.** For a saliency method $\mathcal{M}$, these metrics are defined by taking the worst case of the completeness/soundness over labels a for a given x that is sampled from the underlying input distribution. More precisely, $\alpha(\mathcal{M}) = \mathbb{E}_x\left[\min_a \alpha(x, a)\right], \beta(\mathcal{M}) = \mathbb{E}_x\left[\min_\beta \beta(x, a)\right]$.

Our metrics implicitly require a saliency method to output a meaningful map for every label (or at least all labels assigned non-negligible probabilities by the net) and not just for the top model prediction. Notice, α-completeness means that if the model outputs a high probability for label a, then the probability for label a after seeing only the coordinates in the salient sets is also high, while β-soundness means that this probability is not too high. Checking both conditions verifies if the AUC score $g_{\text{AUC}}(x, a, \boldsymbol{m})$ of a map $\boldsymbol{m}$ is in the interval $\left[\alpha f(x, a), \frac{1}{\beta} f(x, a)\right]$. We want α, β as close

[6]We focus on single label multi-class classification; extensions to multi-label are left for future work.

to 1 as possible. This notion is different from the usual AUC insertion game, which just requires $g_{\text{AUC}}(x, \hat{y}, \boldsymbol{m})$ to be as large as possible for the model prediction $\hat{y} = \arg\max_a f(x, a)$. Moreover, our worst-case completeness and soundness metrics require the $g_{\text{AUC}}(x, a, \boldsymbol{m})$ to roughly *match* the model prediction $f(x, a)$, *for every pair* (x, a),[7] where the thresholds ϵ_1, ϵ_2 in Equation (1) help in ignoring the case where the model outputs very small probabilities — for instance we can ignore any label for completeness if $f(x, a) < \epsilon_1$.

3.3 Illustration of completeness and soundness on an ImageNet example

See Figure 2. For this image x, the model assigns reasonably high probabilities to labels $a_1 =$ "desk" and $a_2 =$ "desktop computer", with $f(x, a_1) \approx 0.67$ and $f(x, a_2) \approx 0.13$. We compare the maps computed by three methods — $\mathcal{M}_1$: our mask learning procedure from Section 4 with TV regularization 0.01, $\mathcal{M}_2$: our method with TV of 0.001, and $\mathcal{M}_3$: the mask learned by Phang et al. [31]. The corresponding maps generated are denoted by $\boldsymbol{m}_i, i \in \{1, 2, 3\}$. For this discussion we only consider the effect of labels a_1 and a_2 on completeness and soundness, and we assume the thresholds satisfy $\epsilon_1 = \epsilon_2 = 0$.

Completeness. For the label a_1, the AUC scores $g_{\text{AUC}}(x, a_1, \boldsymbol{m}_i)$ for the three methods are roughly 0.65, 0.70 and 0.43 respectively. Thus all three maps can "certify" the label a_1 well enough, with $\boldsymbol{m}_2$ doing the best job. Consequently the completeness scores (Equation (1)) for $\boldsymbol{m}_1$ on the pair (x, a_1) is $\alpha_1(x, a_1) = \min\{\frac{0.65}{0.67}, 1\} \approx 0.97$; similarly $\alpha_2(x, a_1) = 1$ and $\alpha_3(x, a_1) = \frac{0.43}{0.67} \approx 0.64$. Similarly for label a_2 that is assigned a probability of 0.13, the completeness scores $\alpha_i(x, a_2)$ are 1, 1, and 0.58. Thus the worst case completeness from Definition 3.7, i.e. $\min_{j \in \{1,2\}} \alpha(x, a_j)$, for the three methods are 0.97, 1 and 0.58. Based on completeness $\mathcal{M}_2$ seems like the best method; however soundness will paint a different picture.

Soundness. Using the same AUC scores we compute the soundness scores, defined in Equation (1) as $\beta_i(x, a) = \min\{\frac{f(x,a)}{g_{\text{AUC}}(x,a,\boldsymbol{m}_i)}, 1\}$. This gives us soundness scores on a_1 of 1, 0.95 and 1. The more interesting case is label a_2, for which the model does not assign very high probability (0.13). Here the mask from $\mathcal{M}_2$ still gets a very high AUC score of 0.29, leading to a soundness score $\beta_2(x, a_2) \approx \frac{0.13}{0.29} \approx 0.44$. $\mathcal{M}_1$ on the other hand is not as overconfident in its explanation as $\mathcal{M}_2$, and gets $\beta_1(x, a_2) = \frac{0.13}{0.15} \approx 0.89$. The worst case soundness for three methods ends up being 0.89, 0.44 and 1.

This examples highlights how completeness and soundness together can give a more nuanced comparisons of saliency maps, including understanding the benefit of TV regularization. Experiments in Section 6 do a more thorough comparison of many saliency methods on these and prior metrics.

4 Procedures for finding masking explanations

We propose a very simple method to find saliency maps with good empirical completeness and soundness scores. Detailed intuitions and implementations appear in Appendix A. Our method is similar to prior work on mask-based saliency maps and is based on the idea of SSR (Smallest Sufficient Region)[8]. In particular for an input x and label a, given a network f, the main goal is to learn a map (or mask) $M \in \{0, 1\}^{hw}$ such that $\mathbb{E}_{\tilde{x} \sim \Gamma(x, M)}[-\log(f(\tilde{x}, a))]$ is minimized, i.e. the probability that the network assigns to a modified (or composite) input $\tilde{x}$ is high. The key difference from prior masked-based methods is our choice of Γ, which retains the pixels of x corresponding to M, but replaces the rest of the pixels with values from a *randomly sampled* image $\bar{x}$ from the training set[9] $\mathcal{X}$, i.e.

$$\tilde{x} \sim \Gamma(x, a) \;\equiv\; \bar{x} \sim \mathcal{X}, \tilde{x} = M \odot x + (1 - M) \odot \bar{x} \tag{2}$$

Modifications procedures (Γ) that have been considered in prior work include: graying or blurring out the remaining pixels [11], and using a conditional image generative model to fill in the remaining pixels [8, 2]. Replacing with random image pixels forces the mask-finding algorithm to solve a more difficult task of having the network predict a high confidence for the label, despite the presence of another

[7] Works like Gradient $\odot$ Input and LRP use *completeness* in a somewhat different sense. It requires the sum of the coordinate scores in the heatmap to *exactly* equal the logit of the label. Our completeness + soundness together try to approximately match the g_{AUC} score to the output probability on this label.

[8] We do not incorporate any SDR (Smallest Destroying Region) component in our method.

[9] We think other random image distributions should work too.

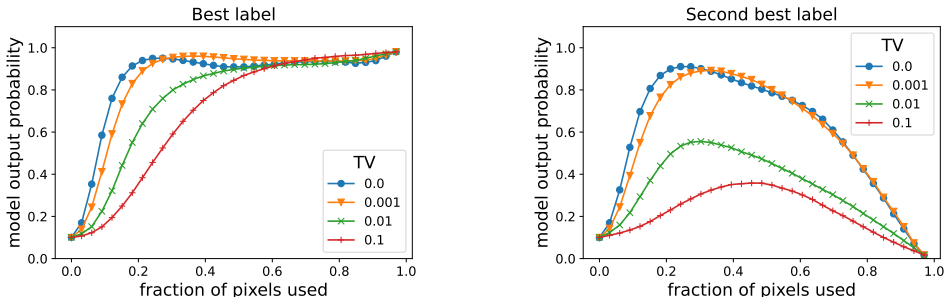


Figure 3: Plot of model output probability as more pixels from the original image are retained using learned masks. The remaining pixels are replaced with gray. Different curves correspond to different values of TV regularization (λ_{TV}). Larger Area-Under-the-Curve (AUC) for the left figure (best label) suggests good completeness, while lower AUC for the right figure suggests good soundness. Plots suggest that adding TV significantly helps with soundness, while only slightly hurting completeness.

image. This amounts to grafting salient pixels of x on top of a random image, reminiscent of BAM evaluations [44] for saliency methods. We also note a methodological difference from Dabkowski and Gal [9], Phang et al. [31] in that we do not train a separate neural network to output the mask.

As is standard, we relax the domain of masks M from binary $\{0,1\}^{hw}$ to continuous $[0,1]^{hw}$ parametrized by a sigmoid. We define a composite input, where the part of x on M is superimposed onto a distractor $\bar{x} \sim \mathcal{X}$ as $\tilde{x} = M \odot x + (1 - M) \odot \bar{x}$. We then have the following simple *vanilla* objective, where the ℓ_1 norm penalty on M helps to reduce the size of masks.

$$L(M; (x, a)) = \mathbb{E}_{\bar{x} \sim \mathcal{X}} \left[-\log(f(\tilde{x}, a)) \right] + \lambda_1 \|M\|_1, \text{where } \tilde{x} = M \odot x + (1 - M) \odot \bar{x}. \tag{3}$$

Promoting soundness. Tricks used in prior mask-finding approaches were tested for their effect on soundness. These include 1) Total-Variation (TV) penalty [11] and 2) upsampling of the mask from a lower resolution one [30] by learning a low-resolution mask at scale s, $M \in [0,1]^{hw/s^2}$. For example, on ImageNet, $s = 1$ corresponds to learning a 224×224 mask, while $s = 4$ corresponds to learning a 56×56 mask and upsampling by a factor of 4. Incorporating 1) and 2) leads to the following *modified* objective

$$\begin{aligned} L(M; (x, a)) &= \mathbb{E}_{\bar{x} \sim \mathcal{X}} \left[-\log(f(\tilde{x}, a)) \right] + \lambda_{TV} TV(M^{\times s}) + \lambda_1 \|M^{\times s}\|_1 \\ &\text{where } \tilde{x} = M^{\times s} \odot x + (1 - M^{\times s}) \odot \bar{x}, \end{aligned} \tag{4}$$

and $M^{\times s} \in \mathbb{R}^{hw}$ is obtained by upsampling M by a factor of $s \in \{1, 4\}$ via bilinear interpolation.

While the motivation cited for these two commonly employed "tricks" is to avoid artifacts [11], in our intrinsic framework "artifacts" does not have an interpretation. What looks like an artifact to a human observer may be relevant to the net's decision-making. Indeed, our experiments show that while TV penalty or upsampling does produce better looking masks, they lead to a drop in the *completeness metric*. But they significantly improve *soundness*. This provides an intrinsic justification for the use of such tricks, instead of extrinsic justifications related to human interpretability.

5 On benefit of TV regularization in saliency

Past justifications for TV regularization in saliency methods focused on its ability to make heatmaps in images that look more natural to humans. But that is not an intrinsic justification. In the intrinsic framework proposed in this paper, experiments in Section 6 show TV regularization helps improve soundness of the saliency method. The current section reinforces this intrinsic benefit of TV by sketching a simple example (details in Appendix F) showing how TV helps even in a simple linear model $\boldsymbol{x} \mapsto \boldsymbol{w}^\top \boldsymbol{x}$ on non-image input $\boldsymbol{x} \in \mathbb{R}^d$ with binary label $y \in \{\pm 1\}$.

Suppose that the classifier has margin at least $\gamma > 0$, that is, $y\boldsymbol{w}^\top \boldsymbol{x} = \sum_{i=1}^d y w_i x_i \geq \gamma$. We focus on saliency methods that can return a binary heatmap $\boldsymbol{m} \in \{0,1\}^d$ to validate the label, where the coordinates marked as 1 constitute a set S of saliency coordinates. Let $\Gamma(x, \boldsymbol{m})$ be the input modification process that assigns 0 to all input coordinates outside the corresponding saliency set S.

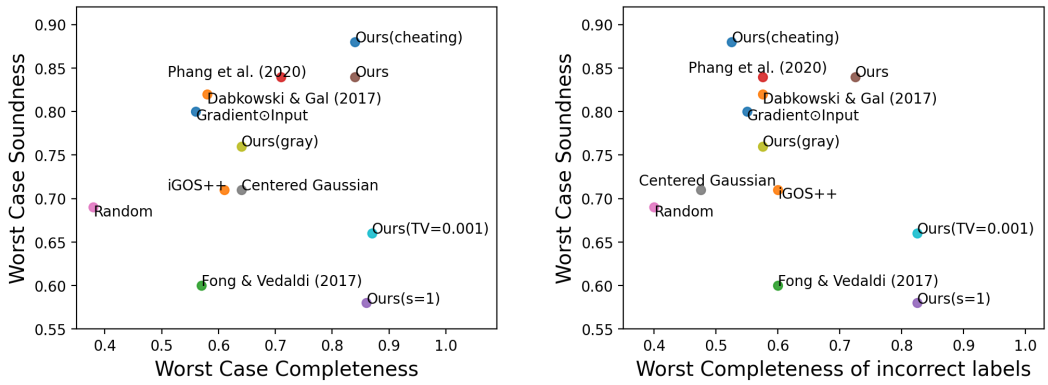


Figure 4: Worst case completeness v.s. soundness plot on Imagenette for different methods. (Upper right corner is best.) Each point represent either a prior method [31, 9, 11, 38] or our method with different settings. By default, our methods use upsampling $s = 4$, TV penalty $\lambda_{TV} = 0.001$ and use other images for composite image (see Equation (2)). "Ours" uses the default setting; "Ours($s = 1$)" uses upsampling $s = 1$; "Ours($TV = 0.001$)" uses TV penalty factor $\lambda_{TV} = 0.001$; "Ours(gray)" fills gray pixels in modifications procedures Γ; and "Ours(cheating)" uses masks generated for the correct label as masks for all labels. The x-axis of the right figure is the average worst completeness of all incorrect labels over samples where the second best label has model probability at least 0.01. See interpretations of the plot in Section 6.2.

Consider an input x such that $\boldsymbol{w}^\top \boldsymbol{x} > 0$, i.e. $y = 1$. Firstly note that any set S of coordinates that contribute positively to $\boldsymbol{w}^\top \boldsymbol{x}$, i.e. $g(x, y, \boldsymbol{m}) := \sum_{i \in S} w_i x_i > 0$, will be able to certify the label $y = 1$ and thus would satisfy *completeness*. However this might not be convincing to a user, since it is also possible to come up with a set S' that certifies the label $y = -1$ by picking only negative coordinates of x. This would not satisfy *soundness* as the original model only predicts $y = 1$

This issue of soundness can be solved by adding a regularization or constraint on the set of allowable maps, in a way that ensures that it is easy to find a mask $\boldsymbol{m}$ for the correct label y, but not for $-y$. Ideally, the constraints should be task specific, but for linear classification with no special structure, even a TV constraint after random permutation suffices. We consider the constraint that salient sets S should be an interval; this corresponds to $\mathrm{TV}(\boldsymbol{m}) \leq 2$. The following result shows that if we further constrain the length of intervals to be $\Theta(\frac{1}{\gamma^2} \log d)$, then with high probability, we can find an interval validating the model prediction y but we are unable to do the same for $-y$ (Details in Appendix F.)

Theorem 5.1. *For $\boldsymbol{x} \in \mathbb{R}^d, y \in \{\pm 1\}$, after random shuffling of the coordinates, the following holds for any $L_1 = \Omega(\frac{1}{\gamma^2} \log \frac{1}{\delta})$, $L_2 = \Omega(\frac{1}{\gamma^2} \log \frac{d}{\delta})$, where probability is over the random shuffle:*

1. *(Completeness) With probability $1 - \delta$, there is an interval of length L_1 that validates y;*
2. *(Soundness) With probability $1 - \delta$, no interval of length L_2 can validate $-y$.*

6 Experiments

In this section, we aim to 1) show TV penalty and upsampling have intrinsic justification that they significantly improve soundness at a small cost in completeness; 2) evaluate and compare prior saliency methods, and our method with different parameter settings, on our completeness and soundness metrics. Experiments are performed either on CIFAR-10 or Imagenette [16], a 10-class subset of ImageNet [10]. We compute maps and evaluate metrics on 1000 randomly drawn images from the original test set. Additional experiments on various models and datasets (including ImageNet and CIFAR-100) appear in Appendix D, as do details of training procedure (beyond those in Section 4). For all our completeness and soundness evaluations, we use thresholds of $\epsilon_1 = 0.01$ and $\epsilon_2 = 0.001$ (see Equation (1)).

[10] Completeness and soundness evaluation on all labels can get computationally expensive on ImageNet as it contains 1000 classes. This cost is only relevant when designing the method; not at deployment time.

Table 1: Saliency methods evaluate on prior metrics, worst case completeness and soundness on Imagenette. Insertion calculated with grey infilling. The best two of each column are marked bold. By default, our methods use upsampling $s = 4$, TV penalty factor $\lambda_{TV} = 0.01$ and use randomly sampled image in the modification procedures Γ. Difference from default setting is noted in parentheses. $\downarrow$ indicates lower is better. *Strong performance of is likely due to centrality bias (cf. Appendix C).

	Deletion $\downarrow$	Insertion (gray) $\uparrow$	Saliency Metric $\downarrow$	Worst Case Completeness $\uparrow$	Worst Case Soundness $\uparrow$
Gradient $\odot$ Input[38]	0.42	0.56	-0.35	0.56	0.80
iGOS++[20]	**0.37**	0.68	-0.36	0.61	0.71
Dabkowski and Gal [9]	0.48	0.66	-0.85	0.58	0.82
Fong and Vedaldi [11]	0.58	0.59	-0.40	0.57	0.60
Phang et al. [31]	**0.41**	0.75	-0.27	0.71	**0.84**
Ours	0.50	0.74	-0.90	0.84	**0.84**
Ours ($s = 1$)	0.49	**0.77**	$\mathbf{-1.18}$	**0.86**	0.58
Ours ($\lambda_{TV} = 0.001$)	0.45	**0.80**	$\mathbf{-1.01}$	**0.87**	0.65
Ours (gray infilling)	0.46	0.65	-0.35	0.64	0.76
Random	0.45	0.45	-0.35	0.38	0.69
Centered Gaussian	0.66	0.74^*	-0.97^*	0.64	0.71

6.1 Effect of TV penalty and upsampling

In this section, we study the effect of "tricks" like TV penalty and upsampling on the learned maps and evaluation metrics. Figure 3 shows how soundness and completeness of our masks changes with TV regularization. It plots the model output probability for CIFAR-10 as more pixels from the original image are retained using the mask. Our high level finding is that adding TV penalty significantly aids soundness, while only slightly hurting completeness. Exact evaluation of soundness and completeness for different λ_{TV} are listed in Table 2 in Appendix C. Relevant points of Figure 4 (which also contain other saliency methods) show effect of upsampling and TV penalty on completeness and soundness. We compare our method from Section 4 with different settings on Imagenette. Our method by default use $s = 4$ and $\lambda_{TV} = 0.01$. By comparing "Ours" against "Ours($s = 1$)" and "Ours($TV = 0.001$)", we observe that upsampling and TV penalty leads to slightly worse completeness but significantly better soundness. The effect of upsampling on previous saliency evaluation metric may be seen in the relevant entries of Table 1, where it leads to slightly worse performance on the insertion, deletion, and saliency metrics.

6.2 Comparison to existing metrics and methods

We compared our procedure to other methods including Gradient $\odot$ Input [38], iGOS++[20], Dabkowski and Gal [9] Fong and Vedaldi [11] and Phang et al. [31] on Imagenette. We also include simple baselines like "Random" map (each entry is a random Gaussian), and "Centered Gaussian" map (2-dimensional isotropic Gaussian distribution placed at the image center). Besides completeness and soundness metrics, we evaluate methods on the Deletion Game and Insertion Game metrics (`https://github.com/eclique/RISE`), and Saliency metric (SM) [9]. Detailed description of the different metrics can be found in the Appendix E.

For our method, we learn a map by optimizing the objective function in Equation (4) with $\lambda_{TV} \in \{0.01, 0.001\}$ and $s \in \{1, 4\}$; default settings are $\lambda_{TV} = 0.01, s = 4$. We also evaluate our method when other pixels are replaced with gray rather than a random image, denoted by "Ours (gray)". For uniformity, we use the identical ResNet50 pretrained on ImageNet as the base classifier for every saliency method. We normalize the maps so that all values lie in $[0, 1]$ before use.

Completeness and soundness. We compare different methods on the metrics from Definition 3.7, through the visualization in Figure 4 (left) and in Table 1. We find that our method achieves better completeness and good soundness compared to other methods and baselines. We posit that completeness is better due to our improved composite input strategy from Equation (2), where non-salient pixels are filled with another image pixels as opposed to other strategies. As an ablation, we see that "Ours (gray)" which uses gray replacement does not perform as well.

On the other hand, our soundness is good only for higher TV regularization $\lambda_{TV} = 0.01$ and higher upsampling $s = 4$, as discussed in Section 6.1. Soundness is very bad (but completeness slightly better) if these "tricks" are not used. We note that while Phang et al. [31] does as well as our method on soundness, it does so via "cheating" in an interesting way: they compute mask for the top label

and return the same mask for all labels. Since the method does not even try to generate masks for labels with low probability, it achieves good soundness for free. To delve deeper, we employ the same "cheating" with our method — return the same mask for all labels — and observe a large bump in soundness, without any drop in completeness. While "cheating" is always possible (even if unintentionally) for any metric, we hope that our results inspire saliency method designers to compute maps for all labels. In this case we find, in Figure 4 (right), that looking at worst case completeness of incorrect labels (all except model prediction) gives a measure of the "effort" a method employs in generating masks for all labels with probability at least 0.01. As evident methods that "cheat", like Phang et al. [31] and "Ours (cheating)", achieve a bad "best effort" score.

Other metrics. Table 1 suggests that our method can achieve comparable performances on most existing intrinsic metrics. This suggests that good completeness and soundness scores are achievable without compromising on other measures. Deletion metric is one where our method falls short, likely due to our method not employing an SDR objective as in Dabkowski and Gal [9]. Although we note that "Random" baseline performs quite well on the Deletion metric, which suggests that deletion metric does not provide much signal on these datasets. Further investigation on the good performance of "Centered Gaussian" baseline can be found in Appendix C.

Our settings that works best on other metrics are those without upsampling ($s = 1$) or lower TV ($\lambda_{TV} = 0.001$). Soundness is the only metric that strongly justifies the upsampling and TV "tricks", thus verifying its utility. Our method with $s = 4, \lambda_{TV} = 0.01$ demonstrates that good performance can be obtained on all metrics simultaneously, with only little price to pay.

Limitations of our work. Completeness and soundness do not involve or address human interpretability of explanations. This can be a strength when considering intrinsic evaluations, but is a weakness when considering extrinsic evaluations. Additionally, though completeness and soundness should be taken into account while designing a saliency method, they are not meant as a replacement for other evaluation methods, which can provide additional information on saliency map quality. For example, the axioms and sanity checks for saliency discussed in Appendix E.1 should also be applied. Finally, at deployment time, there is a risk that these evaluation methods may be used as a "proof" to justify a saliency method is safe to be used in high-risk applications. Completeness and soundness should be taken as a sanity check rather than a proof a saliency method is safe for deployment.

7 Conclusions

Saliency explanations of ML models have proved nebulous and generated much discussion. By avoiding extrinsic considerations (e.g., human interpretability) and sticking to intrinsic notions such as completeness and soundness, this paper has tried to give a rigorous notion of saliency that is intrinsic to the deep net, while avoiding the noisiness in earlier intrinsic ideas that motivated methods like Gradient$\odot$Input and Shapley Values. Other new contributions include clarifying in the intrinsic view the role of TV regularization and other tricks (it hurts completeness slightly but greatly improves soundness); a simple saliency method (Section 4) for producing mask-based explanations that was designed solely with intrinsic considerations and yet has performance competitive with good existing methods, and sometimes better (Table 1). Our experiments suggest soundness provides a new dimension to evaluate saliency methods on, and that good performance on it can be achieved without compromising on other metrics. Note that soundness needs to be considered only while designing and evaluating the method, not at deployment.

Evaluating soundness requires looking at maps for all labels instead of just the top label, and thus seems relevant for object *localization* in images, and for classification in the wild — where multiple objects appear in an image. This benefit is hinted in our evaluation on a prior small dataset (see Appendix B) and deserves further exploration. It may help with certain distribution shifts as well. While this work studies soundness in the context of mask-based saliency methods and evaluations, the concept of soundness could be applicable for the idea of saliency and interpretability in more generality.

Acknowledgements. We thank Ruth Fong for feedback on an earlier draft of this paper. We are also grateful to the valuable comments from various anonymous reviewers that helped improve the paper. This work is supported by funding from NSF, ONR, Simons Foundation, DARPA and SRC.

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: 9505–9515, 2018.

[2] Chirag Agarwal and Anh Nguyen. Explaining image classifiers by removing input features using generative models. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[3] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

[4] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv e-prints*, pages arXiv–2006, 2020.

[5] Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers. In *International Conference on Artificial Neural Networks*, pages 63–71. Springer, 2016.

[6] Lukas Brunke, Prateek Agrawal, and Nikhil George. Evaluating input perturbation methods for interpreting cnns and saliency map comparison. In *European Conference on Computer Vision*, pages 120–134. Springer, 2020.

[7] Brandon Carter, Jonas Mueller, Siddhartha Jain, and David Gifford. What made you do this? understanding black-box decisions with sufficient input subsets. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 567–576. PMLR, 2019.

[8] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. In *International Conference on Learning Representations*, 2018.

[9] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6970–6979, 2017.

[10] Nicola De Cao, Michael Sejr Schlichtkrull, Wilker Aziz, and Ivan Titov. How do decisions emerge across layers in neural models? interpretation with differentiable masking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3243–3255, 2020.

[11] 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.

[12] Daniel Fryer, Inga Strümke, and Hien Nguyen. Shapley values for feature selection: The good, the bad, and the axioms. *ICLR*, 2021.

[13] Jindong Gu, Yinchong Yang, and Volker Tresp. Understanding individual decisions of cnns via contrastive backpropagation. In *Asian Conference on Computer Vision*, pages 119–134. Springer, 2018.

[14] Xin Hong, Pengfei Xiong, Renhe Ji, and Haoqiang Fan. Deep fusion network for image completion. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2033–2042, 2019.

[15] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *NeurIPS*, 2019.

[16] Jeremy Howard. imagenette, 2019. *URL https://github. com/fastai/imagenette*, 2020.

[17] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. Abduction-based explanations for machine learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1511–1519, 2019.

[18] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. On relating explanations and adversarial examples. 2019.

[19] Ashkan Khakzar, Soroosh Baselizadeh, Saurabh Khanduja, Christian Rupprecht, Seong Tae Kim, and Nassir Navab. Improving feature attribution through input-specific network pruning. *arXiv e-prints*, pages arXiv–1911, 2019.

[20] 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*, pages 174–182, 2021.

[21] Beomsu Kim, Junghoon Seo, Seunghyeon Jeon, Jamyoung Koo, Jeongyeol Choe, and Taegyun Jeon. Why are saliency maps noisy? cause of and solution to noisy saliency maps. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4149–4157. IEEE, 2019.

[22] Junho Kim, Seongyeop Kim, Seong Tae Kim, and Yong Man Ro. Robust perturbation for visual explanation: Cross-checking mask optimization to avoid class distortion. *IEEE Transactions on Image Processing*, 31:301–313, 2021.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. Captum: A unified and generic model interpretability library for pytorch, 2020.

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[26] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf`.

[27] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777, 2017.

[28] Jan Macdonald, Stephan Wäldchen, Sascha Hauch, and Gitta Kutyniok. A rate-distortion framework for explaining neural network decisions. *arXiv e-prints*, pages arXiv–1905, 2019.

[29] Jesse Mu and Jacob Andreas. Compositional explanations of neurons. 2020.

[30] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[31] Jason Phang, Jungkyu Park, and Krzysztof J Geras. Investigating and simplifying masking-based saliency methods for model interpretability. *arXiv preprint arXiv:2010.09750*, 2020.

[32] Donald B Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[33] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.

[34] Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. *Explainable AI: interpreting, explaining and visualizing deep learning*, volume 11700. Springer Nature, 2019.

[35] Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. Restricting the flow: Information bottlenecks for attribution. In *International Conference on Learning Representations*, 2019.

[36] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *International Journal of Computer Vision (IJCV)*, 2019.

[37] Junghoon Seo, Jeongyeol Choe, Jamyoung Koo, Seunghyeon Jeon, Beomsu Kim, and Taegyun Jeon. Noise-adding methods of saliency map as series of higher order partial derivative. *2018 ICML Workshop on Human Interpretability in Machine Learning*, pages arXiv–1806, 2018.

[38] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017.

[39] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv e-prints*, pages arXiv–1706, 2017.

[40] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International Conference on Machine Learning*, pages 9269–9278. PMLR, 2020.

[41] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[42] Saeid Asgari Taghanaki, Mohammad Havaei, Tess Berthier, Francis Dutil, Lisa Di Jorio, Ghassan Hamarneh, and Yoshua Bengio. Infomask: Masked variational latent representation to localize chest disease. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 739–747. Springer, 2019.

[43] Richard Tomsett, Dan Harborne, Supriyo Chakraborty, Prudhvi Gurram, and Alun Preece. Sanity checks for saliency metrics. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 6021–6029, 2020.

[44] Mengjiao Yang and Been Kim. Benchmarking attribution methods with relative feature importance. *arXiv e-prints*, pages arXiv–1907, 2019.

[45] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33, 2020.

[46] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

[47] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018.

[48] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

[49] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2131–2145, 2018.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Main contribution of this work is to provide suggestions and sanity checks to improve evaluation systems that aim to open up and explain the black-box that is deep learning. Doing so can help us avoid unintended (or adversarial) negative impacts. Just like any other work, readers and users should be wary of treating this idea as a magic pill that solves all issues. One negative impact could be if these evaluation systems are treated as "proofs", rather than sanity checks, that systems are working well and are safe to deploy.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Intuitions and Implementations of Procedures to Find Masking Explanations

As introduced in Section 3.1, for evaluation we may interest in random binary masks due to its connection to AUC, but in our method for finding masking explanations we only focus on deterministic masks. Given a network f, image $x \in \mathbb{R}^{c \times hw}$ and class a, we wish to find a binary mask $M \in \{0,1\}^{hw}$ such that when the part of x on M is superimposed onto a "distractor" $\bar{x} \sim \mathcal{X}$ (randomly sampled from the train set) as $\tilde{x} = M \odot x + (1-M) \odot \bar{x}$, the output probability of the model $f(\tilde{x}, a)$ is high for the class a.

As in Section 3.2 we compute the average probability assigned to class a over the sampling of the distractor $\bar{x}$, i.e. we are interested in making $\mathbb{E}_{\bar{x} \sim \mathcal{X}}[f(\tilde{x}, a)]$ high. To avoid the hard problem of optimizing over the hypercube $\{0,1\}^{hw}$, a typical strategy (also employed in prior work) is to relax the domain of M to be $[0,1]^{hw}$. Since we do not wish to learn masks of very large size, a ℓ_1 norm penalty on M (corresponding to size of the mask), leading to the following natural objective function[11]

$$L(M) = \mathbb{E}_{\bar{x} \sim \mathcal{X}}\left[-\log(f(M \odot x + (1-M) \odot \bar{x}, a))\right] + \lambda_1 \|M\|_1 \tag{5}$$

However most masking-based methods employ additional "tricks" in order to avoid "artifacts" in the produced saliency maps, like Total-Variation (TV) penalty [11] and upsampling of the mask from a lower resolution one [30]. We also employ the same strategy by learning a low-resolution mask at scale s, $M \in \mathbb{R}^{hw/s^2}$, to minimize the following

$$L(M) = \mathbb{E}_{\bar{x} \sim \mathcal{X}}\left[-\log(f(M^{\times s} \odot x + (1-M^{\times s}) \odot \bar{x}, a))\right] + \lambda_{TV} TV(M^{\times s}) + \lambda_1 \|M^{\times s}\|_1 \tag{6}$$

where $M^{\times s} \in \mathbb{R}^{hw}$ is obtained by upsampling M by a factor of $s \in \{1, 4\}$ via bilinear interpolation.

While the motivation cited for these "trick" is to avoid artifacts, it is not clear whether artifacts are a bad thing, since they might be relevant to the net's decision-making. Indeed, we show that while TV penalty or upsampling does produce better looking masks, they lead to a drop in the *completeness metric*. However we show that adding such tricks leads to significant improvement in the *soundness metric,* thus providing a novel justification for the use of such tricks, beyond just the heuristic argument of getting rid of artifacts. In Section 5 we also provide theoretical justification for why TV penalty can help with soundness, even for the simple case of linear predictors on non-image data.

We optimize the objective in Equation (4) by parametrizing M as a sigmoid of real valued weights $W \in \mathbb{R}^{hw/s^2}$, i.e. $M = \sigma(W)$, and run Adam [23] optimizer for 2000 steps with learning rate 0.05 and by sampling 10 distractor images at every step, for different values of λ_{TV} and upsampling factor s.

B Practical Benefits of Completeness of all labels for Images of Multiple Objects

Images may have multiple plausible labels. Figure 5 and Figure 6 show images where the classifier net gave high probability to a single label even though multiple objects were present. Our saliency method can produce different and meaningful masks for all labels that are valid. Extending the notion of soundness for multi-label settings is an open question.

In Figure 5, images that previously used in Gu et al. 13 can have both elephants and zebras present, but it may not be always clear from the model output if there is such a case, since the model can be much more confident on one label, e.g., elephant, than one would expect it to be. For this reason, finding masking explanations validating other labels, e.g., zebra, could provide more information on how the model makes the prediction.

We also use the relabeling provided by Beyer et al. [4] to select ImageNet validation images with two true labels. Our results may be seen in Figure 6.

[11] A standard way to maximize probability is to minimize the negative log probability

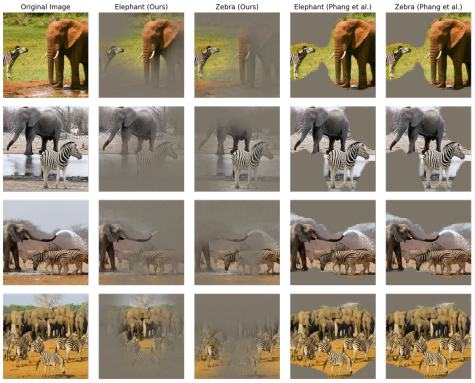


Figure 5: Images containing both elephant(s) and zebra(s), and the corresponding masked ones generated by our method and the best-performing CA model in Phang et al. [31]. The masks by Phang et al. [31] are identical for different labels, and contains both elephant and zebra. In contrast, our method outputs decent masks for elephant and zebra accordingly. For more examples please see Figure 6 in Appendix C.

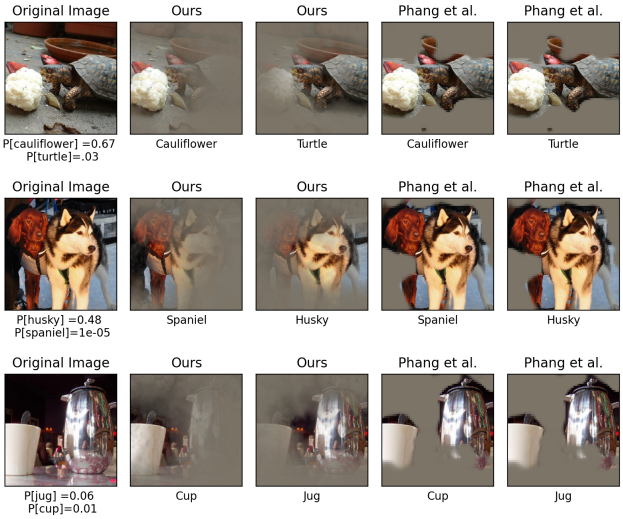


Figure 6: Our masks and masks by Phang et al. [31] for ImageNet images with two ground truth labels. The best-performing CA model in Phang et al. [31] was used. First (leftmost) column depicts original image with original model probabilities for each ground truth class below the image. Next two columns depict our masks with target label below the image. Final two columns depict the masks for Phang et al. [31]. The masks by Phang et al. [31] are identical for different labels, and contains both classes. In contrast, our method outputs descent masks for each class accordingly.

Table 2: Worst case completeness and worst case soundness ($\epsilon_1 = 0.01, \epsilon_2 = 0.1$) for different λ_{TV} in CIFAR-10. The best one in each row is marked bold.

λ_{TV}	0	0.001	0.01	0.1
Completeness	**0.99**	0.99	0.89	0.80
Worst soundness	0.10	0.10	0.11	**0.19**

Table 3: Deletion, insertion, saliency metric, worst case completeness and soundness on Imagenette-Corner. Our method with upsampling factor $s = 4$, TV penalty $\lambda_{TV} = 0.01$ achieves the best completeness and soundness. (↑ indicates higher is better.) The numbers for Centered Gaussian drop compared to Imagenette.

	Deletion ↓	Insertion (gray) ↑	Saliency Metric ↓	Worst Case Completeness ↑	Worst Case Soundness ↑
Gradient $\odot$ Input	**0.13**	0.34	**−0.25**	0.50	0.78
Dabkowski and Gal [9]	0.21	0.33	**−0.25**	0.52	0.78
Fong and Vedaldi [11]	**0.15**	0.53	**−0.25**	0.53	0.48
Phang et al. [31]	0.33	**0.69**	**−0.25**	**0.68**	**0.82**
Ours	0.36	**0.64**	**−0.25**	**0.76**	**0.84**
Random	0.35	0.35	−0.24	0.36	0.64
Centered Gaussian	0.62	0.62	0.458	0.50	0.54

C Additional Results on CIFAR-10 and Imagenette

In this section, we show additional results for CIFAR-10 and Imagenette experiments.

1. In Table 2, we show the worst case completeness and worst case soundness on CIFAR-10 of our method with different TV penalty λ_{TV}. To showcase of generality of our definition of base metric, here we use a modified version of AUC metric where we set $s \sim \{0.2\dim(x), \ldots, 0.6\dim(x)\}$ in Definition 3.5.
2. We showcase the masks of ten randomly drawn images from Imagenette for different methods in Figure 7 and Figure 8. The masks in Figure 7 is generated for model predictions, while the masks in Figure 8 is generated for incorrect labels. Note there are some wired horizontal lines and shapes for some masks. These are caused by default tie breaking of pixels of masking value 1. We also tried tie breaking according to Centered Gaussian, which does not improve the performance of those methods.
3. We create a new dataset Imagenette-Corner based on Imagenette, where each image is a random corner of the original image[12]. We show the results on Imagenette-Corner in Table 3. The results show that the good performance of Centered Gaussian on Imagenette is likely due to the bias of datasets (that objects are centered with high probability). Saliency metric on Imagenette-Corner are the same for most of the methods, likely because of the coarse selection of hyperparameters as in Dabkowski and Gal [9].

D Experimental Details and Additional Experiments

In this section we expand upon the experiments in Section 6 and complement them with more experiments on the ImageNet, CIFAR-10 and CIFAR-100 datasets. For each of the datasets we test the following:

- **Visualization:** For various values of TV regularization (and upsampling for ImageNet), we visualize the mask and also what part of the image a sparse version of the mask highlights. We do so for masks learned for the correct label and also for the second most probable label as predicted by the model. The common trend is that while TV regularization (and upsampling) make the masks more human interpretable, it also makes it harder to find a good mask for the incorrect label, thus improving soundness.

[12] On ImageNet or Imagenette, the common way to process image is first resizing it to 256×256 pixels, and then crop the center 224×224 pixels. In Imagenette-Corner, instead of center cropping, we take one of the four 180×180-pixel corners and resize it to 224×224 pixels.

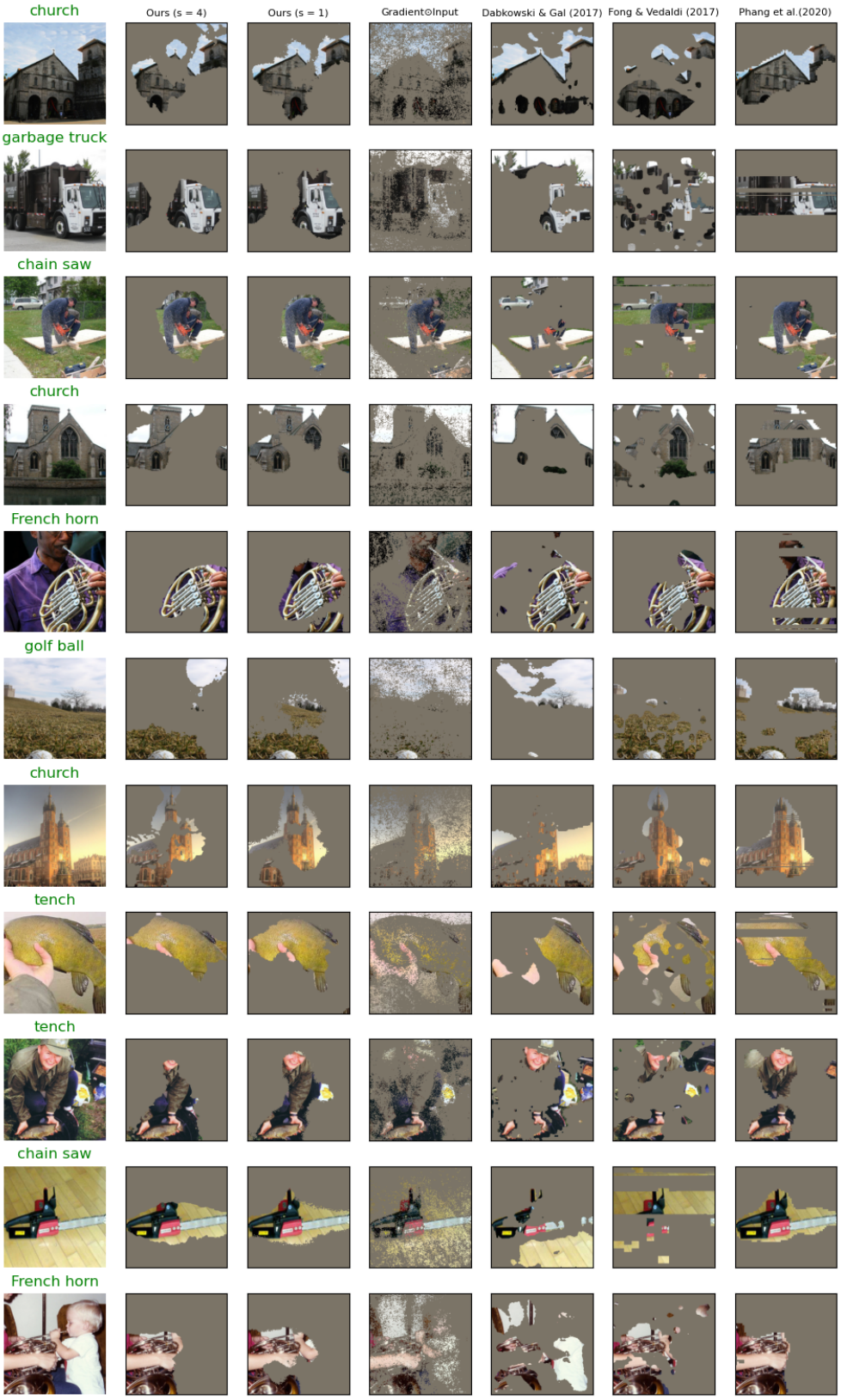


Figure 7: Masks of 10 Imagenette examples generated by different methods. Above each original image is the corresponding correct label. 30% of the pixels are retained, and the rest pixels are filled with grey.

Figure 8: Masks of 10 Imagenette examples generated by different methods for incorrect labels. Above each original image is the target incorrect label. 30% of the pixels are retained, and the rest pixels are filled with grey.

Table 4: Completeness and soundness for a ResNet-164 trained model on CIFAR-10, as defined in Equation (1). Each column represents mask learned using our procedure, with different TV regularization strengths (0.0, 0.001, 0.01, or 0.1). "Gray" indicates pixels were grayed during AUC calculation and "Random image" indicates they were replaced with other images.

Gray	TV = 0.0	TV= 0.001	TV= 0.01	TV= 0.1
Completeness (α)	0.92	0.91	0.83	0.77
Soundness (β)	0.17	0.18	0.39	0.57
Random image	TV = 0.0	TV= 0.001	TV= 0.01	TV= 0.1
Completeness (α)	0.86	0.85	0.77	0.70
Soundness (β)	0.20	0.21	0.44	0.62

- **AUC curve:** We plot the output model probability for a masked input as more pixels from the original image are selected. The 4 plots denote replacing remaining pixels with gray pixels or pixels from a random image, and masks to fit the correct or incorrect labels, i.e. most probable and second most probable labels. Again, we find the TV regularization and upsampling help with soundness; i.e. inability to find mask for the second most probable label. For mask M, if $\bar{M}(p)$ denotes the discrete mask with top p fraction of the pixels from M picked. We plot $\mathbb{E}_x\left[\mathbb{E}_{x'\sim\Gamma}[f(\bar{M}(p)\odot x + (1-\bar{M}(p))\odot x', a)]\right]$ v/s p, where Γ is either a random image or a gray image, and a is either the correct label for x or the second best label. We note that replacing with gray and random image lead to similar looking plots, with the probability estimate of random image being more pessimistic. This justifies the motivation for our procedure that learns a mask to solve a "harder task" of random image replacement.
- **Completeness/soundness:** We evaluate the completeness and soundness scores, as defined in Equation (1) and Definition 3.7. In particular for any input x, we only evaluate the scores for the top model prediction a and the second most probable label a' and report the worst case completeness and soundness for these 2 labels. For all experiments in this section, we use $\epsilon_1 = 0$ and $\epsilon_2 = 0.1$ (from Equation (1)).
- **Intrinsic metrics:** We evaluate our masks on other intrinsic metrics from prior work, and compare to baseline saliency methods. Our baselines include Gradient $\odot$ Input [38], Smooth-Grad [39], Real Time Saliency [9] (for ResNet-50 on ImageNet), and Random indicating a random Gaussian mask as a control. We use Captum [24] for Gradient $\odot$ Input and Smooth-Grad implementations and the original author code[13] for Real Time Saliency. When calculating the Saliency Metric (SM) [9] we tune the threshold δ on a holdout set of size 100 with δ between 0 and 5 in increments of 0.2 as in prior work.

  For the saliency method of Fong and Vedaldi [11] that we only used on the Imagenette, we adapt the most popular implementation on GitHub[14]. The implementation contains minor deviations from the original paper as described on its main page. For Phang et al. [31], we used their best CA model pretrained and provided in original author code[15].

D.1 CIFAR-10 Experiments

We also run our method from Section 4 on the CIFAR-10 dataset using a pretrained ResNet-164 architecture[16]. For all experiments we learn a mask $M \in \mathbb{R}^{32\times 32}$, thus using a scaling factor of $s = 1$ (no upsampling). We train masks for 1600 images that were correctly classified by the pretrained ResNet-164 using regularization parameter $\lambda_{TV} \in \{0, 0.001, 0.01, 0.1\}$. We use a (fixed) L1 regularization value of .001 for all masks.

We visualize the masks learned for the correct label in Figure 10a and in Figure 10b we visualize the same for the second best label predicted by the ResNet-164 model. We also visualize the masks for all labels for some randomly picked images in Figure 11 to demonstrate the commonness of artifact,

[13] `https://github.com/PiotrDabkowski/pytorch-saliency`

[14] `https://github.com/jacobgil/pytorch-explain-black-box`

[15] `https://github.com/zphang/saliency_investigation`

[16] `https://github.com/bearpaw/pytorch-classification`. The ResNet-110 model in this repository is actually a ResNet-164 model.

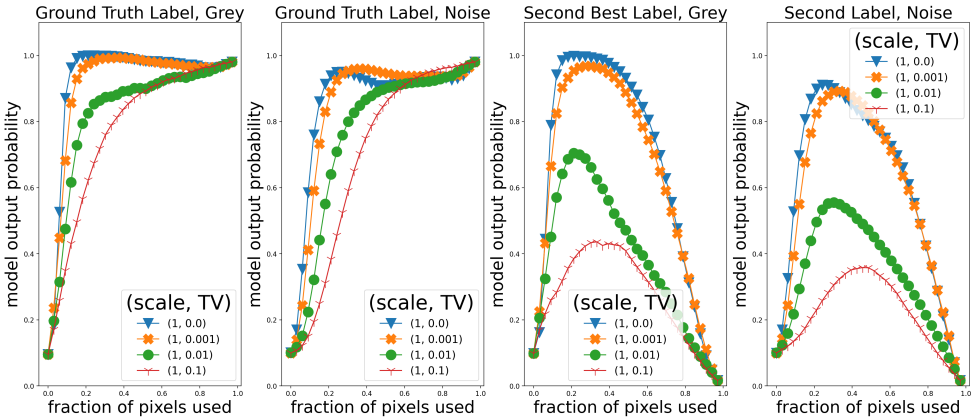


Figure 9: [CIFAR-10] AUC curves with as the fraction of pixels retained from the original images based on the mask varies from 0 to 1.0 on the X-axis. The probabilities assigned by the model (averaged over 1600 images) on the Y-axis. **Left:** Mask learned for ground truth label, probabilities for ground truth label while replacing remaining pixels with gray. **Center Left:** Mask learned for ground truth label, probabilities for ground truth label while replacing remaining pixels with other image pixels. **Center Right:** Mask learned for second best label, probabilities for second best label while replacing remaining pixels with gray. **Right:** Mask learned for second best label, probabilities for second best label while replacing remaining pixels with other image pixels. We see that increasing TV regularization results in only a mild drop in completeness, but significantly improves soundness.

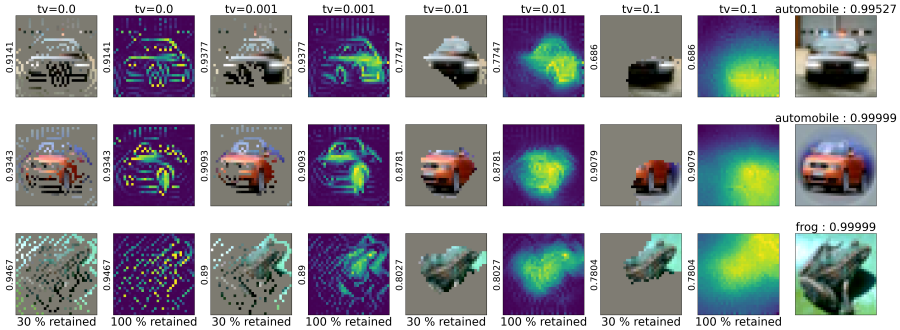
Table 5: Performance of our method on CIFAR-10 and some baselines on various intrinsic saliency metrics proposed in prior work. Downarrow (uparrow) means lower (higher) is better. We find that while both our masks (learned with and without TV) have very good performance on the insertion metric. The deletion and saliency metrics are uninformative in this case, since all methods are as good (or worse) compared to a random mask.

	Gradient $\odot$ Input	Our method ($\lambda_{TV} = 0.01$)	Our Method ($\lambda_{TV} = 0$)	Smooth-Grad saliency	Random
Deletion $\downarrow$	0.32	0.37	0.59	0.31	0.26
Insertion (blur) $\uparrow$	0.60	0.88	0.94	0.66	0.36
Insertion (gray) $\uparrow$	0.51	0.83	0.92	0.55	0.26
Saliency Metric $\downarrow$	0.22	0.22	0.22	0.23	0.22

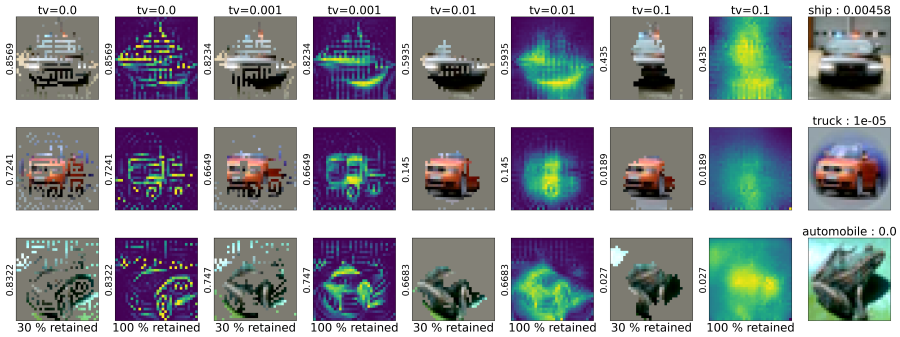
especially for the incorrect labels. The AUC curves in Figure 9 suggest a similar trend to that of Imagenette, adding TV regularization results in only a mild drop in completeness, but significantly improves soundness. Evaluation of our masks, compared to some gradient baselines, on intrinsic metrics can be found in Table 5. We report the completeness and soundness results for CIFAR-10 in Table 4 for TV values in $(0.0, 0.001, 0.01, 0.1)$ calculated using a ResNet-164 model.

D.2 CIFAR-100 Experiments

We run the same experiment for CIFAR-100 using the corresponding ResNet164 model. We visualize the masks learned for the correct label in Figure 13a and in Figure 13b we visualize the same for the second best label predicted by the ResNet-164 model. The AUC curves in Figure 12 suggest a similar trend to that of Imagenette, adding TV regularization results in only a mild drop in completeness, but significantly improves soundness. Evaluation of our masks, compared to some gradient baselines, on intrinsic metrics can be found in Table 7. We place a downarrow after the name of the metric to indicate a lower value is considered better and an uparrow when a higher value is considered better. We evaluate on a randomly selected subset of 1600 data points where the model had correct top 1 accuracy. We report the completeness and soundness results for CIFAR-100 in Table 6 for TV values in $(0.0, 0.001, 0.01, 0.1)$ calculated using a ResNet-164 model.



(a)



(b)

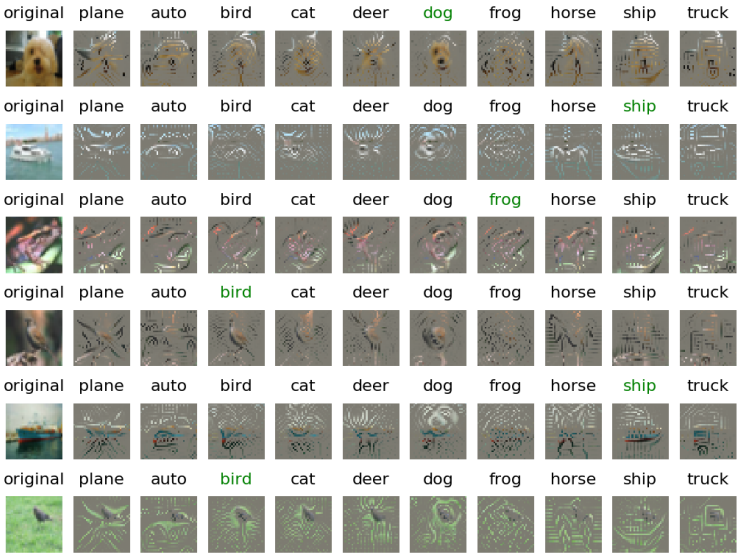
Figure 10: Details in Appendix D.1 **Panel 10a** Masks learned for the correct label of CIFAR-10 images using the procedure outlined in Section 4 with ResNet-164. Columns (1,3,5,7) depict masked images at 30 (retained) % mask sparseness. Columns (2,4,6,8) depict the original mask. TV values shown above. Original image shown in rightmost column. Model probability of correct label for masked images on y axis. **Panel 10b** Masks leared for the second most probable label of CIFAR-10 images using the procedure outlined in Section 4 on ResNet-164. Columns (1,3,5,7) depict masked images at 30 % mask sparseness. Columns (2,4,6,8) depict the original mask.TV values shown above. Original image shown in rightmost column. Model probability of second best label for masked images on y axis.

Table 6: Completeness and soundness for a ResNet-164 trained model on CIFAR-100, as defined in Equation (1). Each column represents mask learned using our procedure, with different TV regularization strengths (0.0, 0.001, 0.01, or 0.1). "Gray" indicates pixels were grayed during AUC calculation and "Random image" indicates they were replaced with other images.

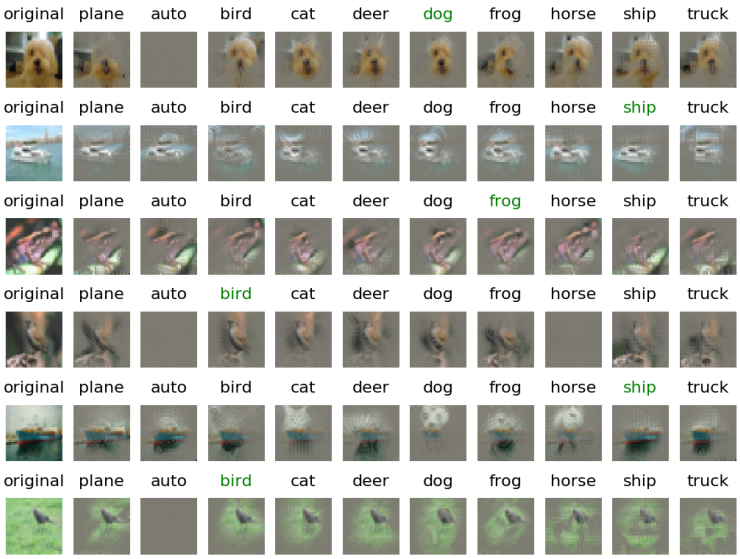
Gray	TV = 0.0	TV= 0.001	TV= 0.01	TV= 0.1
Completeness (α)	0.80	0.74	0.64	0.55
Soundness (β)	0.28	0.34	0.58	0.75
Random image	TV = 0.0	TV= 0.001	TV= 0.01	TV= 0.1
Completeness (α)	0.67	0.66	0.61	0.54
Soundness (β)	0.35	0.40	0.61	0.78

D.3 Experiments on ImageNet

In Figure 14a we depict the the masks for TV values in $\{0.0, 0.01\}$ for a ResNet-18 model on ImageNet for the ground truth label and in Figure 14b we depict the same for the second best label. We also experiment with the effect of upsampling (US) the mask, whereby we learn a mask of size (56,56) and upsample to size (224,224). We use a fixed L1 regularization value of 2e-5. We depict our results on ImageNet and ResNet-18 in Table 9



(a) Our method with no TV regularization



(b) Our method with TV regularization $\lambda_{TV} = 0.01$

Figure 11: A demonstration of artifacts created by masking on CIFAR-10. Pixels (partially) masked out are filled with gray based on the fractions they are masked out. Masks generated without or only with low level regularization can easily produce artifacts. It is more common and/or severe for the incorrect label than correct label.

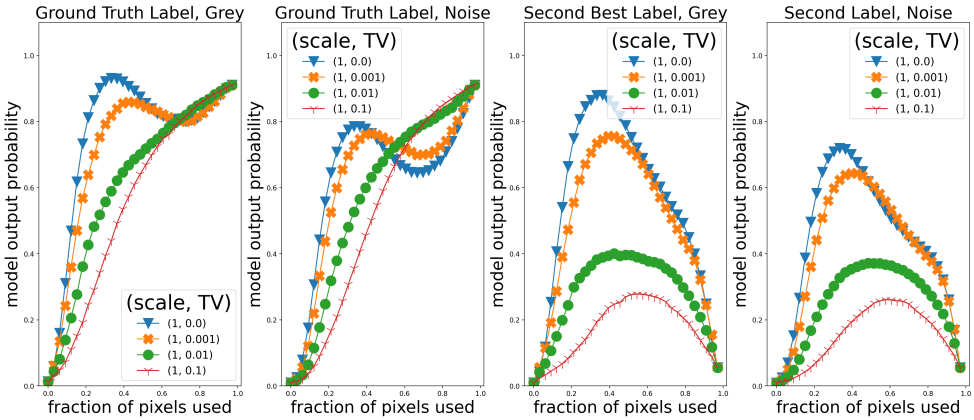


Figure 12: [CIFAR-100] AUC curves with as the fraction of pixels retained from the original images based on the mask varies from 0 to 1.0 on the X-axis. The probabilities assigned by the model (averaged over 1600 images) on the Y-axis. **Left:** Mask learned for ground truth label, probabilities for ground truth label while replacing remaining pixels with gray. **Center Left:** Mask learned for ground truth label, probabilities for ground truth label while replacing remaining pixels with other image pixels. **Center Right:** Mask learned for second best label, probabilities for second best label while replacing remaining pixels with gray. **Right:** Mask learned for second best label, probabilities for second best label while replacing remaining pixels with other image pixels. We see that increasing TV regularization results in only a mild drop in completeness, but significantly improves soundness.

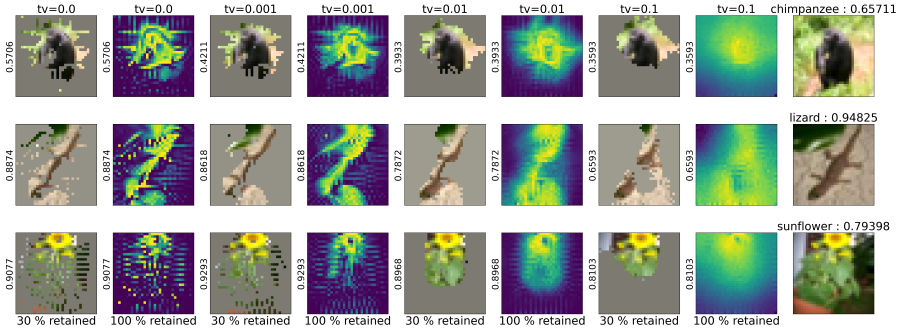
Table 7: Performance of our method on CIFAR-100 and some baselines on various intrinsic saliency metrics proposed in prior work. Downarrow (uparrow) means lower (higher) is better. We find that while both our masks (learned with and without TV) have very good performance on the insertion metric. The deletion and saliency metrics are uninformative in this case, since all methods are as good (or worse) compared to a random mask.

	Gradient $\odot$ Input	Our method ($\lambda_{TV} = 0.01$)	Our Method ($\lambda_{TV} = 0$)	Smooth-Grad saliency	Random
Deletion $\downarrow$	0.10	0.17	0.10	0.29	0.11
Insertion (blur) $\uparrow$	0.36	0.71	0.82	0.39	0.20
Insertion (gray) $\uparrow$	0.27	0.62	0.76	0.29	0.11
Saliency Metric $\downarrow$	0.77	0.77	0.77	0.79	0.77

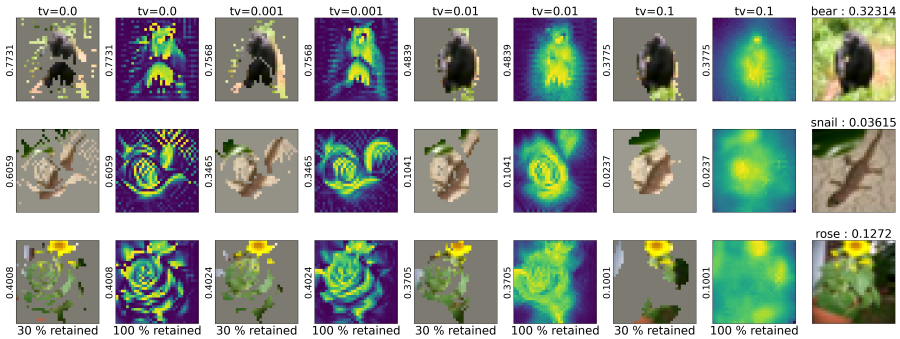
For the deletion metric, we note that most methods have comparable or worse performance than the random mask, which suggests that the metric does not give us much signal about the goodness of the saliency maps. On the insertion metric, we find that mask learned by not adding the TV penalty significantly beats other methods. The mask learned using TV penalty, on the other hand, has impressive performance on both the insertion AUC and saliency metric (SM).

Completeness and Soundness on ImageNet and ResNet-18. We report our results in Table 8 for TV values in $(0, 0.01)$ for both graying (Gray) and replacing with other image pixels (Random image). Additionally, we investigate the effect of upsampling (US) where we derive a (56,56) and upsample by a factor of 4 to a $(224, 224)$ mask.

Effect of ensembling. In order to investigate the effect of ensembling we plot maps in Figure 16 as we vary the number of maps that are ensembled over as $K \in \{1, 2, 4\}$, where we learn multiple masks such that each of them individually validate the label, but are as disjoint as possible. We do not upsample (using a scale of 1.0) and we use a fixed L1 regularization of 2e-5 and a fixed TV regularization of 0.0.



(a)



(b)

Figure 13: Details in Appendix D.2 **Panel 13a** Masks learned for the correct label of CIFAR-100 images using the procedure outlined in Section 4 on ResNet-164. Columns (1,3,5,7) depict masked images at 30 (retained) % mask sparseness. Columns (2,4,6,8) depict the original mask. TV values shown above. Original image shown in rightmost column. Model probability of correct label for masked images on y axis. **Panel 13b** Masks learned for the second most probable label of CIFAR-100 images using the procedure outlined in Section 4 on ResNet-164. Columns (1,3,5,7) depict masked images at 30 % mask sparseness. Columns (2,4,6,8) depict the original mask.TV values shown above. Original image shown in rightmost column. Model probability of second best label for masked images on y axis.

Table 8: Completeness and soundness for a ResNet-18 model on ImageNet as defined in Equation (1). Each column contains a represents mask learned using our procedure, with our without upscaling and different TV regularization strengths. "Gray" indicates pixels were grayed during AUC calculation and "Random image" indicates they were replaced with other images. No US indicates the full (224,224) mask was derived and US indicates a $(56, 56)$ mask was derived then upsampled by a factor of 4. TV indicates a TV regularization value of 0.0 or 0.01.

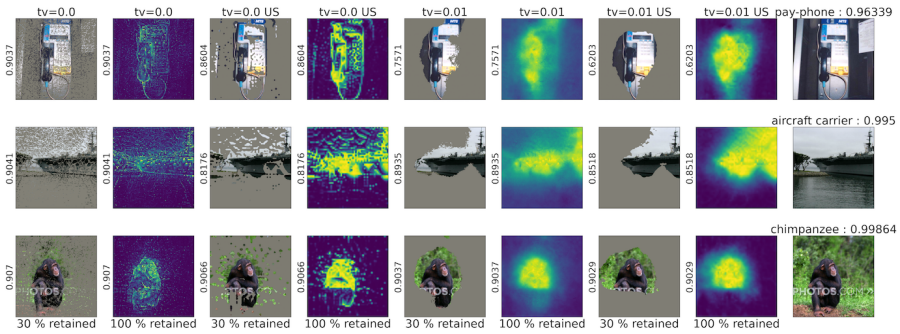
Gray	TV = 0.0	TV = 0.01	US TV = 0.0	US TV = 0.01
Completeness (α)	0.97	0.76	0.87	0.71
Soundness (β)	0.19	0.70	0.38	0.75
Random image	TV = 0.0	TV = 0.01	US TV = 0.0	US TV = 0.01
Completeness (α)	0.89	0.61	0.74	0.59
Soundness (β)	0.25	0.83	0.52	0.86

D.4 Effect on Sanity Checks

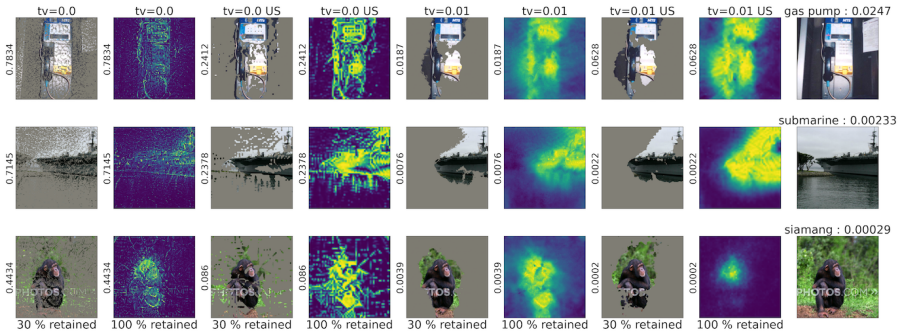
Inspired by [1] we randomize the last layer of a ResNet-18 network and visually inspect the resulting saliency maps in Figure 15. We find that the maps appear less coherent than those of a pre-trained model. We use a fixed L1 regularization of 2e-5 and depict maps with and without upsampling (US) at TV values of $(0, 0.01)$.

Table 9: Performance of our method on ImageNet and ResNet-18 model and some baselines on various intrinsic saliency metrics proposed in prior work. We find that while both our masks (learned with and without TV) have very good performance on the insertion metric, the mask learned with TV has much better performance on the saliency metric. The deletion metric is uninformative in most cases, since most methods are as good (or worse) compared to a random mask.

	Gradient $\odot$ Input	Our method ($\lambda_{TV} = 0.01$)	Our Method ($\lambda_{TV} = 0$)	Smooth-Grad saliency	Random
Deletion $\downarrow$	0.10	0.13	0.21	0.08	0.13
Insertion (blur) $\uparrow$	0.44	0.79	0.85	0.51	0.31
Insertion (gray) $\uparrow$	0.30	0.67	0.92	0.35	0.13
Saliency Metric $\downarrow$	0.31	0.15	0.32	0.32	0.32



(a)



(b)

Figure 14: Details in Appendix D.3. US stands for upsampled mask, where we derive a (56,56) mask and interpolate to (224,224). **Panel 14a** Masks learned for the correct label of ImageNet images using the procedure outlined in Section 4 on ResNet-50. Columns (1,3,5,7) depict masked images at 30 (retained) % mask sparseness. Columns (2,4,6,8) depict the original mask. TV values shown above. Original image shown in rightmost column. Model probability of correct label for masked images on y axis. **Panel 14b** Masks learned for the second most probable label of ImageNet images using the procedure outlined in Section 4 on ResNet-50. Columns (1,3,5,7) depict masked images at 30 % mask sparseness. Columns (2,4,6,8) depict the original mask.TV values shown above. Original image shown in rightmost column. Model probability of second best label for masked images on y axis. We find, unsurprisingly, that adding TV regularization and upsampling make the mask more continuous and "human interpretable" and, more importantly, make it harder to find masks that can get high probability for the second best label, thus ensuring higher soundness.

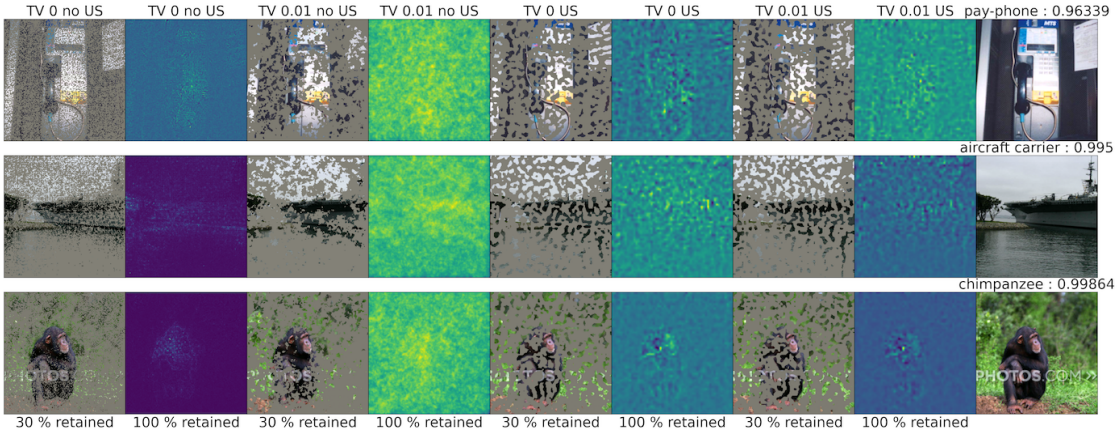


Figure 15: Results of randomizing the last layer of a ResNet-18 model on ImageNet data for the procedure described in Section 4. US indicates a $(56, 56)$ map was learned and upsampled to $(224, 224)$. We find the maps of this randomized network are less visually coherent than the analogous maps of a pre-trained model.

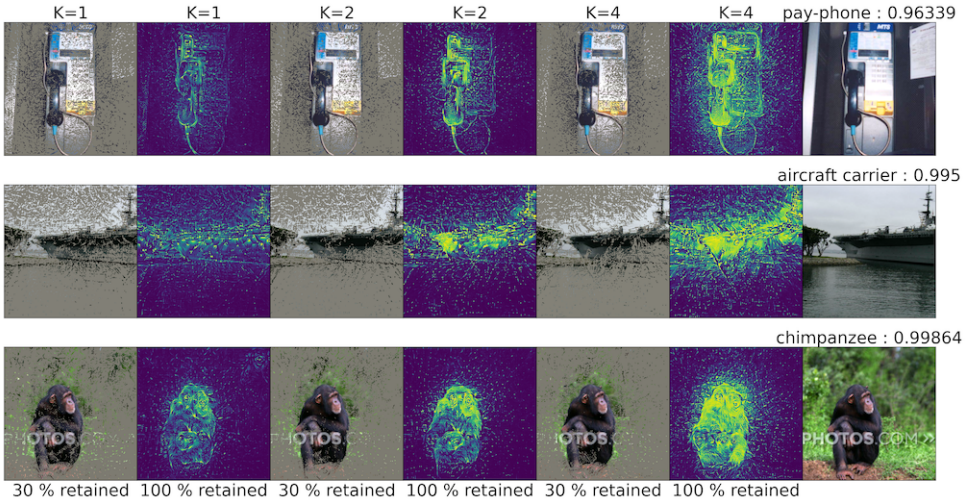


Figure 16: **Effect of ensembling** Partial statistical assignments validating the correct label of ImageNet and ResNet-18 images as we vary K, the number of maps. Details in Appendix D.3.

E Additional Background information

E.1 Additional related work

The pixel replacement strategy we used is closely related to hot deck imputation [32], where features may be replaced either by using the mean feature value (analogous to replacing with grey) or sampling from the marginal feature distribution (analogous to replacing with image pixels sampled from other training images). Some prior work [7] has found that mean imputation does not significantly affect model output on the beer aroma review dataset. On Imagenette, by contrast, we found that replacement strategy can matter.

Additional saliency evaluation tests. Adebayo et al. [1] proposed "sanity checks" for saliency maps. They note that if a model's weights are randomized, it has not learned anything, and therefore the saliency map should not look coherent. They also randomize the labels of the dataset and argue that the saliency maps for a model trained on this scrambled data should be different than the saliency maps for the model trained on the original data. We study the effect of model layer randomization on our method in Appendix section D.4 and find that randomizing the model weights does cause our

saliency maps to look incoherent. Tomsett et al. [43] also discover sanity checks for saliency metrics, finding that saliency evaluation methods can yield inconsistent results. They evaluate saliency maps on reliability, i.e. how consistent the saliency maps are. To measure a method's reliability, because access to ground truth saliency maps are not available, they use three proxies 1) inter-rater reliability, i.e. how whether a saliency evaluation metric is able to consistently rank some saliency methods above others, 2) inter-method reliability, which indicates whether a saliency evaluation metric agrees across different saliency methods, and 3)internal consistency reliability, which measure whether different saliency methods are measuring the same underlying concept.

Saliency axioms. Sundararajan et al. [41] identify two fundamental axioms, Sensitivity (versions a and b), and Implementation Invariance that attribution methods should satisfy. "An attribution method satisfies Sensitivity(a) if for every input and baseline that differ in one feature but have different predictions then the differing feature should be given a non-zero attribution." The definition of Sensitivity b is "If the function implemented by the deep network does not depend (mathematically) on some variable, then the attribution to that variable is always zero.". Implementation invariance simply states that if two networks are equivalent, the saliency maps for those two networks should be the same. These axioms are not captured by completeness and soundness and are good examples of why completeness and soundness cannot be used alone in evaluating saliency maps. Other prior work Carter et al. [7] argues that saliency explanations should be minimal Carter et al. [7] and find sufficient input subsets which are "minimal subsets of features whose observed values alone suffice for the same decision to be reached, even if all other input feature values are missing."

E.2 Saliency Methods

We give a partial list of extant saliency methods here. We broadly categorize explanations into three categories: Back-propagation based explanations, axiomatic methods, and masking methods. **Backpropagation based explanations** shape credit as it is propagated backwards through the neural network according to certain rules. These approaches include **Layerwise Relevance Propagation** [5] which satisfies completeness, **Rect-Grad** which thresholds internal neuron activations [21], and **DeepLIFT** which satisfies the summation to delta rule.

Axiomatic methods. Axiomatic methods decompose the ouput (typically the logit) according to certain axioms like fairness in Shapley based methods **SHAP** [26] and **conceptSHAP** [45]. We also include gradient based approaches like **Gradient** $\frac{\partial S}{\partial x}$ [3] which calculates the partial derivative of the logit with respect to the input. **Gradient** $\odot$ **Input** [38] $\frac{\partial S}{\partial x} \cdot x$, which elementwise multiplies the gradient explanation by the input, and **Grad-CAM** [36] which takes the gradient of the logit with respect to the feature map of the last convolutional unit of a DNN. **Smooth-Grad** [39], which averages the **Gradient** $\odot$ **Input** explanation over several noisy copies of the input $x + \eta$, where η is some Gaussian. The previous methods are intrinsic in the sense that they aim to explain the model decision. The last category of saliency maps, namely masking methods, also aim to explain the model decision, but they frequently aim to do so in a way that is interpretable by a human. Contrastive methods, such as contrastive layerwise propagation Gu et al. [13], also modify LRP by constructing class specific saliency maps, with the goal of object localization, i.e. in an image of an elephant and zebra, the saliency map for elephant should have high overlap with the elephant, and similarly for the corresponding map for zebra.

Masking Methods. Masking Methods are often evaluated using a pointing game or WSOL metric, which measures overlap with human labeled bounding boxes or explanations. These masking methods include techniques based on averaging over randomly sampled masks [30], optimizing over meaningful mask perturbations [11], and real time image saliency using a masking network [9]. Pixels that have been removed from the image by the mask may be replaced by graying out, by Gaussian blurring as in Fong and Vedaldi [11], or with infillers such as CA-GAN [46] used in Chang et al. [8], or DFNet [14]. De Cao et al. [10] find masks using differentiable masking. Taghanaki et al. [42] introduce a method that results in more accurate localization of discriminatory regions via mutual information.

Pruning and information theory Khakzar et al. [19] improve attribution via pruning. Schulz et al. [35] improve attribution by adding noise to intermediate feature maps.

Saliency and Boolean Logic. Previous work has also drawn connections between saliency and notions in logic. Ignatiev et al. [18] relates saliency explanations and adversarial examples by a generalized form of hitting set duality. Ignatiev et al. [17] develops a constraint-agnostic solution for computing explanations for any ML model. Macdonald et al. [28] develop a rate distortion explanation for saliency maps and prove a hardness result. Mu and Andreas [29] find a procedure for explaining neurons by identifying compositional logical concepts. Zhou et al. [49] describe network dissection, which provides labels for the neurons of the hidden representations. We are unaware of frameworks like Section 3.

Arguments about saliency. For discussion including pro/cons of various methods some starting points are Seo et al. [37] Fryer et al. [12] Gu et al. [13] Sundararajan and Najmi [40].

Phang et al. [31] We describe separately the masking procedure used by Phang et al. [31]. They begin by taking a pretrained model on ImageNet. The masker has access to the internal representations of the pre-trained model, and tries to maximize masked in accuracy and masked out entropy. They do not provide the ground truth label to the masker.

E.3 Saliency Evaluation Methods

Saliency evaluation methods attempt to evaluate the quality of a saliency map. Many interpret the heatmap values as a priority order of saliency. Extrinsic evaluation metrics include the **WSOL** metric, which aim to measure overlap of the saliency map with a human annotated bounding box and the **Pointing Game** metric proposed by Zhang et al. [47] in which a pixel count as a hit if it lies within a bounding box and a miss otherwise, and the metric is $\frac{\text{\# Hits}}{\text{\# Hits + \#Misses}}$. Other more intrinsic methods include early saliency evaluation techniques like **MorF** and **LerF** Samek et al. [33], which involve removing pixels either in the order of highest importance or lowest importance and observing the area of the resulting curve. **Insertion and Deletion Games** of Petsiuk et al. [30] uses this too. The deletion game measures the drop in class probability as important pixels are removed, while the insertion game measures the rise in class probability as important pixels are added. (Our AUC discussion in Section 3 relates to this.) **Remove and Retrain (ROAR)** is a saliency evaluation method proposed by Hooker et al. [15]. Input features are ranked and then removed according to a saliency map. A new model is trained on the modified training set, and a larger degradation in accuracy on the modified test set compared to the original model on the original test set is regarded as a better saliency method. (NB: retraining makes this a non-intrinsic method.) Previous work has also introduced datasets specifically designed to test saliency methods. **BAM** Yang and Kim [44] creates saliency maps by pasting object pixels from MSCOCO Lin et al. [25] into scene images from MiniPlaces Zhou et al. [48]. The **Saliency Metric** proposed by Dabkowski and Gal [9] thresholds saliency values above some α chosen on a holdout set, finds the smallest bounding box containing these pixels, upsamples and measures the ratio of bounding box area to model accuracy on the cropped image, $s(a, p) = \log(\max(a, 0.05)) - \log(p)$ where a is the area of the bounding box and p is the class probability of the upsampled image.

E.4 Saliency computations and underlying meanings of saliency

For simplicity this discussion assumes the datapoints are images and the classifier is a deep net. The heatmap in the saliency method is trying to highlight the contribution of individual pixels to the final answer. This is analogous to how a human may highlight relevant portions of the image with a plan. (Classic saliency methods in vision are inspired by studies of human cognition.) Saliency methods operationalize this intuitive definition in different ways, and we try to roughly categorise these as follows.

Variational interpretation. These interpret saliency in terms of effect on final output due to change in a single pixel –captured either via partial derivative of output with respect to pixel value (i.e., effect of infinitesimal change), or via change of output when this pixel is set to 0 or to a random (or "gray") value. Examples include **Gradient**, **Gradient** $\odot$ **Input** Shrikumar et al. [38], **Occlusion**

Credit attribution guided by gradient. These use the gradient to guide the assignment of saliency values. The gradient is interpreted as propagating values from the output to the input layer, and the values are partitioned/recombined at internal nodes of the net following some conservation principles. A key goal is to ensure *completeness*, which means that the sum of the attributions equal the logit value. Examples include **LRP**, **DeepLIFT** Shrikumar et al. [38], **Rect-Grad** Let a_i^l be the activation

of some node in layer l, and R_i^{l+1} be the backpropagated gradient up to a_i^l. Rect-grad replaces the vanilla chain rule, $R^l = \mathbf{1}[a_i > 0]$ with the rule that $R_i^l = \mathbf{1}[R_i^{l+1} a_i > \tau]$ for some threshold τ. Hence, during a backward pass preference is given to nodes with large margin.

Ensembling on top of above two ideas. Ensembling methods combine saliency estimates over multiple inputs an an attempt to reduce noise in the final map. Examples include **Smooth-Grad**, Occlusion based methods, etc. We also include **Shapley Values** in this list.

The Shapley value aims to fairly distribute credit among a coalition of N players. In the context of image saliency, each coordinate of the image input may be seen as a player, and the Shapley value computes $\sum_{S \subseteq N} \setminus \{i\} \frac{|S|!(n-|S|-1)!}{n!} (v(S \cup \{i\}) - v(S))$. It can be interpreted as the marginal contribution of player i, over all possible orderings of the coalition. In this sense, it can be seen as an ensembling method, as it averages over all possible random permutations.

Analysis of saliency methods. Previous work has analyzed ensembling methods like Smooth-grad, and found that it does not smooth the gradient Seo et al. [37]. They conclude that Smooth-Grad does not make the gradient of the score function smooth. Rather Smooth-grad is approximately the sum of a standard saliency map and higher order terms and the standard deviation of the Gaussian noise. It has also been found that Shapley values, despite having a uniqueness result, can differ in the way they depend on the model, data, etc Sundararajan and Najmi [40]. Fryer et al. [12] highlight several nuances that should be taken into account when considering Shapley values. They introduce Shapley values as averaging over submodels, and note that "the performance of a feature across all submodels may not be indicative of the particular performance of that feature in the set of optimal submodels.". They provide specific cases where satisfying the axioms of Shapley values works against the goal of feature selection.

F Clarifying benefit of TV regularization

This section illustrates shows how TV regularizers help ensure soundness even in a linear setting.

Let $\mathcal{S}$ be a dataset of labeled data $(\boldsymbol{x}, y)$ where the inputs are of unit norm and labels are binary, i.e., $\|\boldsymbol{x}\|_2 = 1$, $y \in \{\pm 1\}$. The model in question is a linear classifier $f(\boldsymbol{x}) := \operatorname{sgn}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle)$ parameterized by the weight vector $\boldsymbol{w} \in \mathbb{S}^{d-1}$, and it achieves the perfect accuracy on the set $\mathcal{S}$ with a margin $\gamma := \min_{(\boldsymbol{x},y) \in \mathcal{S}} y \langle \boldsymbol{w}, \boldsymbol{x} \rangle > 0$. We assume that the coordinates of $\boldsymbol{x}$ and $\boldsymbol{w}$ are uniformly bounded by $\frac{10}{\sqrt{d}}$, i.e., $\|\boldsymbol{x}\|_\infty \leq \frac{10}{\sqrt{d}}$, $\|\boldsymbol{w}\|_\infty \leq \frac{10}{\sqrt{d}}$ (10 can be changed to any other constant).

Kaifeng: I will change the wording here

Let Γ be the input modification process that sets all non-salient pixels to 0. We are interested in masking explanations with deterministic salient set, i.e., $\boldsymbol{m}$ assigns probability 1 to some salient set S. According to Section 3.2, a masking explanation validates label a if $\mathbb{E}_{\tilde{x} \sim \Gamma(x, \boldsymbol{m})}[\mathbb{1}_{[f(\tilde{\boldsymbol{x}})=a]}]$ is high. A simple calculation shows that this expectation equals to $\mathbb{1}_{[a \sum_{i \in S} w_i x_i > 0]}$, and thus the goal is to find S so that $a \sum_{i \in S} w_i x_i > 0$.

As we do not consider the full salient set informative, we are interested in masking explanations with size constraint $|S| = L$ for some $1 \leq L \leq d$. There is a simple saliency method that achieves this goal: Given an input $\boldsymbol{x}$ and a label $a \in \{\pm 1\}$, sort the coordinates according to $a w_i x_i$ and take the highest L coordinates as the salient set S.

It is easy to see that this method always produces S with $a \sum_{i \in S} w_i x_i > 0$. Letting $a = y$ proves the completeness. However, this method does not satisfy soundness: a salient set S with $a \sum_{i \in S} w_i x_i > 0$ can also be found for $a \neq y$!

Now we see how the TV constraint helps to ensure soundness (with good probability). A vector can be seen as a 1D image, and the TV of a salient set S can be defined by $\mathrm{TV}(S) := \sum_{i=1}^{d-1} \left| \mathbb{1}_{[i \in S]} - \mathbb{1}_{[i+1 \in S]} \right|$. For simplicity, we consider salient sets with TV at most 2. This means S is just an interval. Given the size and TV constraints $|S| = L$, $\mathrm{TV}(S) \leq 2$, it is easy to come out with the following saliency method: search over all the intervals of length L and if an interval S satisfies $a \sum_{i \in S} w_i x_i > 0$, return it as the salient set. Fortunately, this method does satisfy both completeness and soundness, which is justified by the following theorem.

Theorem F.1. *For any $(\boldsymbol{x}, y) \in \mathcal{S}$, if we shuffle the coordinates of $\boldsymbol{w}$ and those of $\boldsymbol{x}$ according to the same random permutation, then*

1. *For $L = \Omega(\frac{1}{\gamma^2} \log \frac{1}{\boldsymbol{m}})$, with probability $1 - \boldsymbol{m}$, there is an interval of length L that validates y;*
2. *For $L = \Omega(\frac{1}{\gamma^2} \log \frac{d}{\boldsymbol{m}})$, with probability $1 - \boldsymbol{m}$, no interval of length L can validate $-y$.*

Proof. Let S be any fixed interval of length L, then the distribution of $\sum_{i \in S} w_i x_i$ is identical to the distribution of the sum of L i.i.d. random variables drawn from $\{w_1 x_1, \ldots, w_d x_d\}$ without replacement. Note that $dyw_1 x_1, \ldots, dyw_d x_d$ are d numbers with mean γ, and their absolute values are bounded by $10^2 = O(1)$. By Chernoff bound,

$$\Pr\left[\frac{1}{L}\sum_{i \in S} dyw_i x_i - \gamma < -\epsilon\right] \leq e^{-O(\epsilon^2 |S|)}.$$

Set $\epsilon = \gamma$ ensures that $y \sum_{i \in S} w_i x_i > 0$ with probability $e^{-O(\gamma^2 |S|)}$. We can fix any interval S to prove Item 1. Taking union bounds over all intervals of length L, the probability of existing an interval of length L that validates $-y$ should be no greater than $\sum_{|S|=L} e^{-O(\epsilon^2 L)} \leq d^2 e^{-O(\gamma^2 L)}$, which proves Item 2. □

This shows that such masking explanations make sense to humans: if the model predicts y, then we can find an interval of length $\tilde{\Omega}(1/\gamma^2)$ so that computing the inner product only in that interval leads to the same prediction; otherwise if the model does not predict y, such interval cannot be found. Thus it is sufficient to convince humans that the model predicts y by only revealing the existence of such interval and the coordinate values in it.

Although the example given in this section is simple, the conceptual message is clear: saliency methods may not guarantee soundness in itself, but adding regularity constraints such as TV can mitigate this soundness issue effectively.