Appendix

A Schematic description

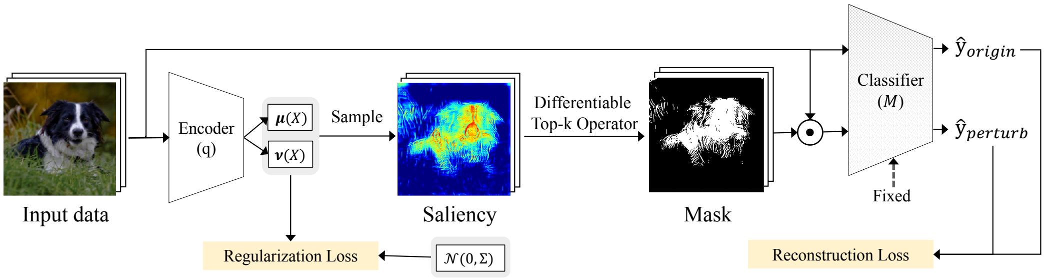
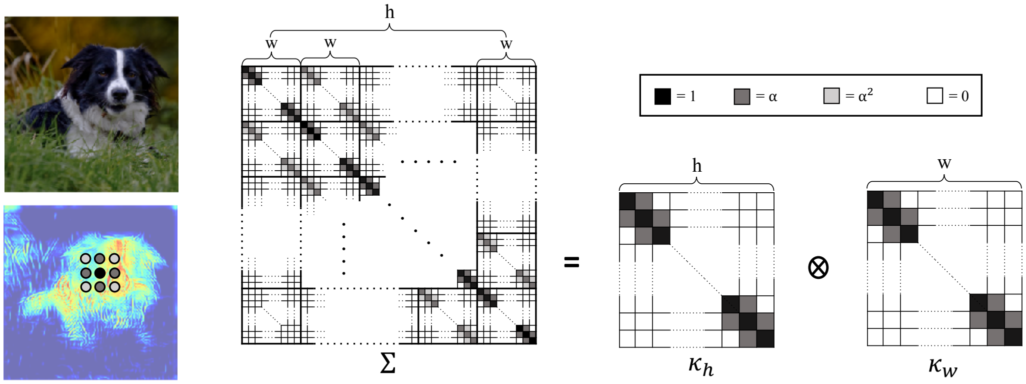


Figure 1: Schematic description.

An input image is fed into the encoder network that gives the mean and the variance of Guassian distribution (which is the approximate posterior). A saliency map is sampled from it, followed by passing a differentiable top-k operator to provide a binary mask. With this mask, the input image is perturbed. The perturbed image is passes a classifier M that gives categorical probability $\hat{\boldsymbol{y}}_{perturb}$. The loss function is composed of two terms: the reconstruction term between $\hat{\boldsymbol{y}}_{perturb}$ and the categorical probability obtained from the original image $\hat{\boldsymbol{y}}_{origin}$, and the regularization term between the approximate posterior $q(\boldsymbol{s}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{s}; \boldsymbol{\mu}(\boldsymbol{x}), \mathrm{diag}(\boldsymbol{\nu}(\boldsymbol{x})))$ and the prior distribution $\mathcal{N}(\boldsymbol{s}; \boldsymbol{0}, \boldsymbol{\Sigma})$. As the classifier M is the model that we aim to interpret, it is fixed so as the parameter not to be updated while training the encoder network.

B Soft-TV Gaussian prior



(a)

(b)

Figure 2: Soft-TV Gaussian prior.

We consider the prior knowledge that adjacent pixels in a saliency map have positive correlation (Figure 2(a)). After modeling the prior as Gaussian distribution $\mathcal{N}(\boldsymbol{s}; \mathbf{0}, \boldsymbol{\Sigma})$, the covariance matrix $\boldsymbol{\Sigma}$ is designed to infuse this prior knowledge into the prior distribution (Figure 2(b)). The covariance matrix is then decomposed by Kronecker product to better calculate the KL divergence of the regularization loss.

C Implementation details

Encoder architecture We use 17 convolution layers for the encoder network. To make the spatial size of the encoder's input and output to be same, we do not use a pooling layer. Every convolution layer is comprised of a convolution with kernel size 3×3, stride 1, and padding 1, followed by batch normalization and a rectified linear unit. The number of output channels for each convolution is as follows: $[64, 64, 64, 64, 32, 32, 32, 32, 32, 16, 16, 16, 16, 16, 16, 2]$. The encoder network provides $\boldsymbol{\mu} \in \mathbb{R}^{h \times w}$ and $\eta \in \mathbb{R}^{h \times w}$ for each channel of the output where $\boldsymbol{\mu}$ is the mean of the Gaussian distribution and $\eta = \log \nu$ with $\nu \in \mathbb{R}^{h \times w}$ the variance of the Gaussian distribution.

hyper-parameters We use Adam (Kingma & Ba, 2014) optimizer with learning rate to be 0.0001, weight decay to be 0.0005, and betas to be $(0.9, 0.99)$. We use batch size of 128 while training the encoder network. We run 10 epochs for the Imagenet dataset, and save the network that has the lowest loss.

D Proof of regularization loss equation

Let us first define the notation. $\otimes$ is the Kronecker product, and $\odot$ is the element-wise multiplication. $\mathrm{sum}(\cdot)$ is the summation of elements. For a vector $\boldsymbol{b} \in \mathbb{R}^{hw}$, we denote $\mathrm{rsh}(\boldsymbol{b}) \in \mathbb{R}^{w \times h}$ as reshaping the vector $\boldsymbol{b}$ to the matrix where the (i, j)-th entry is $\boldsymbol{b}[i + wj]$, and $\mathrm{diag}(\boldsymbol{b})$ as the diagonal matrix where the diagonal is $\boldsymbol{b}$. Also, for a square matrix $\boldsymbol{B}$, $\mathrm{diag}(\boldsymbol{B})$ is the vector where the i-th entry is $B[i, i]$. For a matrix $\boldsymbol{B}$, $\mathrm{vec}(\boldsymbol{B})$ denotes the vectorization by stacking the columns of the matrix $\boldsymbol{B}$ to a single column vector.

Recall that the approximate posterior is $q(\boldsymbol{s}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}_0)$ where $\boldsymbol{\Sigma}_0 = \mathrm{diag}(\boldsymbol{\nu})$ with $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^{hw}$, and the prior distribution is $p(\boldsymbol{s}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_1)$ where $\boldsymbol{\Sigma}_1 = \boldsymbol{\kappa}_h \otimes \boldsymbol{\kappa}_w$ with $\boldsymbol{\kappa}_h \in \mathbb{R}^{h \times h}$ and $\boldsymbol{\kappa}_w \in \mathbb{R}^{w \times w}$. KL divergence between two Gaussian distribution is:

$$
\begin{aligned}
D_{\mathrm{KL}}[\, q(\boldsymbol{s}|\boldsymbol{x}) \parallel p(\boldsymbol{s}|\boldsymbol{x}) \,] &= D_{\mathrm{KL}}[\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}_0) \parallel \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_1)\,] \\
&= \frac{1}{2}\left(\mathrm{tr}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0\right) + \boldsymbol{\mu}^T\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu} - hw + \log\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|}\right) . \qquad (1)
\end{aligned}
$$

We compute each term in RHS of equation 1 to make it computationally efficient.

$$
\begin{aligned}
\mathrm{tr}\left(\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0\right) &= \mathrm{tr}\left(\left(\boldsymbol{\kappa}_h \otimes \boldsymbol{\kappa}_w\right)^{-1} \cdot \mathrm{diag}\left(\boldsymbol{\nu}\right)\right) \\
&= \mathrm{tr}\left(\left(\boldsymbol{\kappa}_h^{-1} \otimes \boldsymbol{\kappa}_w^{-1}\right) \cdot \mathrm{diag}\left(\boldsymbol{\nu}\right)\right) \\
&= \mathrm{diag}\left(\boldsymbol{\kappa}_h^{-1} \otimes \boldsymbol{\kappa}_w^{-1}\right) \odot \mathrm{diag}\left(\boldsymbol{\nu}\right) \\
&= \mathrm{diag}\left(\boldsymbol{\kappa}_w^{-1}\right)^T \cdot \mathrm{rsh}\left(\boldsymbol{\nu}\right) \cdot \mathrm{diag}\left(\boldsymbol{\kappa}_h^{-1}\right) . \qquad (2)
\end{aligned}
$$

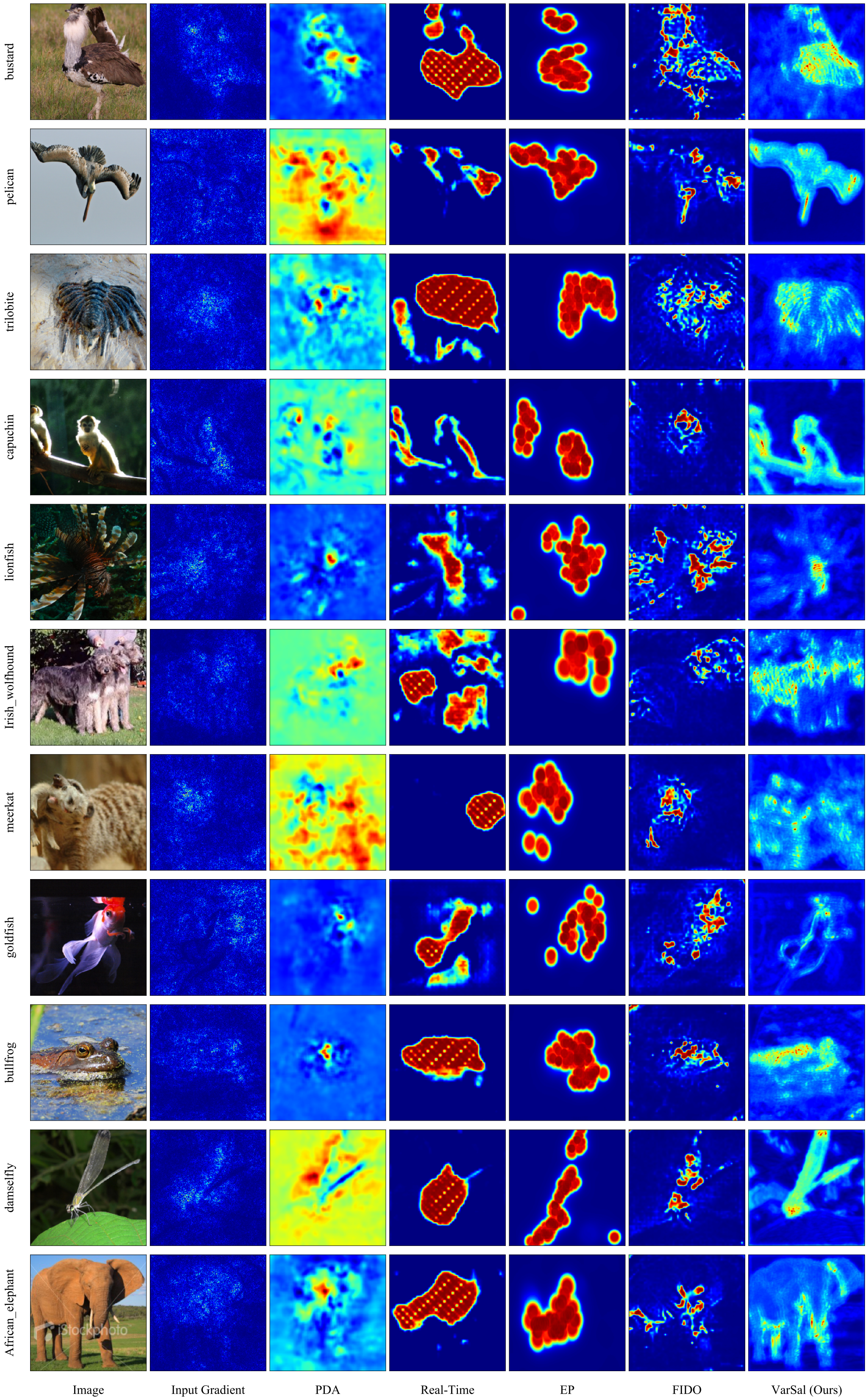
$$
\begin{aligned}
\boldsymbol{\mu}^T\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu} &= \boldsymbol{\mu}^T \cdot \left(\boldsymbol{\kappa}_h^{-1} \otimes \boldsymbol{\kappa}_w^{-1}\right) \cdot \boldsymbol{\mu} \\
&= \boldsymbol{\mu}^T \cdot \left(\boldsymbol{\kappa}_h^{-1} \otimes \boldsymbol{\kappa}_w^{-1}\right) \cdot \mathrm{vec}\left(\mathrm{rsh}\left(\boldsymbol{\mu}\right)\right) \\
&= \boldsymbol{\mu}^T \cdot \mathrm{vec}\left(\boldsymbol{\kappa}_h^{-1} \cdot \mathrm{rsh}\left(\boldsymbol{\mu}\right) \cdot \boldsymbol{\kappa}_w^{-1^T}\right) \\
&= \mathrm{sum}\left(\mathrm{rsh}\left(\boldsymbol{\mu}\right) \odot \left(\boldsymbol{\kappa}_w^{-1} \cdot \mathrm{rsh}\left(\boldsymbol{\mu}\right) \cdot \left(\boldsymbol{\kappa}_h^{-1}\right)^T\right)\right) . \qquad (3)
\end{aligned}
$$

$$
\begin{aligned}
\log\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|} &= \log|\boldsymbol{\kappa}_h \otimes \boldsymbol{\kappa}_w| - \log|\mathrm{diag}\left(\boldsymbol{\nu}\right)| \\
&= \log|\boldsymbol{\kappa}_h|^w|\boldsymbol{\kappa}_w|^h - \log\prod_{i=1}^{hw}\boldsymbol{\nu}_i \\
&= w \cdot \log|\boldsymbol{\kappa}_h| + h \cdot \log|\boldsymbol{\kappa}_w| - \log\prod_{i=1}^{hw}\boldsymbol{\nu}_i \\
&= -\mathrm{sum}\left(\log\boldsymbol{\nu}_i\right) + \mathrm{const}\,. \qquad (4)
\end{aligned}
$$

Therefore, the equation 1 is derived as:

$$
\begin{aligned}
D_{\mathrm{KL}}[\, q(\boldsymbol{s}|\boldsymbol{x}) \parallel p(\boldsymbol{s}|\boldsymbol{x}) \,] =\; &\mathrm{diag}\left(\boldsymbol{\kappa}_w^{-1}\right)^T \cdot \mathrm{rsh}\left(\boldsymbol{\nu}\right) \cdot \mathrm{diag}\left(\boldsymbol{\kappa}_h^{-1}\right) \\
&+ \mathrm{sum}\left(\mathrm{rsh}\left(\boldsymbol{\mu}\right) \odot \left(\boldsymbol{\kappa}_w^{-1} \cdot \mathrm{rsh}\left(\boldsymbol{\mu}\right) \cdot \left(\boldsymbol{\kappa}_h^{-1}\right)^T\right)\right) \qquad (5) \\
&- \mathrm{sum}\left(\log\boldsymbol{\nu}_i\right) + \mathrm{const}\,.
\end{aligned}
$$

E Qualitative results



References

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