EXPLAINING IMAGE CLASSIFIERS BY COUNTERFACTUAL GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

When a black-box classifier processes an input to render a prediction, which input features are relevant and why? We propose to answer this question by efficiently marginalizing over the universe of plausible alternative values for a subset of features by conditioning a generative model of the input distribution on the remaining features. In contrast with recent approaches that compute alternative feature values ad-hoc—generating counterfactual inputs far from the natural data distribution—our model-agnostic method produces realistic explanations, generating plausible inputs that either preserve or alter the classification confidence. When applied to image classification, our method produces more compact and relevant per-feature saliency assignment, with fewer artifacts compared to previous methods.

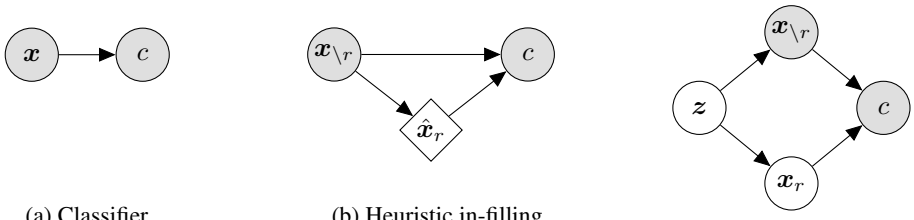
1 INTRODUCTION

Neural networks achieve state-of-the-art performance in many domains, but their decisions are difficult to interpret. Saliency maps are a tool for interpreting neural network classification that, given a particular input example and output class, score the relevance of each input dimension to the resulting classification. Fong & Vedaldi (2017) and Dabkowski & Gal (2017) cast saliency computation an optimization problem informally described by the following question: *which inputs, when replaced by an uninformative reference value, maximally change the classifier output?* Because these methods use heuristic reference values, e.g. blurred input (Fong & Vedaldi, 2017) or random colors (Dabkowski & Gal, 2017), they ignore the context of the surrounding pixels, often producing unnatural in-filled images (Figure 2). Therefore these approaches have to deal with a somewhat unusual question of how the classifier responds to images outside of its training distribution.

To encourage explanations that are consistent with the data distribution, we modify the question at hand: *which region, when replaced by plausible alternative values, would maximally change classifier output?* We marginalize out the masked region, conditioning the generative model on the non-masked parts of the image to sample counterfactual inputs that either change or preserve classifier behavior. In this paper we provide a new model-agnostic framework for visualizing feature importance from any differentiable classifier, based on variational Bernoulli dropout (Gal & Ghahramani, 2016). By leveraging a powerful in-filling conditional generative model we produce saliency maps on ImageNet that identify relevant and concentrated pixels better than existing methods.

2 RELATED WORK

Gradient-based approaches (Simonyan et al., 2013; Springenberg et al., 2014; Zhang et al., 2016; Selvaraju et al., 2016) derive a saliency map for a given input example and class target by computing the gradient of the classifier output with respect to each component (e.g., pixel) of the input. The reliance on the local gradient information induces a bias due to gradient saturation or discontinuity in the DNN activations (Shrikumar et al., 2017). *Reference-based approaches* analyze the sensitivity of classifier outputs to the substitution of certain inputs/pixels with an uninformative reference value. Shrikumar et al. (2017) linearly approximates this change in classifier output using an algorithm resembling backpropagation. This method is efficient and addresses gradient discontinuity, but ignores nonlinear interactions between inputs. Chen et al. (2018) optimizes a variational bound on the mutual information between a subset of inputs and the target, using a variational family that sets



(a) Classifier (b) Heuristic in-filling (c) Generative infilling

Figure 1: **Graphical models**. $p_{\mathcal{M}}(c|\boldsymbol{x})$ is classifier whose behavior we wish to analyze (1a) To explain its response to a particular input $\boldsymbol{x}$ we use variational Bernoulli dropout to induce a distribution over partitions of the input into masked (unobserved) features and their complement $\boldsymbol{x} = \boldsymbol{x}_r \cup \boldsymbol{x}_{\setminus r}$. Input features are *salient* if (i) they belong to a $\boldsymbol{x}_{\setminus r}$ whose inclusion yields classifier outcomes $p_{\mathcal{M}}(c|\boldsymbol{x}_{\setminus r})$ similar to $p_{\mathcal{M}}(c|\boldsymbol{x}_r, \boldsymbol{x}_{\setminus r})$ or (ii) they belong to an $\boldsymbol{x}_r$ whose omission yields $p_{\mathcal{M}}(c|\boldsymbol{x}_{\setminus r})$ different from $p_{\mathcal{M}}(c|\boldsymbol{x}_r, \boldsymbol{x}_{\setminus r})$. Fong & Vedaldi (2017); Dabkowski & Gal (2017) measure classifier response $p_{\mathcal{M}}(c|\hat{\boldsymbol{x}}_r, \boldsymbol{x}_{\setminus r})$ with $\hat{\boldsymbol{x}}_r$ computed ad-hoc (e.g., image blur) (1b). This biases the explanation when samples $[\hat{\boldsymbol{x}}_r, \boldsymbol{x}_{\setminus r}]$ deviate from the data distribution $p(\boldsymbol{x}_r, \boldsymbol{x}_{\setminus r})$. We instead marginalize $\boldsymbol{x}_r$ efficiently by sampling from a conditional generative model $\boldsymbol{x}_r \sim p_G(\boldsymbol{x}_r|\boldsymbol{x}_{\setminus r})$ (1c)

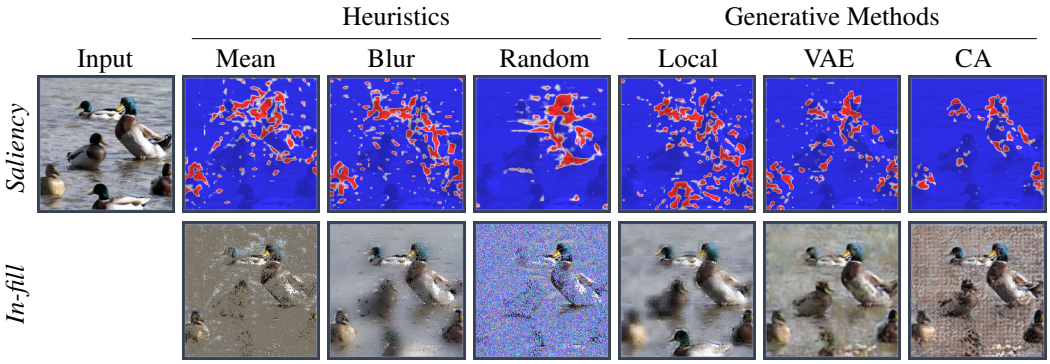


Figure 2: **Computed saliency for a variety of in-filling techniques**. Each saliency map (top row) results from maximizing in-class confidence by mixing a minimal region (red) of the original image with some reference image in the complementary (blue) region. The resulting mixture (bottom row) is fed to the classifier during the inner loop of optimization. We compare 6 methods for computing the reference, 3 heuristics and 3 generative models. We argue that strong generative models—e.g., Contextual Attention GAN (CA) (Yu et al., 2018)—ameliorate in-fill artifacts, making explanations more plausible under the data distribution.

input features outside the chosen subset to zero. In both cases, the choice of background value as reference limits applicability to simple image domains with static background like MNIST.

Zintgraf et al. (2017) computes the saliency of a pixel (or image patch) by treating it as unobserved and marginalizing it out, then measuring the change in classification outcome. This approach is similar in spirit to ours. The key difference is that where Zintgraf et al. (2017) iteratively execute this computation for each region, we leverage a variational Bernoulli distribution to efficiently search for optimal solution while encouraging sparsity. This reduces computational complexity and allows us to model the interaction between disjoint regions of input space. Fong & Vedaldi (2017) computes saliency by optimizing the change in classifier outputs with respect to a perturbed input, expressed as the pixel-wise convex combination of the original input with a reference image. They offer three heuristics for choosing the reference: mean input pixel value (typically gray), Gaussian noise, and blurred input. Dabkowski & Gal (2017) amortize the cost of estimating these perturbations by training an auxiliary neural network.

3 Proposed Method

Dabkowski & Gal (2017) propose two objectives for computing the saliency map:

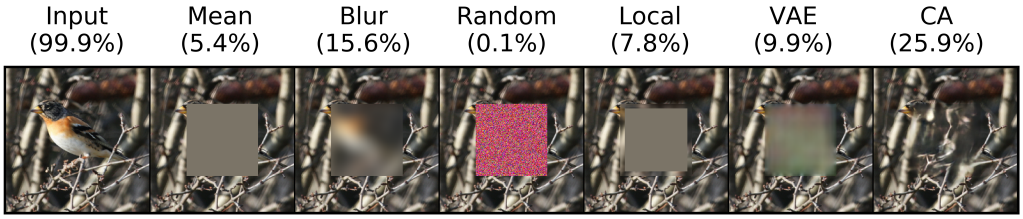


Figure 3: **Visualization of reference value infilling methods under centered mask**. The ResNet output probability of the correct class is shown for each imputed image.

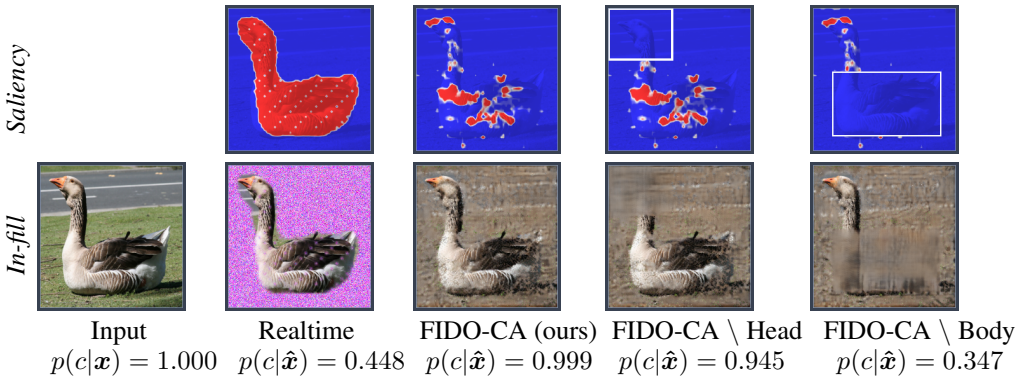


Figure 4: **Classifier confidence of infilled images.** Given an input, FIDO-CA finds a minimal pixel region that preserves the classifier score following in-fill by CA-GAN (Yu et al., 2018). The real-time method (Dabkowski & Gal, 2017) assigns saliency coarsely around the central object, and the heuristic infill reduces the classifier score. We mask further regions (head and body) of the FIDO-CA saliency map by hand, and observe a drop in the infilled classifier score.

- Smallest Deletion Region (SDR) considers a saliency map as an answer to the question: *What is the smallest input region that could be removed and swapped with alternative reference values in order to minimize the classification score?*
- Smallest Supporting Region (SSR) instead poses the question: *What is the smallest input region that could substituted into a fixed reference input in order to maximize the classification score?*

Solving these optimization problems (which we formalize below) involves a search over input masks, and necessitates *reference values* to be substituted inside (SDR) or outside (SSR) the masked region. These values were previously chosen heuristically, e.g., mean pixel value per channel. We instead consider inputs inside (SDR) or outside (SSR) the masked region as unobserved variables to be marginalized efficiently by sampling from a strong conditional generative model[1]. We describe our approach for an image application where the input comprises pixels, but our method is more broadly applicable to any domain where the classifier is differentiable.

Consider an input image $\boldsymbol{x}$, class c, and classifier with output distribution $p_{\mathcal{M}}(c|\boldsymbol{x})$. Denote by r a subset of the input pixels that implies a partition of the input $\boldsymbol{x} = \boldsymbol{x}_r \cup \boldsymbol{x}_{\setminus r}$. We refer to r as a region, although it may be disjoint. We are interested in the classifier output when $\boldsymbol{x}_r$ are unobserved, which can be expressed by marginalization as

$$p_{\mathcal{M}}(c|\boldsymbol{x}_{\setminus r}) = \mathbb{E}_{\boldsymbol{x}_r \sim p(\boldsymbol{x}_r|\boldsymbol{x}_{\setminus r})} \left[p_{\mathcal{M}}(c|\boldsymbol{x}_{\setminus r}, \boldsymbol{x}_r) \right] . \tag{1}$$

We then approximate $p(\boldsymbol{x}_r|\boldsymbol{x}_{\setminus r})$ by some generative model with distribution $p_G(\boldsymbol{x}_r|\boldsymbol{x}_{\setminus r})$ (specific implementations are discussed in section 4.1). Then given a binary mask[2] $\boldsymbol{z}$ and the original image $\boldsymbol{x}$,

[1] Whereas (Zintgraf et al., 2017) iteratively marginalize single patches conditioned on their surroundings, we model a more expressive conditional distribution considering the joint interaction of disparate regions.

[2] $\boldsymbol{z}_u = 0$ means the u-th pixel of $\boldsymbol{x}$ is dropped out. The remaining image is $\boldsymbol{x}_{\boldsymbol{z}=\boldsymbol{0}}$.

Algorithm 1: BBMP (Fong & Vedaldi, 2017)	**Algorithm 2:** FIDO (Ours)
Input image $\boldsymbol{x}$, classifier score $s_{\mathcal{M}}$, sparsity hyperparameter λ, objective $L \in \{L_{SSR}, L_{SDR}\}$	**Input** image $\boldsymbol{x}$, classifier score $s_{\mathcal{M}}$, sparsity hyperparameter λ, objective $L \in \{L_{SSR}, L_{SDR}\}$, generative model p_G
Initialize single mask $\boldsymbol{z} \in [0,1]^d$	Initialize dropout rate $\boldsymbol{\theta} \in [0,1]^d$
while loss L is not converged **do**	**while** loss L is not converged **do**
Clip $\boldsymbol{z}$ to $[0,1]$	Sample minibatch $\boldsymbol{z} \in \{0,1\}^d \sim \text{Bern}(\boldsymbol{\theta})$
Compute reference image $\hat{\boldsymbol{x}}$ heuristically	Sample reference image $\hat{\boldsymbol{x}} \sim p_G(\hat{\boldsymbol{x}}\|\boldsymbol{z}, \boldsymbol{x})$
Compute in-fill $\phi(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x} \cdot \boldsymbol{z} + \hat{\boldsymbol{x}} \cdot (1 - \boldsymbol{z})$	Compute in-fill $\phi(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x} \cdot \boldsymbol{z} + \hat{\boldsymbol{x}} \cdot (1 - \boldsymbol{z})$
With ϕ, compute L by Equation 4 or 5	With ϕ, compute L by Equation 4 or 5
Update $\boldsymbol{z}$ with $\nabla_z L$	Update $\boldsymbol{\theta}$ with $\nabla_\theta L$ (Maddison et al., 2016)
end while	**end while**
Return $\boldsymbol{z}$ as per-feature saliency map	Return $\boldsymbol{\theta}$ as per-feature saliency map

Figure 5: **Pseudo code comparison**. Differences between the approaches are shown in blue.

we define an infilling function ϕ as a convex mixture of the input and reference with binary weights,

$$\phi(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{z} \odot \boldsymbol{x} + (\boldsymbol{1} - \boldsymbol{z}) \odot \hat{\boldsymbol{x}} \;\text{ where } \hat{\boldsymbol{x}} \sim p_G(\hat{\boldsymbol{x}}|\boldsymbol{x}_{\boldsymbol{z}=\boldsymbol{0}}). \tag{2}$$

3.1 Objective functions

The classification score function $s_{\mathcal{M}}(c)$ represents a score of classifier confidence on class c; in our experiments we use log-odds:

$$s_{\mathcal{M}}(c|\boldsymbol{x}) = \log p_{\mathcal{M}}(c|\boldsymbol{x}) - \log(1 - p_{\mathcal{M}}(c|\boldsymbol{x})). \tag{3}$$

SDR seeks a mask $\boldsymbol{z}$ yielding low classification score when a small number of reference pixels are mixed into the mask regions. Without loss of generality[3] we can specify a parameterized distribution over masks $q_\theta(\boldsymbol{z})$ and optimize its parameters. The SDR problem is a minimization w.r.t θ of

$$L_{SDR}(\theta) = \mathbb{E}_{q_\theta(\boldsymbol{z})}\left[s_{\mathcal{M}}(c|\phi(\boldsymbol{x}, \boldsymbol{z})) + \lambda\|\boldsymbol{z}\|_1\right]. \tag{4}$$

On the other hand, SSR aims to find a masked region that maximizes classification score while penalizing the size of the mask. For sign consistency with the previous problem, we express this as a minimization w.r.t θ of

$$L_{SSR}(\theta) = \mathbb{E}_{q_\theta(\boldsymbol{z})}\left[-s_{\mathcal{M}}(c|\phi(\boldsymbol{x}, \boldsymbol{z})) + \lambda\|\boldsymbol{1} - \boldsymbol{z}\|_1\right]. \tag{5}$$

Naively searching over all possible $\boldsymbol{z}$ is exponentially costly in the number of pixels U. Therefore we specify $q_\theta(\boldsymbol{z})$ as a factorized Bernoulli:

$$q_\theta(\boldsymbol{z}) = \prod_{u=1}^{U} q_{\theta_u}(z_u) = \prod_{u=1}^{U} \text{Bern}(z_u|\theta_u). \tag{6}$$

This corresponds to applying Bernoulli dropout (Srivastava et al., 2014) to the input pixels and optimizing the per-pixel dropout rate. θ is our saliency map since it has the same dimensionality as the input and provides a probability of each pixel being marginalized (SDR) or retained (SSR) prior to classification. We call our method FIDO because it uses a strong generative model (see section 4.1) to Fill-In the DropOut region.

To optimize the θ through the discrete random mask $\boldsymbol{z}$, we follow Gal et al. (2017) in computing biased gradients via the Concrete distribution (Maddison et al., 2016; Jang et al., 2016); we use temperature 0.1. We initialize all our dropout rates θ to 0.5 since we find it increases the convergence speed and avoids trivial solutions. We optimize using Adam (Kingma & Ba, 2014) with learning rate 0.05 and linearly decay the learning rate for 300 batches in all our experiments. Our PyTorch implementation takes about one minute on a single GPU to finish one image.

[3] We can search for a single mask $\boldsymbol{z}'$ using a point mass distribution $q_{\boldsymbol{z}'}(\boldsymbol{z}) = \delta(\boldsymbol{z} = \boldsymbol{z}')$

3.2 COMPARISON TO FONG & VEDALDI (2017)

Fong & Vedaldi (2017) compute saliency by directly optimizing the continuous mask $\boldsymbol{z} \in [0, 1]$ under the SDR objective, with $\hat{\boldsymbol{x}}$ chosen heuristically; we call this approach Black Box Meaningful Perturbations (BBMP). We instead optimize the parameters of a Bernoulli dropout distribution $q_\theta(\boldsymbol{z})$, which enables us to sample reference values $\hat{\boldsymbol{x}}$ from a learned generative model. Our method uses mini-batches of samples $\boldsymbol{z} \sim q_\theta(\boldsymbol{z})$ to efficiently explore the huge space of binary masks and obtain uncertainty estimates, whereas BBMP is limited to a local search around the current point estimate of the mask $\boldsymbol{z}$. See Figure 5 for a pseudo code comparison. In Appendix A.1 we investigate how the choice of algorithm affects the resulting saliency maps.

To avoid unnatural artifacts in $\phi(\boldsymbol{x}, \boldsymbol{z})$, (Fong & Vedaldi, 2017) and (Dabkowski & Gal, 2017) additionally included two forms of regularization: upsampling and total variation penalization. Upsampling is used to optimize a coarser θ (e.g. 56×56 pixels), which is upsampled to the full dimensionality (e.g. 224×224) using bilinear interpolation. Total variation penalty smoothes θ by a ℓ_2 regularization penalty between spatially adjacent θ_u. To avoid losing too much signal from regularization, we use upsampling size 56 and no total variation effect unless otherwise mentioned. We examine the individual effects of these regularization terms in Appendices A.2 and A.4, respectively.

4 EXPERIMENTS

We first evaluate the various infilling strategies and objective functions for FIDO. We then compare explanations under several classifier architectures. In section 4.5 we show that FIDO saliency maps outperform BBMP (Fong & Vedaldi, 2017) in a successive pixel removal task where pixels are in-filled by a generative model (instead of set to the heuristic value). FIDO also outperforms the method from (Dabkowski & Gal, 2017) on the so-called Saliency Metric on ImageNet. Appendices A.1–A.5 provide further analysis, including consistency and the effects of additional regularization.

4.1 INFILLING METHODS

We describe several methods for producing the reference value $\hat{\boldsymbol{x}}$. The heuristics do not depend on $\boldsymbol{z}$ and are from the literature. The generative approaches, which produce $\hat{\boldsymbol{x}}$ by conditioning on the non-masked inputs $\boldsymbol{x}_{\boldsymbol{z}=0}$, are novel to saliency computation.

Heuristics: **Mean** sets each pixel of $\hat{\boldsymbol{x}}$ according to its per-channel mean across the training data. **Blur** generates $\hat{\boldsymbol{x}}$ by blurring $\boldsymbol{x}$ with Gaussian kernel ($\sigma = 10$) (Fong & Vedaldi, 2017). **Random** samples $\hat{\boldsymbol{x}}$ from independent per-pixel per-channel Gaussians ($\sigma = 0.2$) (Dabkowski & Gal, 2017).

Generative Models: **Local** computes $\hat{\boldsymbol{x}}$ as the average value of the surrounding non-dropped-out pixels $\boldsymbol{x}_{\boldsymbol{z}=0}$ (we use a 15×15 window). **VAE** is an image completion Variational Autoencoder (Iizuka et al., 2017). Using the predictive mean of the decoder network worked better than sampling. **CA** is the Contextual Attention GAN (Yu et al., 2018); we use the authors' pre-trained model.

Figure 3 compares these methods with a centered mask. The heuristic in-fills appear far from the distribution of natural images. This is ameliorated by using a strong generative model like CA, which in-fills texture consistent with the surroundings. See Appendix A.5 for a quantitative comparison.

4.2 COMPARING THE SSR AND SDR OBJECTIVE FUNCTIONS

Here we examine the choice of objective function between L_{SSR} and L_{SDR}; see Figure 6. We observed more artifacts in the L_{SDR} saliency maps, especially when a weak in-filling method (Mean) is used. We suspect this unsatisfactory behavior is due to the relative ease of optimizing L_{SDR}. There are many degrees of freedom in input space that can increase the probability of any of the 999 classes besides c; this property is exploited when creating adversarial examples (Szegedy et al., 2013). Since it is more difficult to infill unobserved pixels that increase the probability of a particular class c, we believe L_{SSR} encourages FIDO to find explanations more consistent with the classifier's training distribution. It is also possible that background texture is easier for a conditional generative model to fit. To mitigate the effect of artifacts, we use L_{SSR} for the remaining experiments.

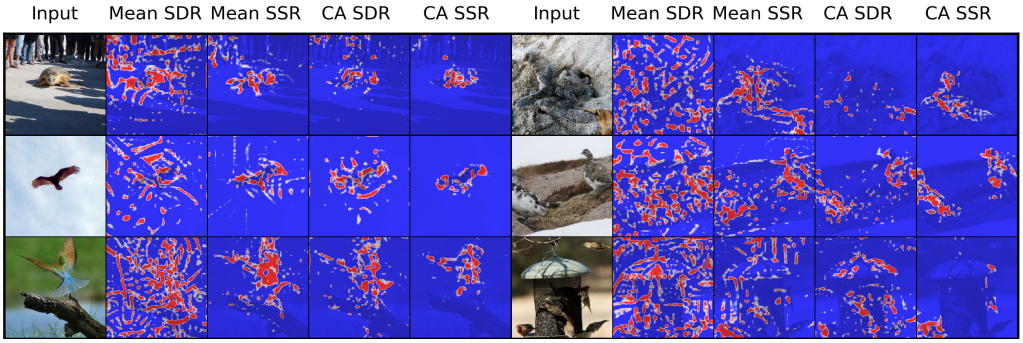


Figure 6: **Choice of objective between L_{SSR} and L_{SDR}.** The classifier (ResNet) gives correct predictions for all the images. We show the L_{SSR} and L_{SDR} saliency maps under 2 infilling methods: Mean and CA. Here the red means important and blue means non-important. We find that L_{SDR} is more susceptible to artifacts in the resulting saliency maps than L_{SSR}.

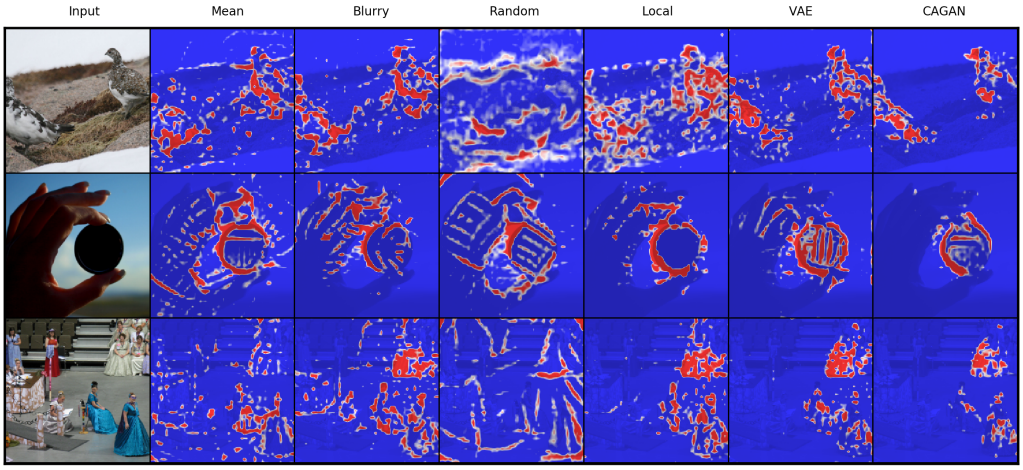


Figure 7: **Comparison of saliency map under different infilling methods** by SSR using ResNet. Heuristics baselines (Mean, Blur and Random) tend to produce more artifacts, while generative approaches (Local, VAE, CA) produce more focused explanations on the targets.

4.3 Comparing infilling methods

Here we demonstrate the merits of using strong generative model that produces substantially fewer artifacts and a more concentrated saliency map. In Figure 7 we generate saliency maps of different

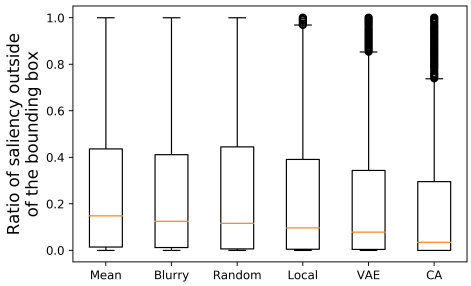


Figure 8: **Proportion of saliency map outside bounding box.** Different in-filing methods evaluating ResNet trained on ImageNet, $1,971$ images. The lower the better.

infilling techniques by interpreting ResNet using L_{SSR} with sparsity penalty $\lambda = 10^{-3}$. We observed a susceptibility of the heuristic in-filling methods (Mean, Blur, Random) to artifacts in the resulting saliency maps, which may fool edge filters in the low level of the network. The use of generative in-filling (Local, VAE, CA) tends to mitigate this effect; we believe they encourage in-filled images to lie closer to the natural image manifold. To quantify the artifacts in the saliency maps by a proxy: the proportion of the MAP configuration ($\theta > 0.5$) that lies outside of the ground truth bounding box. FIDO-CA produces the fewest artifacts by this metric (Figure 8).

4.4 Interpreting Various Classifier Architectures

We use FIDO-CA to compute saliency of the same image under three classifier architectures: AlexNet, VGG and ResNet; see Figure 9. Each architecture correctly classified all the examples. We observed a qualitative difference in the how the classifiers prioritize different input regions (according to the saliency maps). For example in the last image, we can see AlexNet focuses more on the body region of the bird, while Vgg and ResNet focus more on the head features.

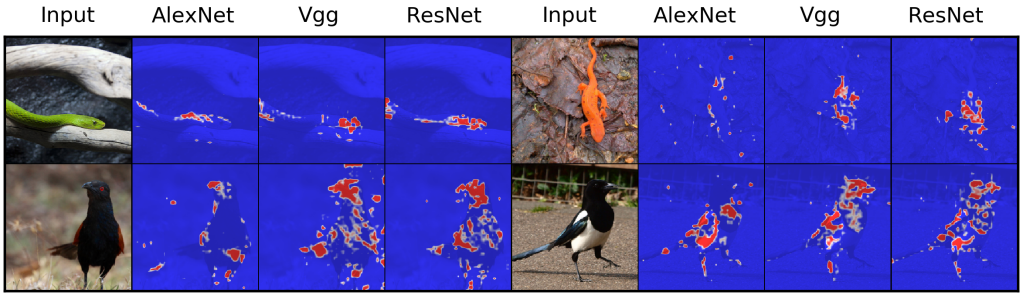


Figure 9: **Comparison of saliency maps for several classifier architectures**. We compare 3 networks: AlexNet, Vgg and ResNet using FIDO-CA with $\lambda = 10^{-3}$)

4.5 Quantitative Evaluation

We follow Fong & Vedaldi (2017) and Shrikumar et al. (2017) in measuring the classifier's sensitivity to successively altering pixels in order of their saliency scores. Intuitively, the "best" saliency map should compactly identify relevant pixels, so that the predictions are changed with a minimum number of altered pixels. Whereas previous works flipped salient pixel values or set them to zero, we note that this moves the classifier inputs out of distribution. We instead dropout pixels in saliency order and infill their values with our strongest generative model, CA-GAN. To make the log-odds score suppression comparable between images, we normalize per-image by the final log-odds suppression score (all pixels infilled). In Figure 10 we evaluate on ResNet and carry out our scoring procedure on 763 randomly-selected correctly-predicted ImageNet validation images, and report the number of pixels required to reduce the normalized log-odds score by a given percent. We evaluate FIDO under various in-filling strategies as well as BBMP with Blur and Random in-filling strategies. We attempt to put both algorithms on equal footing by using $\lambda = 1\text{e}{-3}$ for FIDO and $\lambda = 5\text{e} - 3$ for BBMP (see Section A.1 for further discussion). We find that strong generative infilling (VAE and CA) yields more parsimonious saliency maps, which is consistent with our qualitative comparisons. FIDO-CA can achieve a given normalized log-odds score suppression using fewer pixels than competing methods.

We compare our algorithm to a strong baseline on two established metrics[4]. We first evaluate whether the FIDO saliency map can solve weakly supervised localization (WSL) (Dabkowski & Gal, 2017). After thresholding the saliency map θ above 0.5, we compute the smallest bounding box containing all salient pixels. This prediction is "correct" if it has intersection-over-union (IoU) ratio over 0.5 with any of the ground truth bounding boxes. Using FIDO with various infilling methods, we report the average error rate across $1,000$ randomly selected validation images in Table 1. We evaluate the

[4]In Appendix A.6 we provide larger sample sizes for a subset of models

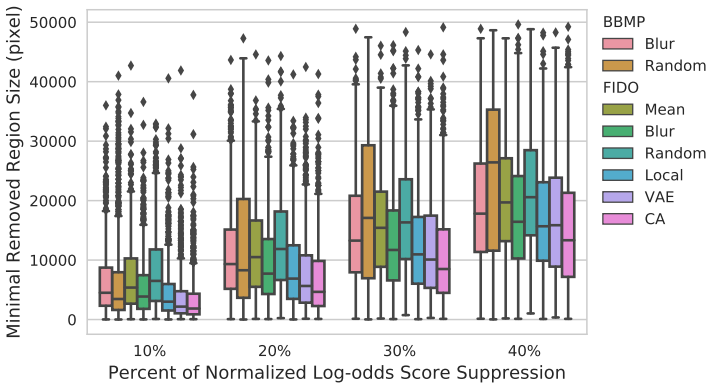


Figure 10: **Number of salient pixels required to change normalized classification score.** Pixels are sorted by saliency score and successively replaced with CA-GAN in-filled values. We select $\lambda = 5\text{e}{-3}$ for BBMP and $1\text{e}{-3}$ for FIDO. The lower the better.

			FIDO							
	Max	Center	Mean	Blur	Random	Local	VAE	CA	Dabkowski & Gal (2017)	BBox
WSL Err (1k)	63.5%	45.3%	51.7%	48.7%	53.3%	**38.1%**	43.5%	52.1%	38.5%	0.0%
SM (1k)	1.118	0.434	0.862	0.763	0.784	0.467	0.359	**-0.092**	0.129	0.432

Table 1: Weakly Supervised Localization (WSL) error and Saliency Metric (SM). FIDO (various in-filling methods) evaluating ResNet trained on ImageNet. For both metrics the lower the better.

authors' pre-trained model of Dabkowski & Gal (2017)[5]. We also include two simple baselines: Max (entire input as the bounding box) and Center (centered bounding box occupying half the image).

FIDO-CA frugally assigns saliency to contextually important pixels, preserving classifier confidence (Figure 4), so we do not necessarily expect our saliency maps to correlate with the typically large human labeled bounding boxes. Perhaps surprisingly, FIDO-Local performs comparably to Dabkowski & Gal (2017). The reliance on human-labeled bounding boxes makes WSL suboptimal for evaluating saliency maps, so we evaluate the so-called Saliency Metric proposed by Dabkowski & Gal (2017), which eschews the human labeled bounding boxes. The smallest bounding box A is computed as before. The image is then cropped using this bounding box and upscaling to its original size. The Saliency Metric is $\log max(\text{Area}(A), 0.05) - \log p(c|\text{CropAndUpscale}(\boldsymbol{x}, A))$ the log ratio of the bounding box area and the in-class classifier probability after upscaling. This metric represents the information concentration about the label within the bounded region; from the superior performance of FIDO-CA we conclude that a strong generative model regularizes explanations towards the natural image manifold and finds concentrated region of features relevant to the classifier's prediction.

5 Conclusion

We proposed FIDO, a new framework for explaining differentiable classifiers that uses adaptive Bernoulli dropout with strong generative in-filling to combine the best properties of recently proposed methods (Fong & Vedaldi, 2017; Dabkowski & Gal, 2017; Zintgraf et al., 2017). We compute saliency by marginalizing over plausible alternative inputs, revealing concentrated pixel areas that preserve label information. By quantitative comparisons we find the FIDO saliency map provides more parsimonious explanations than existing methods. FIDO provides novel but relevant explanations for the classifier in question by highlighting contextual information relevant to the prediction and consistent with the training distribution.

[5] We use the authors' PyTorch pre-trained model `https://github.com/PiotrDabkowski/pytorch-saliency`.

REFERENCES

Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. Learning to explain: An information-theoretic perspective on model interpretation. *arXiv preprint arXiv:1802.07814*, 2018.

Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 6970–6979, 2017. URL http://papers.nips.cc/paper/7272-real-time-image-saliency-for-black-box-classifiers.

R. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. *arXiv preprint arXiv:1705.07832*, 2017.

Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and Locally Consistent Image Completion. *ACM Transactions on Graphics (Proc. of SIGGRAPH)*, 36(4):107, 2017.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

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

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*, 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.

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

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *arXiv preprint arXiv:1801.07892*, 2018.

Jianming Zhang, Zhe Lin, Shen Xiaohui Brandt, Jonathan, and Stan Sclaroff. Top-down neural attention by excitation backprop. In *European Conference on Computer Vision(ECCV)*, 2016.

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.

A FURTHER ANALYSIS

A.1 COMPARISONS OF BBMP AND FIDO

Here we compare FIDO with two previously proposed methods, BBMP with Blur in-filling strategy (Fong & Vedaldi, 2017) and BBMP with Random in-filling strategy (Dabkowski & Gal, 2017). One potential concern in qualitatively comparing these methods is that each method might have a different sensitivity to the sparsity parameter λ. Subjectively, we observe that BBMP requires roughly 5 times higher sparsity penalty λ to get visually comparable saliency maps. In our comparisons we sweep λ over a reasonable range for each method and show the resulting sequence of increasingly sparse saliency maps (Figure 11).

We use $1\text{e}{-3}, 2\text{e}{-3}, 5\text{e}{-3}$ and 0.01 for BBMP methods and $2\text{e}{-5}, 5\text{e}{-4}, 1\text{e}{-3}$ and $2\text{e}{-3}$ for FIDO. We observe that all methods are prone to artifacts in the low λ regime, so the appropriate selection of this value is clearly important. Interestingly, BBMP Blur and Random respectively find artifacts with different quality: small patches and pixels for Blur and structure off-object lines for Random. FIDO with CA is arguably the best saliency map, producing fewer artifacts and concentrating saliency on small regions for the images.

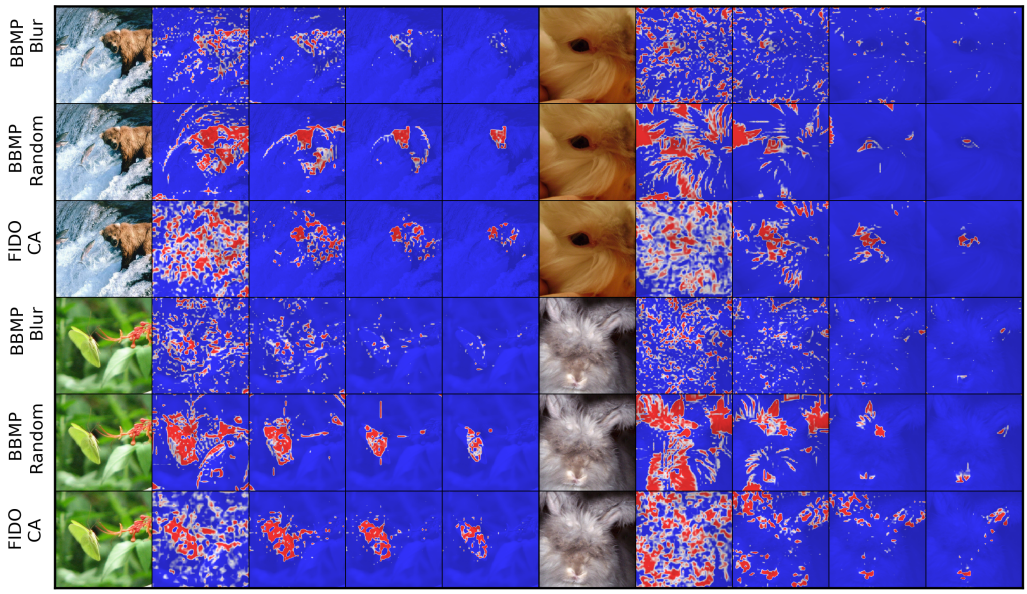


Figure 11: **BBMP vs FIDO saliency map** by increasing the sparsity penalty λ value from left to right. We compare with BBMP under Blur and Random, and FIDO under CA in-filling strategies. Note that here BBMP and FIDO methods use different λ (see main text for details). We show the heuristics produce more artifacts (BBMP Blur) or produces weird lines (BBMP Random) compared to our method.

A.2 UPSAMPLING EFFECT

Here we examine the effect of learning a reduced dimensionality θ that upsampled to the full image size during optimization. We consider a variety of upsampling rates, and in a slight abuse of terminology we refer to the upsampling "size" as the square root of the dimensionality of θ before upsampling, so smaller size implies more upsampling. In Figure 12, we demonstrate two examples with different upsampling size under Mean and CA infilling methods with SSR objectives. The weaker infilling strategy Mean apparently requires stronger regularization to avoid artifacts compared to CA. Note that although CA produces much less artifacts compared to Mean, it still produces some small artifacts outside of the objects which is unfavored. We then choose 56 for the rest of our experiments to balance between details and the removal of the artifacts.

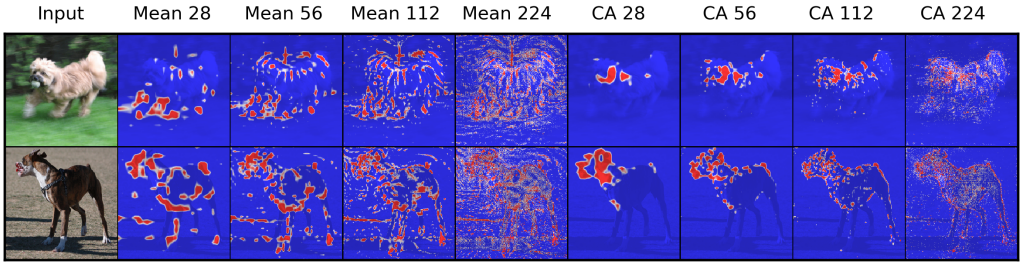


Figure 12: **Comparisons of upsampling effect** in Mean and CA infilling methods. We show the upsampling regularization removes the artifacts especially in the weaker infilling method Mean.

A.3 Stability

To show the stability of our method, we test our method with different random seeds and observe if they are similar. In Figure 13, our method produces similar saliency map for 4 different random seeds.

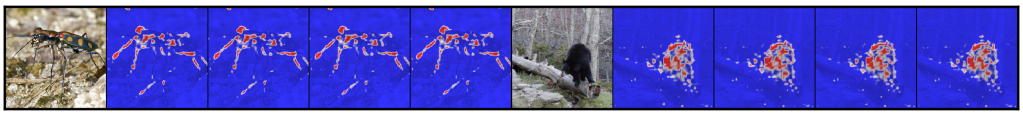


Figure 13: **Testing the stability of our method** with 4 different random seeds. They produce similar saliency maps. (Using CA infilling method with ResNet and $\lambda = 10^{-3}$)

A.4 Total Variation Effect

Although we do not include total variation smoothing prior in our experiments, we still test the effect of total variation regularization in Figure 14. We find the total variation can reduce the adversarial artifacts further, while risking losing signals when the total variation penalty is too strong.

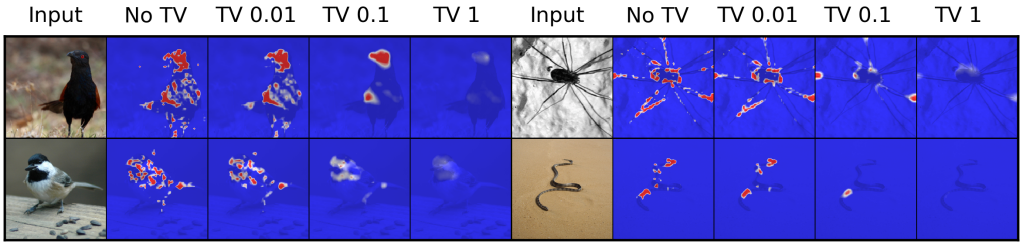


Figure 14: **Total Variation Regularization Effect**. We show the saliency maps of 4 increasing total variation (TV) regularization. We show that strong regularization risks removing signal.

A.5 Analysis of generative model infilling

Here we quantitatively compare the in-filling strategies. The generative approaches (VAE and CA) perform visually sharper images than four other baselines. Since we expect this random removal should not remove the target information, we use the classification probability of the ResNet as our metric to measure how good the infilling method recover the target prediction. We quantitatively evaluate the probability for $1,000$ validation images in Figure 15. We find that VAE and CA

consistently outperform other methods, having higher target probability. We also note that all the heuristic baselines (Mean, Blur, Random) perform much worse since the heuristic nature of these approaches, the images they generate are not likely under the distribution of natural images leading to the poor performance by the classifier.

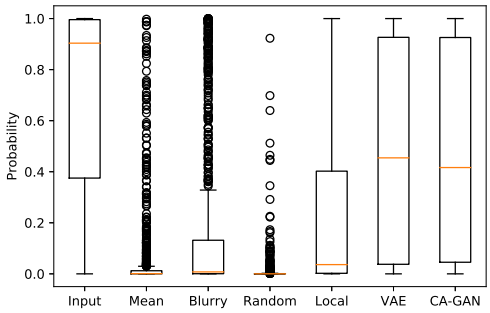


Figure 15: **Box plot of the classifier probability under different infilling with respect to random masked pixels** using ResNet under 1000 images. We show that generative models (VAE and CA) performs much better in terms of classifier probability.

A.6 Larger Sample Size for WSL and SM metrics

To verify the results from Table 1, we expand the sample size to $15,000$ randomly selected validation images from ImageNet. Table 2 shows the WSL and SM for FIDO-CA and the various baselines.

	Max	Center	FIDO-CA	Dabkowski & Gal (2017)	BBox
WSL Err (15k)	58.7%	42.5%	54.9%	**40.8%**	0.0%
SM (15k)	0.929	0.274	**-0.190**	-0.073	0.248

Table 2: Weakly Supervised Localization (WSL) error and Saliency Metric (SM). FIDO (various in-filling methods) evaluating ResNet trained on ImageNet. For both metrics the lower the better.

B More Examples

Figure 16 shows several more infilled counterfactual images, along with the counterfactuals produced by the method from Dabkowski & Gal (2017). More examples comparing the various FIDO infilling approaches can be found in Figure 17. We provide additional examples of FIDO-CA with total variation regularization alongside the method from Dabkowski & Gal (2017) in Figure 18.

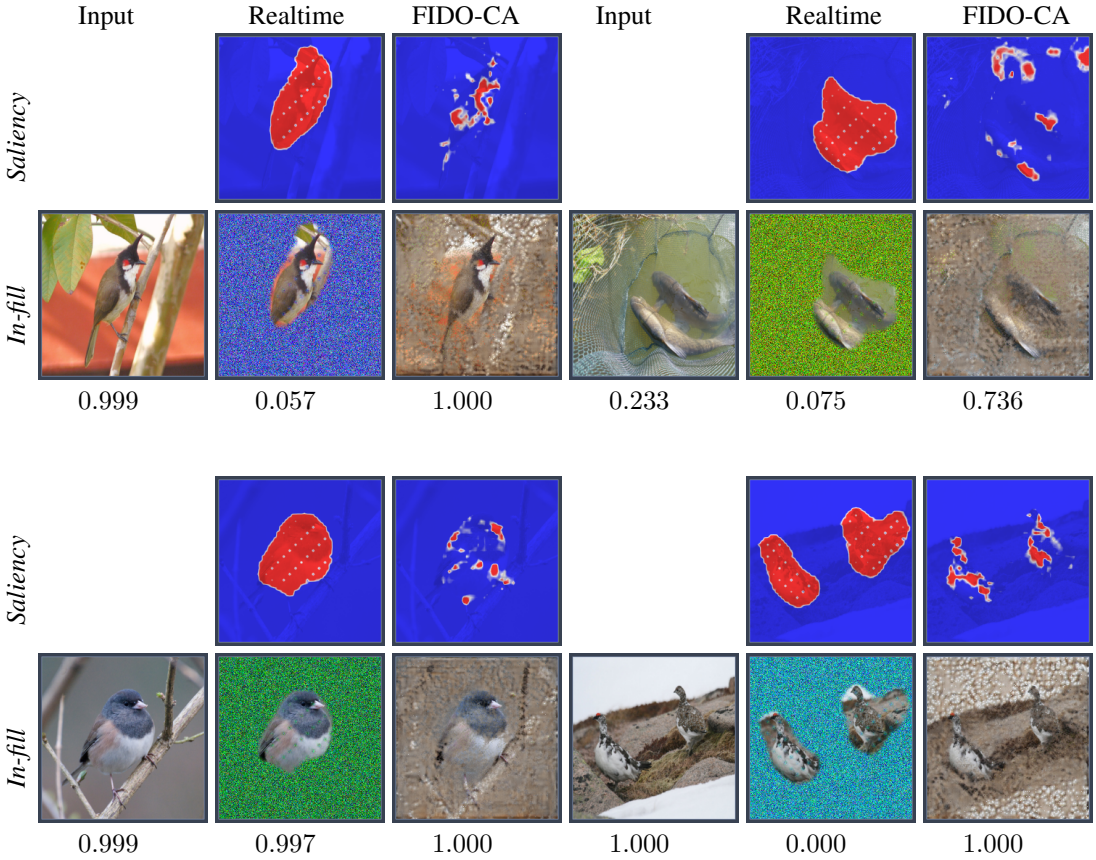


Figure 16: **More examples of classifier confidence on infilled images.** Realtime denotes the method of Dabkowski & Gal (2017); FIDO-CA is our method with CA-GAN infilling Yu et al. (2018). Classifier confidence $p(c|\hat{\boldsymbol{x}})$ is reported below the input and each infilled image. We hypothesize by that FIDO-CA is able to isolate compact pixel areas of contextual information. For example, in the upper right image pixels in the net region around the fish are highlighted; this context information is missing from the Realtime saliency map but are apparently relevant to the classifier's prediction.

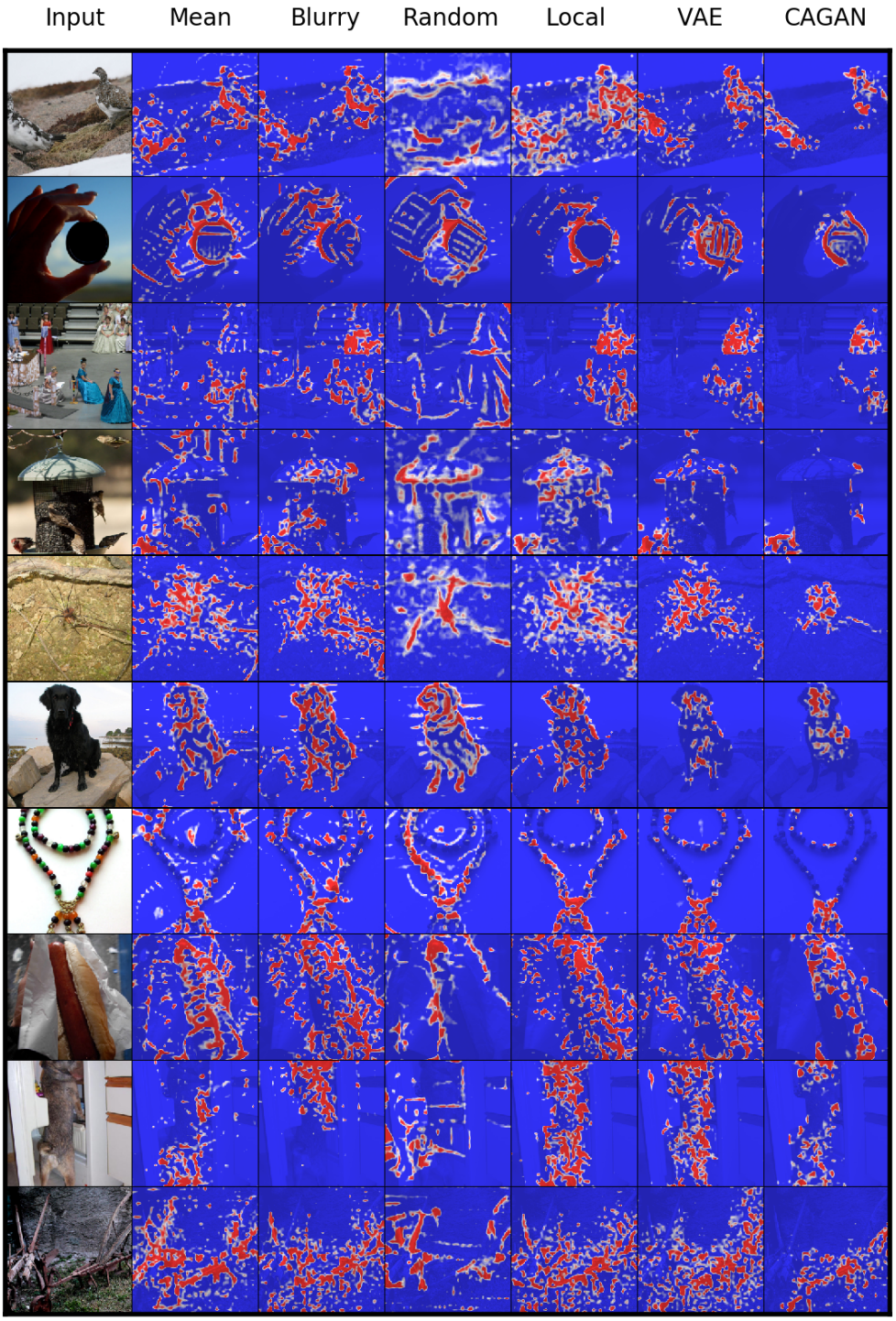


Figure 17: **Additional Saliency Maps** for FIDO with a variety of in-filling methods.

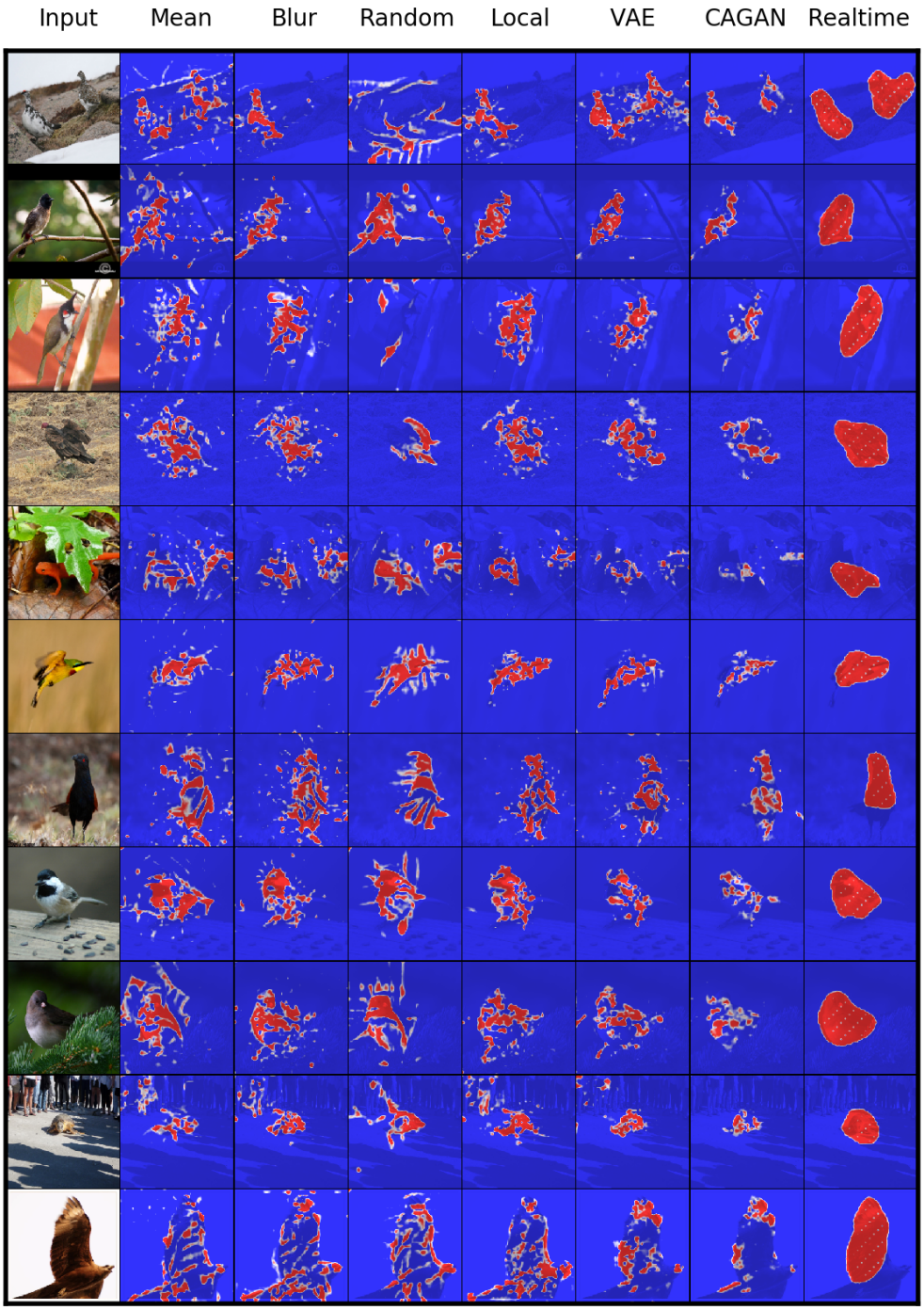


Figure 18: **Additional Saliency Maps** for FIDO under total variation 0.01 with a variety of in-filling methods. We include the method from Dabkowski & Gal (2017) in the right-most column.