VARIATIONAL PERTURBATIONS FOR VISUAL FEATURE ATTRIBUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Explaining a complex black-box system in a post-hoc manner is important to understand its predictions. In this work we focus on two objectives, namely on how well the estimated explanation describes the classifier's behavior (faithfulness), and how sensitive the explanation is to input variations or model configurations (robustness). To achieve both faithfulness and robustness, we propose an uncertainty-aware explanation model, Variational Perturbations (VP), that learns a distribution of feature attribution for each image input and the corresponding classifier outputs. This differs from existing methods, which learn one deterministic estimate of feature attribution. We validate that according to several robustness and faithfulness metrics our VP method provides more reliable explanations compared to state-of-the-art methods on MNIST, CUB, and ImageNet datasets while also being more efficient.

1 INTRODUCTION

Explainable AI is becoming an essential field to understand the underlying process of a black-box system. This is especially required in safety-critical applications where transparent communication between machine learning system and human users is crucial (Lipton, 2018; Doshi-Velez & Kim, 2017). We particularly focus on explaining an image classification model in a post-hoc manner via feature attribution (Simonyan et al., 2013), i.e. contribution of each feature of an input image towards the classifier's prediction.

We propose a perturbation-based explanation method called Variational Perturbations (VP) to learn a posterior distribution of feature attribution for each image input and corresponding classifier outputs. We design the likelihood function to follow the property of explanation that features are likely to have higher attribution when they are considered sufficient to provide similar classifier outputs compared with that of original image input. As our prior being the standard Gaussian distribution makes the posterior intractable, we use a variational Bayesian method (Hoffman et al., 2013; Kingma & Welling, 2013) to approximate it. The approximate posterior is represented by a neural network which is optimized with the dataset used for training the classifier.

When it comes to reliably estimating feature attribution-based explanations, faithfulness measures the quality of an estimated feature attribution describing the classifier's behavior and robustness seeks to have roughly similar feature attribution when explaining subtle variations of different inputs, i.e. stability, or the same input but with different experimental configurations, i.e. consistency (Ribeiro et al., 2016; Ghorbani et al., 2019). Considering both faithfulness and robustness have been discussed previously in the literature for self-explainable models (Alvarez-Melis & Jaakkola, 2018a; Lee et al., 2019) and gradient-based methods (Dombrowski et al., 2019), but not for perturbation-based methods.

The faithfulness of our VP model comes from the design of the likelihood function. Furthermore, when optimizing the approximate posterior, globally exploring the search space of feature attribution by sampling from the approximate posterior makes the explanation more robust to variations of input or experimental configurations. To evaluate these two traits, we compare VP with state-of-the-art methods with respect to several consistency, stability, and perturbation metrics on MNIST, CUB and the large-scale ImageNet datasets. We observe that our method has the best performance in the overall ranking of the methods while being efficient, over $\times 1000$ faster compared to some of

previous methods, e.g., MP (Fong & Vedaldi, 2017), RISE (Petsiuk et al., 2018), EP (Fong et al., 2019), and AP (Elliott et al., 2021).

In summary, our contributions are: 1) We propose a perturbation-based explanation method, VP, that estimates a distribution of feature attribution, 2) We conduct our experiments on three datasets of varying difficulty and compare our method to seven state-of-the-art visual explanation methods, and 3) We show that our method is robust to input perturbations and experimental settings and faithful to explanation of the classifier decision while being efficient in inferring the feature attribution.

2 Related Works

Faithfulness on explanation. Estimating the importance of features of an input image in a post-hoc manner has been studied in many different ways. Perturbation-based explanation is one of the methodology where a feature attribution is estimated by locally perturbing an image or features of an image to infer the importance. The simplest way of perturbing a region of image is to replace the region with zero values (Ribeiro et al., 2016; Lundberg & Lee, 2017; Petsiuk et al., 2018) or with a blurred image (Fong & Vedaldi, 2017; Fong et al., 2019). However, images perturbed by these methods could lead to out-of-distribution samples, leading to unintended artifacts for explanation. To address this issue, Zintgraf et al. (2017) and Chang et al. (2019) replace the region by inpainting. Previously mentioned methods need a considerable time to infer a single explanation since they have to perform optimization. To address this issue, Dabkowski & Gal (2017) and Chen et al. (2018) propose a learnable explainer where the output is a feature attribution, and it is obtained by a single forward pass in the inference phase. Our method belongs to a perturbation-based explanation method. We make a comparison with some of the key methods in this domain.

Backpropagation-based explanation methods aim to explain a classifier's output signal by tracing it back through the classifier to endow relavance score for each input features. The first deep neural network approach in this family, i.e. Simonyan et al. (2013) approximates a classifier of a given input as a linear function by Taylor expansion, and considers the gradient of the input as feature attribution. However since the gradient only considers local sensitivity, it does not satisfy several axioms of explanation. Follow-up works have proposed methods satisfying axioms (Sundararajan et al., 2017; Srinivas & Fleuret, 2019) or reducing noise in saliency by ensembling (Smilkov et al., 2017; Adebayo et al., 2018; Hooker et al., 2019; Kapishnikov et al., 2021). Also along with gradient-based methods, handcrafted propagation rules have been proposed (Zeiler & Fergus, 2014; Springenberg et al., 2015; Bach et al., 2015; Shrikumar et al., 2017; Kindermans et al., 2018).

Robustness on explanation. The notion of robustness on explanation was first introduced by Alvarez-Melis & Jaakkola (2018b). There have been approaches to improve the robustness on stability by building self-explainable models (Alvarez-Melis & Jaakkola, 2018a; Lee et al., 2019) or gradient-based methods (Dombrowski et al., 2019). Zhao et al. (2021) introduce a Bayesian framework that incorporates prior knowledge to build a local surrogate model which ensures robustness to kernel settings and stability of explanation. Finally, Slack et al. (2020) recently proposed BayesLIME, a Bayesian local explanation which can captures two kinds of uncertainty, which are feature importance uncertainty and error uncertainty. Based on the uncertainty, it ensures reliability and provides the way to address inconsistency on randomness derived from sampling nearest inputs for training a local explainer. While BayesLIME captures uncertainty of each feature attribution caused by finite number of sampling from the nearest inputs, our method captures that uncertainty caused by ambiguity of relative importance between different features.

3 Variational Perturbations

In this section, we introduce our Variational Perturbations (VP) method for generating perturbation-based explanations. Let us define a pretrained multi-class classifier that we aim to interpret as $f : \mathbb{R}^{C\times H\times W} \rightarrow \mathbf{\Delta}^{K-1}$ where $\boldsymbol{x} \in \mathbb{R}^{C\times H\times W}$ is an input with C, H, and W to be channel, height, and width of the input image, and $\hat{\boldsymbol{y}} \in \mathbf{\Delta}^{K-1}$ is a K-dimensional predictive probability output with sum of elements equal to 1. To explain the behaviour of the classifier's output signal by feature attribution map, we introduce a

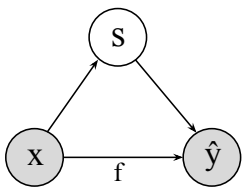


Figure 1: Graphical model.

random variable $\boldsymbol{s} \in \mathbb{R}^{H \times W}$ that describes the importance of each feature in a given image. By observing the feature attribution $\boldsymbol{s}$ with input image $\boldsymbol{x}$, we understand why the function f gives output signal $\hat{\boldsymbol{y}}$ (Figure 1). Our goal is to calculate the posterior distribution of the feature attribution, $p(\boldsymbol{s}|\boldsymbol{x}, \hat{\boldsymbol{y}})$. By Bayes' rule, the posterior is stated as:

$$p(\boldsymbol{s}|\boldsymbol{x}, \hat{\boldsymbol{y}}) \;=\; p(\hat{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{s})\; p(\boldsymbol{s}|\boldsymbol{x}) \;/\; \mathrm{Z}\,, \tag{1}$$

where Z is the marginal likelihood. We should model two terms, the likelihood $p(\hat{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{s})$ and the prior $p(\boldsymbol{s}|\boldsymbol{x})$, in order to calculate the posterior.

3.1 Modeling Likelihood and Prior

For designing the likelihood, we focus on the "response to occluding" property in explanation: the importance of each feature in an image is determined by observing the response of the classifier's output when the feature is occluded (Zeiler & Fergus, 2014). If output signal changes considerably when some part of an image is deleted, that erased region is regarded as cue for the classifier's output. Based on this concept, Fong & Vedaldi (2017); Dabkowski & Gal (2017) suggest "smallest sufficient region (SSR)" where they aim to find the smallest but sufficient cue region for explaining the classifier output. The difference between SSR and our approach is that we do not consider the *smallest* sufficient region, but rather *rank* the features to obtain relative importance among different features. Moreover, we consider not the *target class* but the *predictive probability output* $\hat{y}$ to interpret the model itself. The likelihood is designed such that the aforementioned properties satisfy:

$$-\log p(\hat{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{s}, k) \;=\; D_{\mathrm{KL}}[\, f(\boldsymbol{x}) \parallel f(\boldsymbol{x} \odot \tau^{(k)}(\boldsymbol{s}))\,] + \mathrm{const}\,, \tag{2}$$

$$p(\hat{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{s}) \;=\; \mathbb{E}_{p(k)}[\, p(\hat{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{s}, k)\,]\,, \tag{3}$$

where D_{KL} is a Kullback-Leibler (KL) divergence, $\tau^{(k)}(\cdot)$ is a top-k operator, and $\odot$ is a perturb operation that makes local perturbation of input $\boldsymbol{x}$. The top-k operator applied to the feature attribution map, $\tau^{(k)}(\boldsymbol{s}) \in \{0, 1\}^{H \times W}$, acts as a mask where $[\tau^{(k)}(\boldsymbol{s})]_{h,w} = 1$ when $\boldsymbol{s}_{h,w}$ corresponds to one of the biggest $k\%$ attributions in $\boldsymbol{s}$. This way, the conditional likelihood in equation 2 considers only the selected features in the input. We make local perturbation of input $\boldsymbol{x}$ by replacing the unselected region in an image with the baseline input $\tilde{\boldsymbol{x}}$, $\boldsymbol{x} \odot \boldsymbol{m} = \boldsymbol{x} \circ \boldsymbol{m} + \tilde{\boldsymbol{x}} \circ (1 - \boldsymbol{m})$, where $\circ$ is a pointwise multiplication. We bring three settings for the baseline input $\tilde{\boldsymbol{x}}$ in our experiment: blurred baseline[1], noise baseline[2], and mean baseline[3].

Equation 2 states that the predictive probability $\hat{\boldsymbol{y}}$ is more likely when the distance between the classifier's predictive probability of input $\boldsymbol{x}$ and that of perturbed input is close for a given k. We do not consider the ground-truth class or the top-1 predicted class, but rather all of the classes with predictive probability to examine the classifier's behavior itself. We set $p(k)$ as uniform distribution $U(0, 100)$ to consider various values of k, and take the expectation over $p(k)$ to get the likelihood $p(\hat{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{s})$ in Equation 3.

The easiest way of modeling the prior distribution $p(\boldsymbol{s}|\boldsymbol{x})$ is to first assume that $\boldsymbol{s}$ and $\boldsymbol{x}$ are independent, $p(\boldsymbol{s}|\boldsymbol{x}) = p(\boldsymbol{s})$, and then design $p(\boldsymbol{s})$ as a standard Gaussian distribution, $p(\mathrm{vec}(\boldsymbol{s})) = \mathcal{N}(\mathrm{vec}(\boldsymbol{s}); \boldsymbol{0}, \boldsymbol{I})$, where $\mathrm{vec}(\boldsymbol{s}) \in \mathbb{R}^{HW}$ is a vectorized version of $\boldsymbol{s}$. Other ways of modeling the prior (Carvalho et al., 2010; Blundell et al., 2015; Louizos & Welling, 2016) is out of our scope.

3.2 Variational Inference on Feature Attribution

As modeling the likelihood as Equation 3 and the prior as standard Gaussian distribution makes the posterior intractable in Equation 1, we approximate it with the distribution $q_\theta(\boldsymbol{s}|\boldsymbol{x})$ parameterized

[1]We define "blurred baseline" as an input image blurred with Gaussian kernel.

[2]The "noise baseline" is defined as Gaussian noise.

[3]We term "mean baseline" when the baseline is set to be the per channel mean of an original image and added by Gaussian noise.

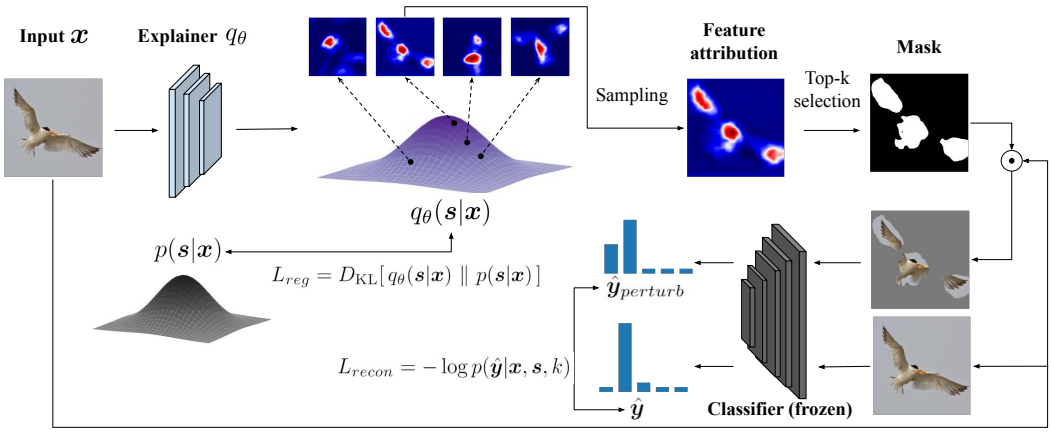


Figure 2: Schematic description. Our Variational Perturbations (VP) method is based on training an explainer q_θ of which the output is a distribution of feature attribution. The reconstruction loss L_{recon} forces the explainer to provide a faithful feature attirbution while the regularization loss L_{reg} regularizes the explainer to follow a prior distribution. Since our goal is to explain a classifier's prediction, we freeze the classifier in training.

by θ where the objective is to minimize the KL divergence between $q_\theta(\boldsymbol{s}|\boldsymbol{x})$ and $p(\boldsymbol{s}|\boldsymbol{x},\hat{\boldsymbol{y}})$:

$$
\begin{aligned}
&\underset{\theta}{\text{minimize}}\ D_{\mathrm{KL}}[\, q_\theta(\boldsymbol{s}|\boldsymbol{x}) \parallel p(\boldsymbol{s}|\boldsymbol{x}, \hat{\boldsymbol{y}}) \,] \\
=&\underset{\theta}{\text{minimize}}\ \mathbb{E}_{q_\theta(\boldsymbol{s}|\boldsymbol{x})}[\, -\log p(\hat{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{s}) \,] + D_{\mathrm{KL}}[\, q_\theta(\boldsymbol{s}|\boldsymbol{x}) \parallel p(\boldsymbol{s}|\boldsymbol{x}) \,] \qquad (4) \\
\leq&\underset{\theta}{\text{minimize}}\ \underbrace{\mathbb{E}_{q_\theta(\boldsymbol{s}|\boldsymbol{x})}\mathbb{E}_{p(k)}[\, -\log p(\hat{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{s}, k) \,]}_{(*)} + \underbrace{D_{\mathrm{KL}}[\, q_\theta(\boldsymbol{s}|\boldsymbol{x}) \parallel p(\boldsymbol{s}|\boldsymbol{x}) \,]}_{(**)} . \qquad (5)
\end{aligned}
$$

The inequality in Equation 5 stems from the Jensen's inequality. We apply a mean-field approximation with univariate Gaussian for each factorized term of approximate posterior, $q_\theta(\mathrm{vec}(\boldsymbol{s})|\boldsymbol{x}) = \mathcal{N}(\,\mathrm{vec}(\boldsymbol{s})\,; \boldsymbol{\mu}_\theta(\boldsymbol{x}), \mathrm{diag}(\boldsymbol{\nu}_\theta(\boldsymbol{x}))\,)$, where $\boldsymbol{\mu}_\theta(\cdot) \in \mathbb{R}^{HW}$ is a mean of the distribution and $\mathrm{diag}(\boldsymbol{\nu}_\theta(\cdot)) \in \mathbb{R}^{HW \times HW}$ is a diagonal covariance matrix with the main diagonal to be $\boldsymbol{\nu}_\theta(\cdot) \in \mathbb{R}^{HW}$. The parameter θ is optimized with the training dataset used for training the classifier f, and the reparameterization trick (Kingma & Welling, 2013) is applied during optimization. The schematic description is shown in Figure 2.

3.3 Optimization

The non-differentiable top-k operator in Equation 2 prevents the gradient flowing back through the parameter θ when using gradient descent for optimizing the Equation 5. Chen et al. (2018) address this issue by using k times of Gumbel softmax (Jang et al., 2017; Maddison et al., 2017). However, this does not select the exact $k\%$ pixels, but a number less than $k\%$. This makes a distribution shift in the explanation manifold between the training phase and the inference phase (more in Appendix A). Instead, we use a SOFT operator proposed by Xie et al. (2020). This is a differentiable operator that approximates the top-k operator with almost exact selection of $k\%$ pixels.

There are two terms in the objective function in Equation 5, i.e. the reconstruction loss (*) and the regularization loss (**). As we dissect the the regularization loss, it is factorized into HW number of KL divergence between two univariate Gaussian. This causes the regularization loss to be dominant in the objective function. To keep a balance between two terms, we introduce a regularization coefficient β ($\approx 1/HW$) multiplied to the regularization loss (Higgins et al., 2017).

In practice, we generate a feature attribution map starting from a smaller resolution followed by upsampling. More specifically, a spatial dimension of $\mu_\theta(\cdot)$ and $\nu_\theta(\cdot)$ is $H' \times W'$ where $H' = H/m$ and $W' = W/m$ with an upsample size m. It is then upsampled using a bilinear interpolation to have a spatial size of $H \times W$.

4 EXPERIMENTS

In this section we present experiments on our method and baseline methods. The quantitative comparisons on robustness (consistency and stability) and faithfulness are shown in §4.2. Then we compare qualitative results and show that our method passes a sanity check in §4.3. In §4.4 we compare the time complexity of the explanation methods. Finally, we summarize the overall results and provide a conclusion in §4.5.

4.1 DATSETS, METRICS, AND IMPLEMENTATION DETAILS

Datasets and classifiers. We use the standard benchmark MNIST (LeCun et al., 1998), CUB (Welinder et al., 2010), and ImageNet (Russakovsky et al., 2015) datasets with varying levels of difficulty for our experiments. For MNIST classifier, we use three convolutional layers followed by two linear layers that have 99.52% accuracy. For CUB and ImageNet classifiers we use ResNet50 (He et al., 2016) model that has 77.8% and 76.1% accuracies, respectively.

Training details. We use convolutional neural networks with five layers for MNIST dataset, and 12 layers for CUB and ImageNet datasets to represent $\mu_\theta(\cdot)$ and $\nu_\theta(\cdot)$. We use the downsample size $m = 2$ for MNIST and $m = 8$ for CUB and ImageNet datasets as a default setting. The regularization coefficient β is set to be $1/(1000HW)$ for MNIST and $1/(100HW)$ for CUB and ImageNet datasets where H and W are spatial size of input image. If not mentioned, the mean of approximate posterior $\mu_\theta(\cdot)$ is used while performing qualitative and quantitative experiments.

Compared methods. We compare our method with InputGrad (Simonyan et al., 2013), MP (Fong & Vedaldi, 2017), RealTime (Dabkowski & Gal, 2017), L2X (Chen et al., 2018), RISE (Petsiuk et al., 2018), EP (Fong et al., 2019), and AP (Elliott et al., 2021). Since our method belongs to perturbation-based method, we focus on comparing our method with previous perturbation-based methods which are MP, RISE, EP, and AP. We also compare our method with RealTime and L2X that infer a feature attribution in a real time. InputGrad represents the gradient-based methods. For Realtime, AP and EP we use the authors implementations, for RISE we use the TorchRay[4] library and we reimplemented other methods. We carefully optimized all the methods on these datesets to obtain the best results.

VP and nVP. To examine whether the uncertainty-aware explanation method is necessary for providing a faithful and robust explanation, we additionally propose a non-variational method where other settings are same with VP except for the standard deviation part. While training the explainer of this method, we ignore the regularization loss and the explainer provides only the mean values $\boldsymbol{\mu}_\theta(\cdot)$ and not the standard deviation values $\boldsymbol{\nu}_\theta(\boldsymbol{x})$. As a result, in the training process the feature attribution is not sampled from the output of the explainer but the mean $\boldsymbol{\mu}_\theta(\cdot)$ itself is used as an input of the top-k operator. We name this method as nVP. By comparing nVP with VP, we examine the effectiveness of globally exploring the search space of feature attribution by sampling from the distribution in the training process.

Metrics. We evaluate the robustness and faithfulness of each explanation method using three metrics. We take a look at the robustness of explanation to different hyper-parameter setting (consistency), and then observe the robustness of explanation to subtle change of input (stability) (Alvarez-Melis & Jaakkola, 2018a). For faithfulness, we measure the perturbation metric (Samek et al., 2016; Petsiuk et al., 2018). We explain the details of each metrics in each subsection.

4.2 QUANTITATIVELY EVALUATING ROBUSTNESS AND FAITHFULNESS

While most of the compared methods, i.e. InputGrad, RealTime, MP, RISE, EP, and AP, provide explanation on the classifier's output of specific target class, L2X and our method offer explanation on classifier's output of all classes. For a fair comparison with our method, we take quantitative evaluations on samples where classifier's prediction is correct with probability over than 0.5. Furthermore, we use randomly selected 200 samples as the default subset of the dataset for quantitative evaluation and report the mean and the standard deviation due to the variations in the time complexity of the evaluated methods (we provide a quantitative analysis of this in Sec.4.4).

[4]https://github.com/facebookresearch/TorchRay

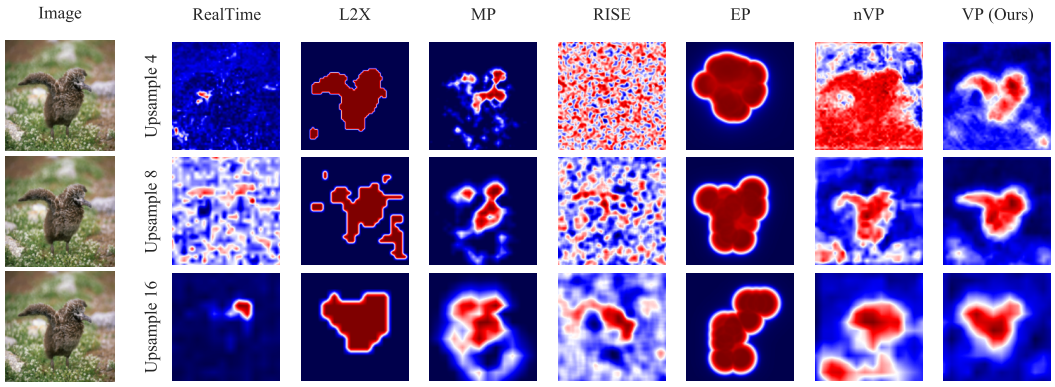


Figure 3: Consistency over different resolution. We evaluate feature attribution results with different upsampling size. Red (blue) colors represent higher (lower) attribution. We observe that our method highlights similar regions of the bird across different size of upsampling. However, for instance, nVP with upsample size 8 captures a bird object while upsample size 4 highlights background as well as the object. This indicates that the variational training leads to more robust explanations.

	SSIM			PC			PSNR		
Dataset	MNIST	CUB	ImageNet	MNIST	CUB	ImageNet	MNIST	CUB	ImageNet
RealTime	N/A	0.56	0.47	N/A	0.38	0.27	N/A	14.59	0.39
L2X	0.40	0.62	0.32	0.59	0.58	0.37	9.49	9.56	8.95
MP	0.52	0.54	0.42	0.48	0.67	0.60	11.50	14.10	10.25
RISE	0.11	0.47	0.30	0.07	0.30	0.01	12.51	15.63	13.56
EP	**0.58**	**0.72**	0.58	0.81	**0.83**	0.57	13.15	13.95	10.09
nVP	0.49	0.60	0.72	0.81	0.47	0.47	13.07	13.44	11.34
VP (Ours)	0.56	0.70	**0.78**	**0.85**	0.74	**0.70**	**13.81**	**16.34**	**13.79**

Table 1: Consistency evaluation. We measure the similarity of feature attribution on different size of upsampling. For every measurement in SSIM and PC metrics, the standard deviation was smaller than 0.05, and in PSNR it was under 0.5. We can not report RealTime results on MNIST as this explainer fundamentally uses features extracted from ResNet50 pre-trained on ImageNet not applicable for MNIST dataset.

4.2.1 Evaluating Explanation Robustness via Consistency To Parameters

In consistency evaluation, we measure the robustness of the explanation to different hyper-parameter settings by measuring the difference of feature attribution maps by structural similarity (SSIM)Wang et al. (2004), Pearson's correlation (PC), and peak signal-to-noise ratio (PSNR) metrics as well as we conduct a qualitative study. For the perturbation-based explanation methods that use mask as feature attribution, first the mask is usually smaller than the input image. It is then interpolated by bilinear upsampling to get the final feature attribution. The explanation method should provide a consistent feature attribution for different size of upsampling. As InputGrad and AP does not use mask for estimating the feature attribution, they do not appear in our consistency evaluation.

Our qualitative evaluation in Figure 3 shows consistent results by highlighting important regions for the classification no matter the upsample resolution. However, for MP, RealTime, L2X, and RISE the attribution maps differ based on the upsample resolution. For instance, L2X with upsample size 4 captures the bird while with upsample size 8 it highlights a larger portion of the background. Similarly, nVP highlights the background for the feature attribution with upsample size 4 indicating that the VP is more consistent than its non-variational counterpart.

For our quantitative evaluation on CUB and Imagenet datasets, for each input sample and explanation method, we measure the SSIM, PC, PSNR scores between two feature attribution maps with upsample sizes of $\{4, 8\}$, $\{8, 16\}$, and $\{4, 16\}$, respectively, then average three measurements. For MNIST dataset, $\{1, 2\}$, $\{2, 4\}$, and $\{1, 4\}$ are used. We randomly draw 50 input samples and average the evaluation results. We repeat this 4 times and report the mean of the averages in Table 1. Since the standard deviation of the averages is lower than 0.05 in the SSIM and Pearson's correlation metrics and 0.5 in the PSNR metric for all measurements, we omit to report them in the table.

	Probability difference			Logit difference			KL divergence		
Dataset	MNIST	CUB	ImageNet	MNIST	CUB	ImageNet	MNIST	CUB	ImageNet
Input-grad.	0.44	0.40	0.06	7.61	4.07	0.86	0.24	1.59	0.39
RealTime	N/A	0.29	0.04	N/A	2.84	0.97	N/A	0.92	1.07
L2X	0.43	0.48	0.15	7.65	4.21	2.25	0.24	1.38	**2.05**
MP	0.37	**0.54**	**0.17**	5.61	4.78	**2.51**	0.30	1.64	1.85
RISE	0.22	0.52	0.01	4.52	4.96	0.43	0.33	1.66	0.50
EP	0.40	0.31	0.14	6.87	2.63	1.08	0.35	0.58	0.31
AP	0.51	0.50	0.08	8.25	5.07	1.75	0.31	**1.94**	1.52
nVP	**0.70**	0.43	0.16	12.91	4.21	2.16	**0.37**	1.38	1.57
VP (Ours)	**0.70**	**0.54**	**0.17**	**12.97**	**5.18**	2.30	**0.37**	1.71	1.68

Table 3: Perturbation evaluation. We measure the faithfulness of each explanation method by perturbing the input. The area between plots of top-k and low-k perturbations on $k = \{0, 10, 20, \cdots, 100\}$ is reported. KL divergence values on CUB are $\times 10^{-2}$ and on ImageNet are $\times 10^{-3}$.

We observe that EP and our method have the best performance on the consistency: out of nine evaluations our method has the best on six measurements and three for EP. In case of SSIM metric on MNIST and CUB datasets, there is a 0.02 difference favoring EP over our method, on the large-scale ImageNet dataset that difference is 0.2 where our method performs significantly better than EP. Also, VP surpasses nVP across all datasets and metrics, i.e. see the last two rows in Table 1, indicating that our variational explanation is robust to different resolutions.

4.2.2 Evaluating Explanation Robustness via Stability To Input Image

Stability measures the robustness of explanation to subtle change in an input image. To measure stability, Alvarez-Melis & Jaakkola (2018b) propose a local Lipschitz value, $\text{stability(x)} = \max_{\boldsymbol{x}_i \in \mathcal{B}_\epsilon(\boldsymbol{x})} \frac{\|g(\boldsymbol{x}) - g(\boldsymbol{x}_i)\|_2}{\|\boldsymbol{x} - \boldsymbol{x}_i\|_2}$, where $g(\cdot)$ is the explainer with output to be a feature attribution, and $\mathcal{B}_\epsilon(\boldsymbol{x})$ is the ball of radius ϵ centered at $\boldsymbol{x}$. Since some of explanation methods are non-differentiable, the authors propose Bayesian optimization to get the stability measurement. In our case, we use the Bayesian optimization proposed by Wang et al. (2016) to solve the problem on high dimension (e.g., for ImageNet dataset the search space is $3 \times 224 \times 224$ dimension). We randomly select 30 images and report the mean and the standard deviation in Table 2.

Dataset	MNIST	CUB	ImageNet
Input-grad.	1.53 ± 0.27	0.32 ± 0.13	0.37 ± 0.11
RealTime	N/A	0.84 ± 0.85	0.47 ± 0.38
L2X	0.99 ± 0.25	0.69 ± 0.21	0.69 ± 0.14
MP	3.73 ± 2.08	0.93 ± 0.52	1.42 ± 0.37
RISE	0.16 ± 0.19	$\mathbf{0.23 \pm 0.24}$	$\mathbf{0.03 \pm 0.03}$
EP	3.48 ± 0.88	1.73 ± 0.56	2.06 ± 0.57
AP	1.60 ± 0.26	0.52 ± 0.11	0.64 ± 0.07
nVP	0.13 ± 0.03	0.52 ± 0.11	0.33 ± 0.22
VP (Ours)	$\mathbf{0.06 \pm 0.01}$	0.33 ± 0.16	0.29 ± 0.17

Table 2: Stability evaluation.

We observe that in MNIST dataset our method has the best performance by 0.06. RISE has the best performance on CUB and ImageNet datasets, followed by our method with 0.33 and 0.29, respectively. RISE has a higher stability since it repeats sampling the mask from the uniform distribution (e.g., 5000) and perform weighted average of masks, making the effect of subtle change of input insignificant with the cost of efficiency. Compared to nVP, VP has better performance across all datasets. This also supports the necessity of variational perturbations.

4.2.3 Evaluating Explanation Faithfulness via Perturbing The Input

We consider two types of perturbation metrics: top-k perturbation and low-k perturbation (Petsiuk et al., 2018; Samek et al., 2016). For the top-k (low-k) perturbation metric, image pixels corresponding to the largest (smallest) $k\%$ attributions are substituted with gray color, then measure the output change of the classifier. The explanation method is thought to be better when the output change is bigger for top-k perturbation, and smaller for low-k perturbation. To observe the unified measurement for perturbation metric, we first plot top-k perturbation and low-k perturbation measurements from $k = 0$ to 100,

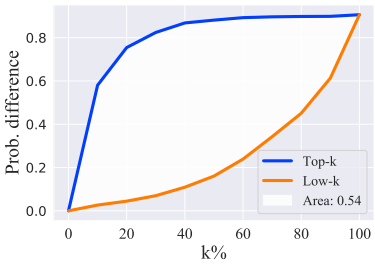


Figure 4: Perturbation evaluation.

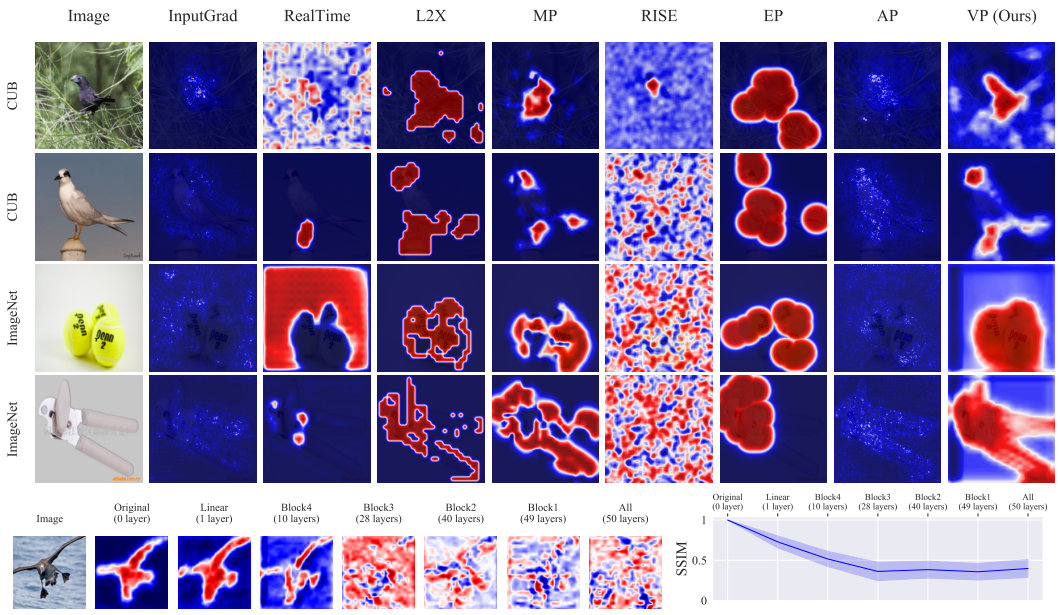


Figure 5: Qualitative results on CUB and ImageNet (top) and Sanity Check of VP on CUB (bottom). Red (blue) represents higher (lower) attribution. The first and the second rows are example attribution maps from CUB dataset, the third and the fourth rows are from ImageNet dataset. In the fourth row (left) we show visual inspection for sanity check and (right) each x-tick label indicates a layer to which the classifier's (ResNet50) weight is initialized from the top layer.

and then calculate the area between two plots as shown in Figure 4. The explanation method is considered better when the area is bigger. We look at two different types of output change, i.e. probability difference on ground-truth class and KL divergence on predictive probability vectors. The results are reported in Table 3.

For the metric of probability difference, our method has tied for the first place with nVP or MP by scoring 0.7, 0.54, and 0.17 for MNIST, CUB, and Imagenet datasets, respectively. For the metric of logit difference, our method ranked first on MNIST and CUB with the score 12.97 and 5.18 while MP had the best score with 2.51 on ImageNet dataset. For the KL divergence metric, our method, AP, and L2X had the best score of 0.37, 1.94, and 2.05 on MNIST, CUB, and ImageNet datasets, respectively. We also observe in the last two rows in Table 3 that compared with nVP, VP has equal or better performance across three datasets and three perturbation metrics.

4.3 Qualitative Evaluation of Feature Attribution Maps

Visualizing Feature Attribution Maps on CUB and ImageNet. To observe which region is considered important for each explanation method, we visualize samples on CUB and ImageNet datasets in Figure 5 (top). For L2X and EP, $k = 20\%$. We observe that while InputGrad has a noisy map, AP reduces the adversarial noise by constraining the difference of classifier's response of intermediate layers with perceptual loss. MP shows distracted attribution for some sample images since it does not globally explore the importance of features when optimizing the mask. EP highlights the region related with objects. For L2X method, we should manually choose the size k. However, it is non-trivial to decide whether k should be large or small to explain each samples. For the first example in the figure, the object (or possible cue location) is smaller than 20% of the size of input image, making the L2X explanation to capture on the background with $k = 20\%$ setting. Our method sometimes highlights a small and discrete region (e.g., bird head and leg in the second row in Figure 5), and sometimes a bigger region (e.g., all parts of tennis ball in third row).

Sanity Check. The prerequisite for the explanation method is to pass the sanity check (Adebayo et al., 2018). This is to identify whether the explanation method provides a feature attribution that is dependent of a classifier. It is tested by randomizing the classifier's parameters. The difference

between the feature attribution obtained from the original classifier and the parameter-initialized classifier is measured by structural similarity index (SSIM). The explanation method is regarded as passing the sanity check if the SSIM decreases as the number of initialized layers of classifier increases as shown in Figure 5 (bottom). This is because the explanation of randomized classifier should differ from that of original classifier. Since the similarity measurement converges to a small value as the number of initialized layers increases (0.3 for SSIM), this indicates that our method passes the sanity check. The sanity check on Pearson's correlation metric and more examples are shown in Appendix B.

4.4 Time Complexity Analysis

We measure the time taken to infer a feature attribution of a single image for each explanation method. We observe in Figure 6 that some of the perturbation-based methods, e.g. MP, RISE, EP, and AP, take a considerable amount of time, up to $\times 1000$ slower for inferring an explanation of a single sample compared to our method. This is because, for MP, EP, and AP, there is an optimization process to generate the explanation, and for RISE a process of inferring multiple samples takes a considerable time. For InputGrad, it takes more time than our method since it requires back-propagate operation. Instead, RealTime, L2X, and our method infer the explanation in real time since they obtain the result by single forward propagation. Therefore, these methods are advantageous in situation where multiple inputs are required for explanation.

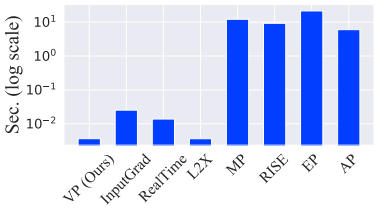


Figure 6: Time complexity. Time to infer a feature attribution in Quadro RTX 6000.

4.5 Summary of Results, Discussion and Conclusion

In this section, we summarize all our results of robustness and faithfulness to compare explanation methods at a glance. First, we rank all explanation methods with respect to all datasets by the evaluation metric. We then count the number of times each explanation method is ranked to a specific rank. For instance, VP takes the first place six times out of nine in consistency evaluation as observed in Table 1. This counting is recorded for all explanation methods with all ranking in a matrix form. We then compute the mean rank of each explanation method where this mean is used for sorting the rank matrix. The results are shown in Figure 7.

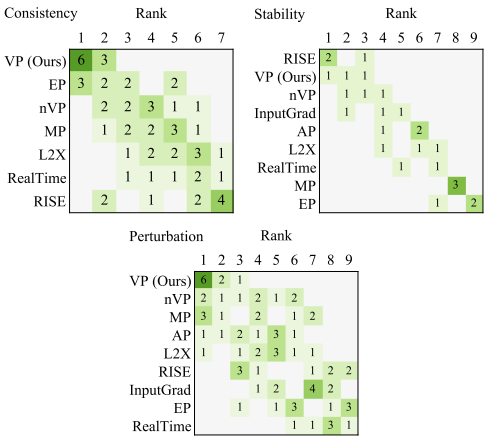


Figure 7: Rank matrix over all metrics. The score in each cell of the rank matrix indicates the number of times the explanation method had a specific ranking. From the left matrix to the right one, it shows the results of the consistency, stability, and perturbation benchmark. Darker cells indicate higher numbers.

We observe that our method has the first ranking in the consistency and perturbation benchmarks, and takes the second place in the stability benchmark, meaning that VP has a steady performance in both robustness and faithfulness. On the other hand, baseline methods tend to be placed in a high rank only in one or two of the three benchmarks. For instance, while RISE takes the first ranking in the stability benchmark, it places fifth and sixth in the consistency and perturbation benchmarks, respectively. The final observation in Figure 7 is that VP outperforms nVP across all three benchmarks. There is two gap of ranking between VP and nVP in the consistency benchmark, and one gap for the stability and perturbation benchmarks.

In conclusion, we proposed a perturbation-based explanation model that estimates a distribution of feature attribution. This model is trained to optimize a loss function that forces the model to provide a faithful and robust feature attribution. In the inference phase, our method is timely efficient since it only takes a single forward-propagation for inferring the feature attribution. By comparing our method with baseline methods in both quantitative and qualitative manner, we showed that our method achieves both faithfulness and robustness of explanation.

5 ETHICS AND REPRODUCABILITY STATEMENT

Explaining a black-box system is essential especially when it is deployed in a real world, for example in medicine or in societal contexts, where users or developers require a trustworthy about the system's prediction. Explainability has been on a strong emphasis in the European Data Protection Regulation (GDPR) and will be increasingly more important in the future for AI legislation. The methodology we propose makes one step progress towards building a trustworthy system. However, the explanation methods should yet to be handled with care since the ground-truth explanation usually does not exist. Therefore, better metric for evaluating the explanation methods should be developed further to address the issue.

Regarding our experimental setup: The properties of the datasets depend on data collection practices and other design choices made during dataset creation. In this work, we use standard open-sourced datasets that do not require ethical approval.

For reproducability of our method, we provide the detailed hyperparameter setting in §4.1 and the pseudo-code in Appendix E. We will release the fully executable code as soon as our paper get accepted. Since our method does not require a lot of resources for training, e.g., an hour for CUB and 18 hours for ImageNet dataset with single Quadro RTX 6000 GPU, its low computational cost also makes it easy to reproduce the results presented in our paper.

REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 9525–9536, 2018.

David Alvarez-Melis and Tommi S Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7786–7795, 2018a.

David Alvarez-Melis and Tommi S Jaakkola. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*, 2018b.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pp. 1613–1622. PMLR, 2015.

Carlos M Carvalho, Nicholas G Polson, and James G Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=B1MXz20cYQ`.

Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pp. 883–892, 2018.

Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*, pp. 6967–6976, 2017.

Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. *Advances in Neural Information Processing Systems*, 32:13589–13600, 2019.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Andrew Elliott, Stephen Law, and Chris Russell. Explaining classifiers using adversarial perturbations on the perceptual ball. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10693–10702, 2021.

Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2950–2958, 2019.

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

Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3681–3688, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *Iclr*, 2(5):6, 2017.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *Advances in Neural Information Processing Systems*, 32: 9737–9748, 2019.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net, 2017.

Andrei Kapishnikov, Subhashini Venugopalan, Besim Avci, Ben Wedin, Michael Terry, and Tolga Bolukbasi. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5050–5058, 2021.

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

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Guang-He Lee, David Alvarez-Melis, and Tommi S. Jaakkola. Towards robust, locally linear deep networks. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=SylCrnCcFX`.

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

Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. In *International Conference on Machine Learning*, pp. 1708–1716. PMLR, 2016.

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*, pp. 4768–4777, 2017.

C Maddison, A Mnih, and Y Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the international conference on learning Representations*. International Conference on Learning Representations, 2017.

Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In *BMVC*, 2018.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

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.

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

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

Dylan Slack, Sophie Hilgard, Sameer Singh, and Himabindu Lakkaraju. Reliable post hoc explanations: Modeling uncertainty in explainability. *arXiv preprint arXiv:2008.05030*, 2020.

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

J Springenberg, Alexey Dosovitskiy, Thomas Brox, and M Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*, 2015.

Suraj Srinivas and François Fleuret. Full-gradient representation for neural network visualization. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 4124–4133, 2019.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328. PMLR, 2017.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando de Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

Yujia Xie, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister. Differentiable top-k operator with optimal transport. *arXiv preprint arXiv:2002.06504*, 2020.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Xingyu Zhao, Wei Huang, Xiaowei Huang, Valentin Robu, and David Flynn. Baylime: Bayesian local interpretable model-agnostic explanations. In *37th Conference on Uncertainty in Artificial Intelligence*, 2021.

Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*, 2017.

APPENDIX

A APPROXIMATING TOP-K OPERATOR: L2X VS VP

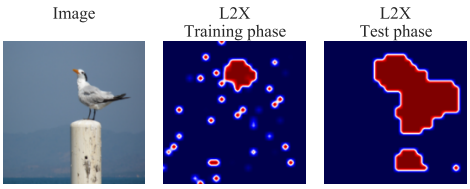
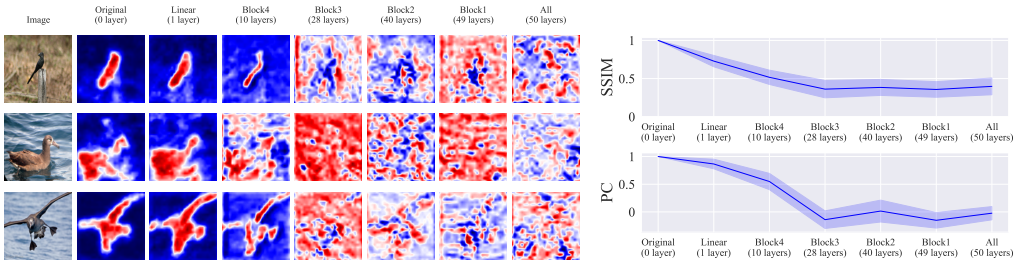


Figure 8: Explanation discrepancy between training phase and test phase for L2X when $k = 20\%$.

Top-k operator is non-differentiable since this operator selects k indices of pixels corresponding to top-k values. This hinders the gradient to flow backward through the explainer in the training phase of L2X or VP. To address this issue, L2X uses k times of Gumbel softmax (Jang et al., 2017; Maddison et al., 2017) for differentiable approximation of the top-k operator. More specifically, they first draw k times of sampling, say $M^1, M^2, \cdots, M^k$, from the Concrete distribution, where M^j is the Gumbel softmax with having HW size of dimension. They then get a final mask M by taking the maximum value along each indices of k samples, $M_i = \max_j M_i^j$. This process allows duplicate selection, leading to selecting less number of pixels than k. As seen in Figure 8, when $k = 20\%$, top-k approximator of L2X captures less size of region than 20% (middle figure), while in the test phase we exactly choose 20% pixels by top-k operator (right figure). This makes a distribution shift in generating a mask between the training phase and the inference phase. Instead, we use a SOFT operator (Xie et al., 2020) which is a differentiable approximator of top-k operator that almost exactly selects $k\%$ pixels.

B SANITY CHECK



(a) **(b)**

Figure 9: Sanity Check on CUB dataset.

We show more examples of sanity check in Figure 9 (a) and the plots of SSIM and PC metrics in Figure 9 (b). Since the similarity measurement converges to a small value as the number of initialized layers increases, i.e. 0.3 for SSIM and 0 for PC, this indicates that our method passes the sanity check.

C MORE QUALITATIVE EXAMPLES

Additional qualitative examples on MNIST, CUB, and ImageNet datasets are shown in Figure 10, 11, and 12, respectively.

D PERTURBATION EVALUATION

We present a Table 4 that is same with Table 3, but with standard deviation reported.

	probability difference			logit difference			KL divergence		
Dataset	MNIST	CUB	ImageNet	MNIST	CUB	ImageNet	MNIST	CUB	ImageNet
Input-grad.	0.44	0.40	0.06	7.61 ± 0.46	4.07 ± 0.19	0.86 ± 0.40	0.24 ± 0.03	1.59 ± 0.09	0.39 ± 0.35
RealTime *	N/A	0.29	0.04	N/A	2.84 ± 0.31	0.97 ± 0.35	N/A	0.92 ± 0.09	1.07 ± 0.22
L2X	0.43	0.48	0.15	7.65 ± 0.43	4.21 ± 0.26	2.25 ± 0.31	0.24 ± 0.01	1.38 ± 0.05	$\mathbf{2.05 \pm 0.24}$
MP	0.37	**0.54**	**0.17**	5.61 ± 0.87	4.78 ± 0.27	$\mathbf{2.51 \pm 0.16}$	0.30 ± 0.03	1.64 ± 0.08	1.85 ± 0.14
RISE *	0.22	0.52	0.01	4.52 ± 0.65	4.96 ± 0.46	0.43 ± 0.12	0.33 ± 0.04	1.66 ± 0.15	0.50 ± 0.12
EP *	0.40	0.31	0.14	6.87 ± 0.48	2.63 ± 0.18	1.08 ± 0.14	0.35 ± 0.03	0.58 ± 0.08	0.31 ± 0.08
AP *	0.51	0.50	0.08	8.25 ± 0.36	5.07 ± 0.31	1.75 ± 0.19	0.31 ± 0.02	$\mathbf{1.94 \pm 0.12}$	1.52 ± 0.15
nVP	**0.70**	0.43	0.16	12.91 ± 0.54	4.21 ± 0.60	2.16 ± 0.25	$\mathbf{0.37 \pm 0.02}$	1.38 ± 0.14	1.57 ± 0.15
VP (Ours)	**0.70**	**0.54**	**0.17**	$\mathbf{12.97 \pm 0.54}$	$\mathbf{5.18 \pm 0.47}$	2.30 ± 0.24	$\mathbf{0.37 \pm 0.02}$	1.71 ± 0.09	1.68 ± 0.16

Table 4: Perturbation evaluation. We measure the faithfulness of each explanation methods by perturbing an input. For every measurements on probability difference, the standard deviation was below 0.05. KL divergence values on CUB are $\times 10^{-2}$ and on ImageNet are $\times 10^{-3}$.

E PSEUDO-CODE

```
classifier = ResNet50()
explainer_vp = ExplainerVP()
soft_topk_approximator = SOFTTopkApproximator()

optimizer = torch.optim.Adam(explainer_vp.parameters())

# loss function
def loss_fn(outputs_masked, outputs_origin,
            mu, logvar, targets, beta):
    B = outputs_masked.size(0)
    recon_loss = kl_divergence(outputs_masked, outputs_origin)
    reg_loss = -0.5 * \
               torch.sum(1 + logvar - mu.pow(2) - logvar.exp()) / B
    return recon_loss + beta * reg_loss

# train
for inputs, targets in train_loader:
    B, C, H, W = inputs.size()

    # output of origin image
    with torch.no_grad():
        outputs_origin = classifier(inputs)

    # get mask from feature attribution
    topk_ratio = torch.FloatTensor(1).uniform_(0., 1.).item()
    attr_mu, attr_logvar, attr_sampled = explainer_vp(inputs)
    mask = soft_topk_approximator(attr_sampled, topk_ratio)

    # output of perturbed image
    inputs_baseline = get_baseline_images(inputs, baseline_type)
    inputs_masked = inputs * mask + inputs_baseline * (1 - mask)
    outputs_masked = classifier(inputs_masked)

    # update parameters of VP explainer
    loss = loss_fn(outputs_masked, outputs_origin,
                   attr_mu, attr_logvar, beta)
    loss.backward()
    optimizer.step()
```

Listing 1: Python pseudo-code for VP.

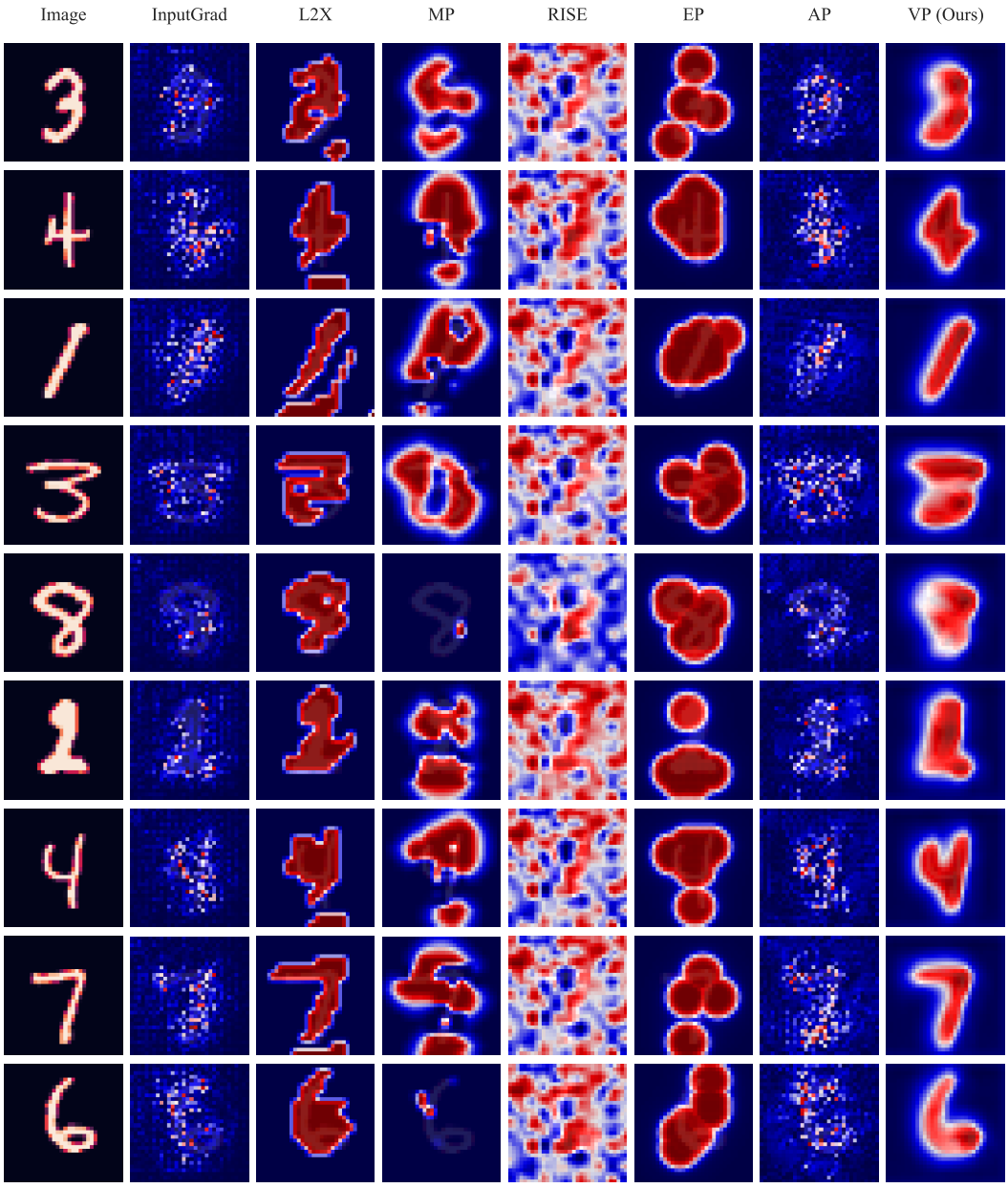


Figure 10: Examples of feature attribution on MNIST dataset.

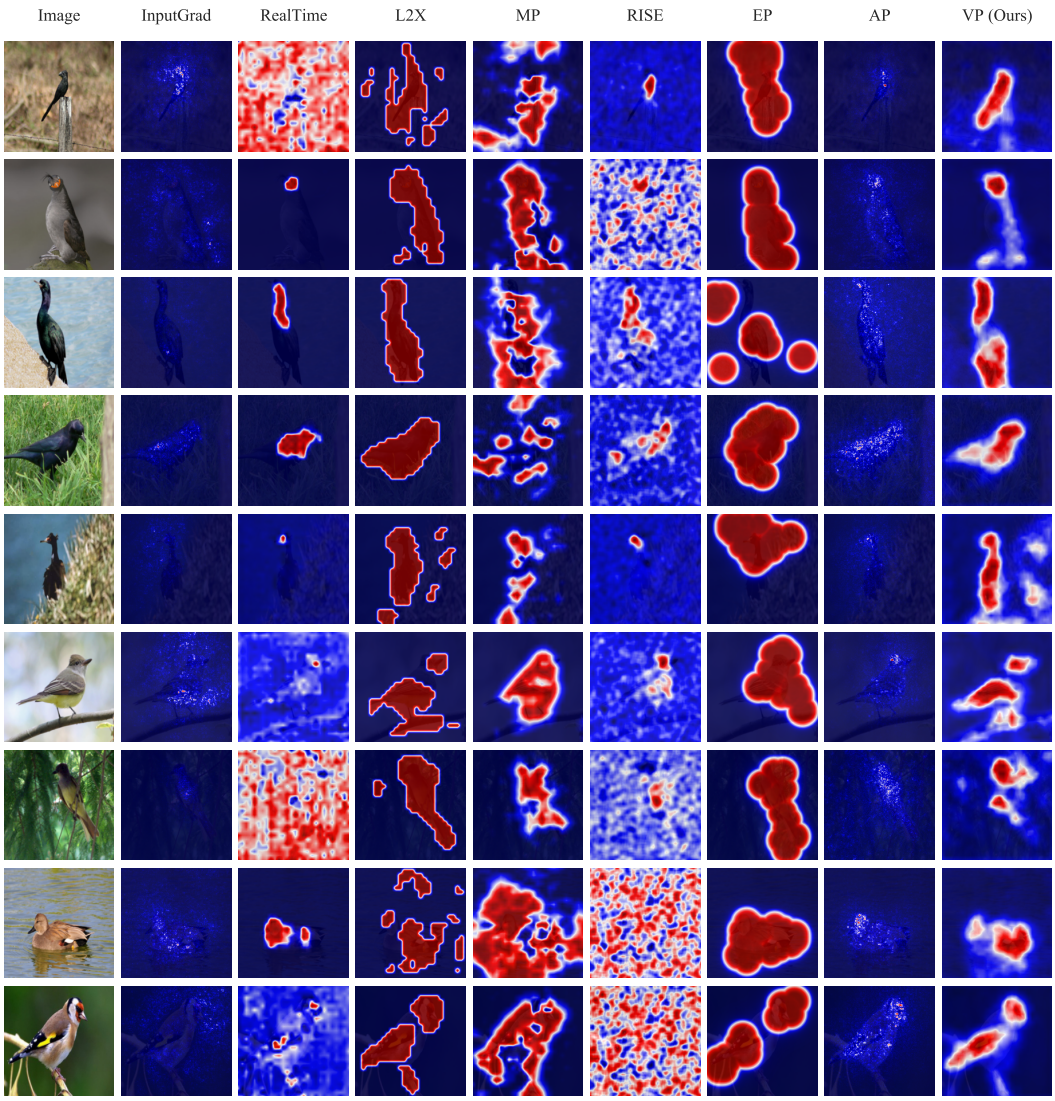


Figure 11: Examples of feature attribution on CUB dataset.

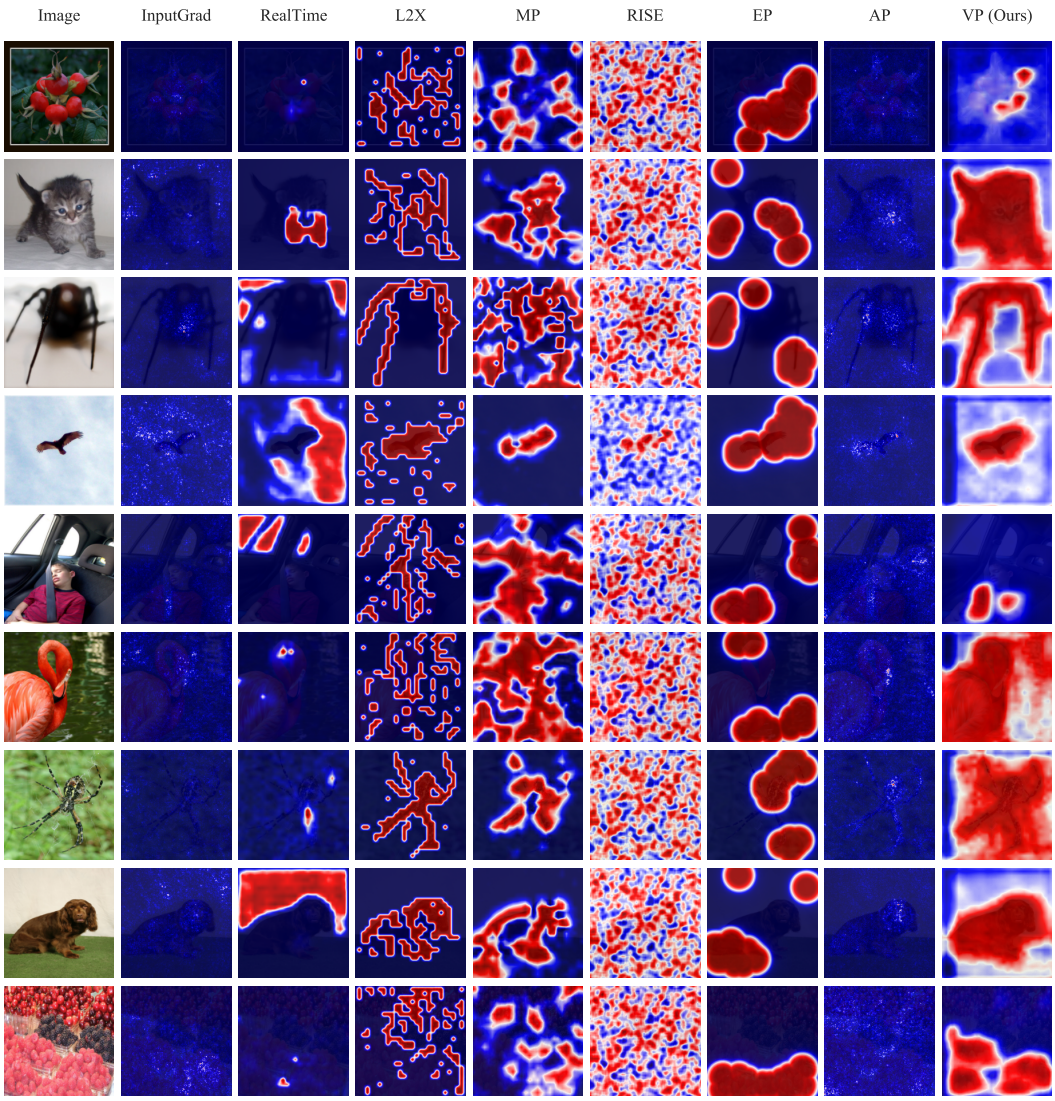


Figure 12: Examples of feature attribution on ImageNet dataset.