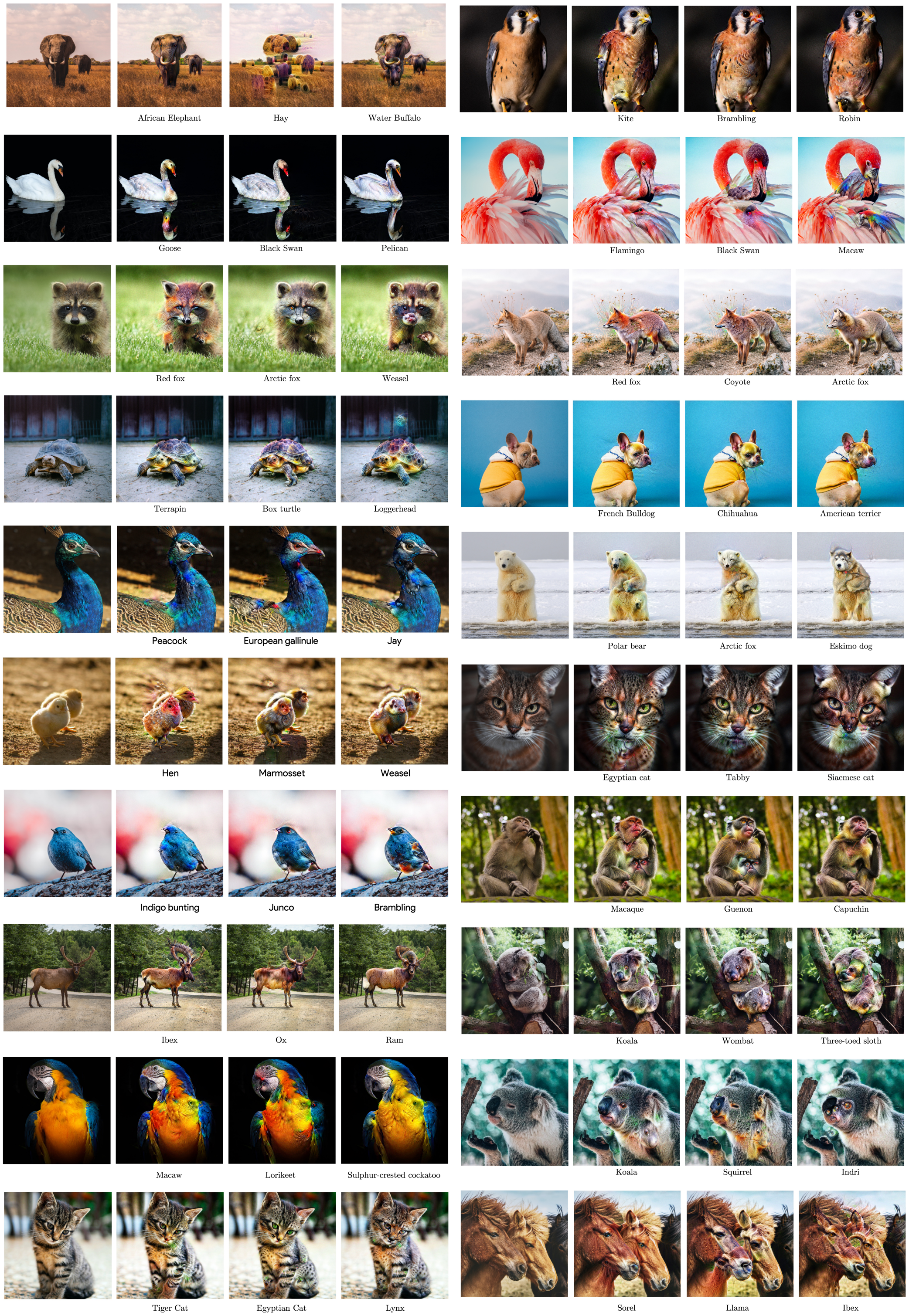
Appendix for Feature Accentuation



A MASKING

Here we will briefly expound upon our normalized attribution method, put forward in section 2.4. for a data sample $\boldsymbol{X}$ we consider a set of attribution maps $\{\boldsymbol{\varphi}(\boldsymbol{f_v}, \boldsymbol{x})\} \mid \boldsymbol{x} \in \boldsymbol{X}\}$. Let $\boldsymbol{v}$ represent the flattened, ordered vector of every scalar attribution in this set. We can then choose a percentile range $(p_1,\ p_2)$, such that values outside the range are fully masked/fully visible in the upsampled mask, and values in between are partially masked. We can then normalize every element, u, of a particular attribution map, $\boldsymbol{\varphi}(\boldsymbol{f_v}, \boldsymbol{x})$, by;

$$\mathcal{N}(u; \boldsymbol{v}, p_1, p_2) = \begin{cases} 0 & \text{if } u \leq P(\boldsymbol{v}, p_1) \\ \frac{u - P(\boldsymbol{v}, p_1)}{P(\boldsymbol{v}, p_2) - P(\boldsymbol{v}, p_1)} & \text{if } P(\boldsymbol{v}, p_1) < u < P(\boldsymbol{v}, p_2) \\ 1 & \text{if } u \geq P(\boldsymbol{v}, p_2) \end{cases} \tag{1}$$

By setting p_1, the user can control how globally salient a region must be before it constitutes a partial expression of the feature and can be unmasked. Conversely, by setting p_2, the user determines the percentile past which a feature is 'fully expressed', and fully unmasked. As previously stated, this approach is agnostic to the attribution method used; for example we applied this normalization approach to *gradCAM* (Selvaraju et al., 2017a) attribution maps to generate the masks for Figure 9, but simply used the features activation map itself as our attribution for latent accents in Figure 10, as convolutional layers are already spatialized.

B PATH EXPERIMENT DETAILS

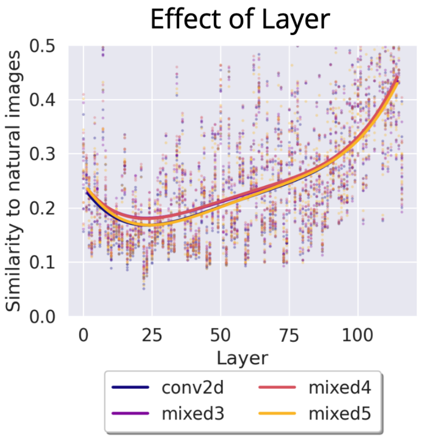


Figure 11: Small effect of regularization layer on path coherence.)

For each of the 5 tested (0,.1,.2,.5,10) we generated 1751 matched accentuations, corresponding to all the correctly predicted images across 50 random classes in the Imagenet(Deng et al., 2009) validation set. We accentuated each image toward its class label for 100 optimization steps using the Adam optimizer, with a .05 learning rate. We used 16 augmentations each optimizition step, cropping with a `maximum` box size of .99, and a scheduled `minimum` box size from .75 to .05. We regularized through layer {mixed3a}, but note that the optimal regularization layer seems to interact with the layer of $\boldsymbol{f_v}$. Besides this and λ, we observe the above hyperparameter produce quality accentuations in the general case.

In addition to our experiments with lambda, we tested the effect of regularization layer on the class-wise path similarity metric. We found only a small effect, with the earliest and latest layers tested (*conv2d0* and *mixed5a*) performing slightly worse, in agreement with our qualitative evaluation of the corresponding accentuations.

Figure 12 shows a random sample of our *super-natural* accentuations. These images are processed by way of hidden vectors that better correlate with natural class images than the natural images correlate with themselves.

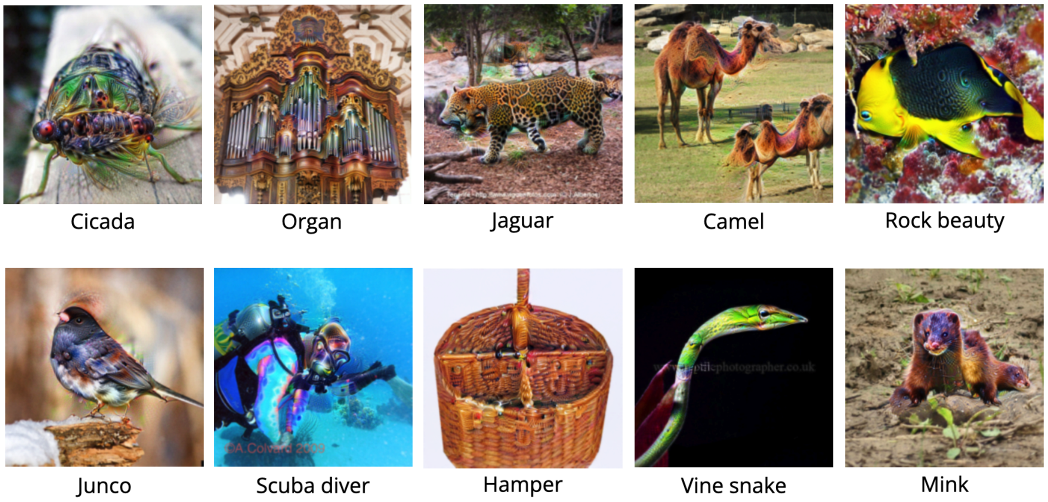


Figure 12: A sample of *Super-natural* class images.

C A note on Learning Rate

We observe that the learning rate is a far more sensitive parameter for feature accentutions than for the corresponding feature visualization. A *'good'* learning rate for a noise seeded feature visualizations will often be too large for feature accentuation, causing the visualization to deviate drastically from the target image in the the initial steps and never make it back. A small learning rate keeps the visualization perceptually similar to the seed image.

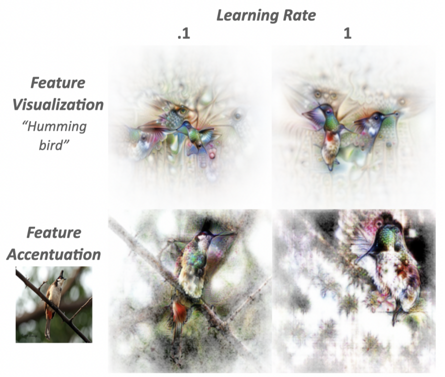


Figure 13: feature visualizations and accentuations for 'hummingbird' at two learning rates.

D Additional examples for Hyperparameters

In the interest of conserving space, we initially utilized a single example image to illustrate the impact of manipulating hyperparameters that we consider pivotal for *Feature accentuation*. However, to bolster our demonstration, we present here a series of supplementary examples drawn from our comprehensive testing dataset, aimed at showcasing the consistent and robust nature of the described effects across various instances. We deliberately choose to show some examples multiples times, so the reader can get a sense for how images change along multiple hyperparameter axes.

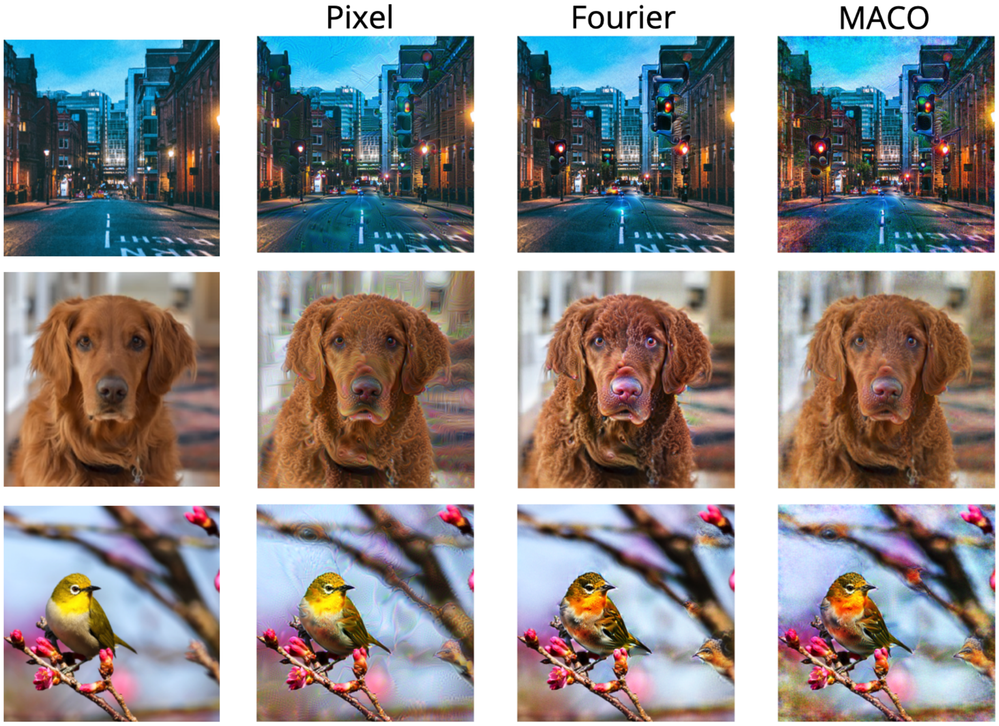


Figure 14: Additional example for our set of image parametrization.

Parametrization. The Fourier and MACO parametrization, while undoubtedly more intricate, emerge as good contenders in generating meaningful perturbations. On the other hand, the Pixel parametrization method, while comparatively simpler in its approach, easily introduce adversarial pattern. The seemingly basic alterations to pixel values can yield strikingly impactful results on latent representation, akin to the subtleties found in adversarial perturbations.

Augmentation. With small crop augmentations, the regularizer and feature detector are essentially *zooming in* and making local edits to the image. Unsurprisely, this yields crisper accentuations. However, small crops can also lead to miniature hallucinatory features scattered throughout the accnetuation, especially problematic given we seek explanations for the original, uncropped image. Supplying a uniform mixture of crops each optimization step seems to regularize against these hallucinations.

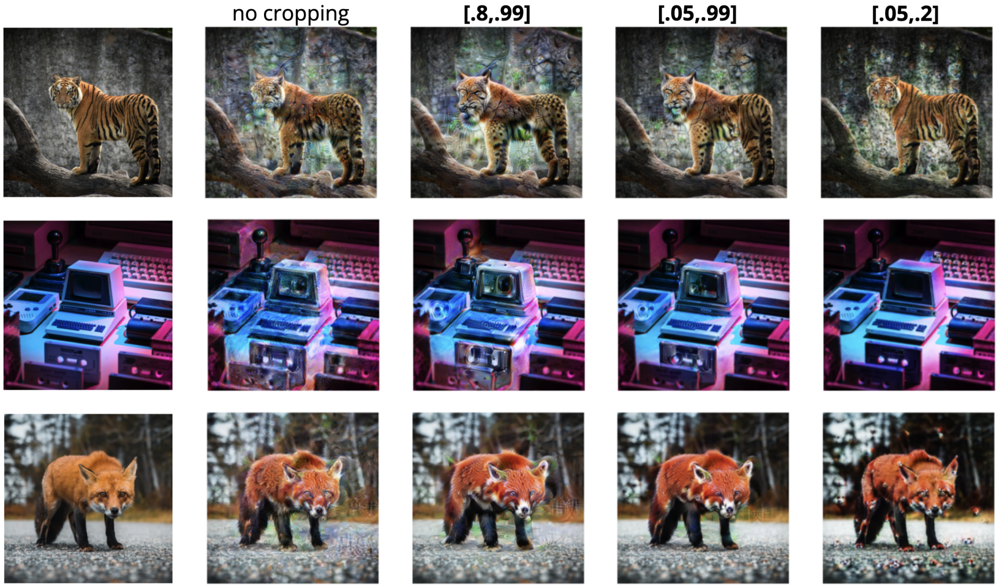


Figure 15: Additional regularization lambda examples

Regularization. One of the major hyperparameter influencing our optimization process is λ which serves as a regularizer. It plays a vital role in striking a balance between accentuation and preservation of the original data characteristics. More examples are shown confirming that a lower λ value can result in more pronounced accentuations, while a higher value tends to maintain the fidelity of the input data.

Effect of the Layer Examining the impact of layer selection within our neural network architecture, we provide additional examples showing the effect of this hyperparameter on *Feature accentuation*.

The choice of the layer has a notable influence on the perturbation: different layers within our neural network exhibit distinct tendencies in accentuating specific features or patterns within the input data. Early layer tends to preserve pixel information while latter seems to allow greater perturbation but preserve semantic (and often class) information.

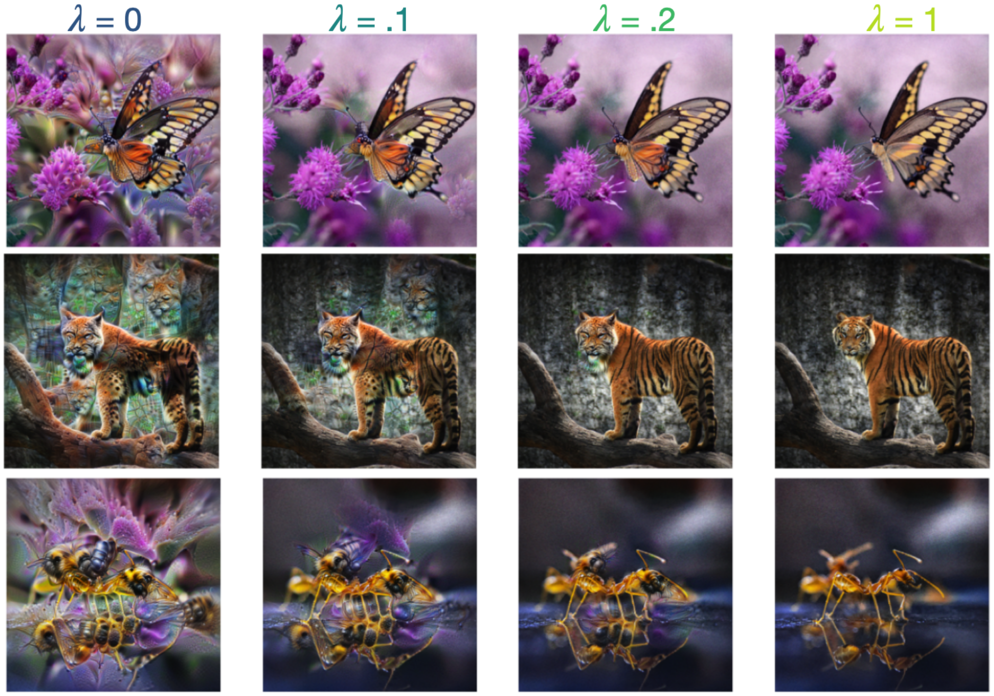


Figure 16: Additional regularization lambda examples

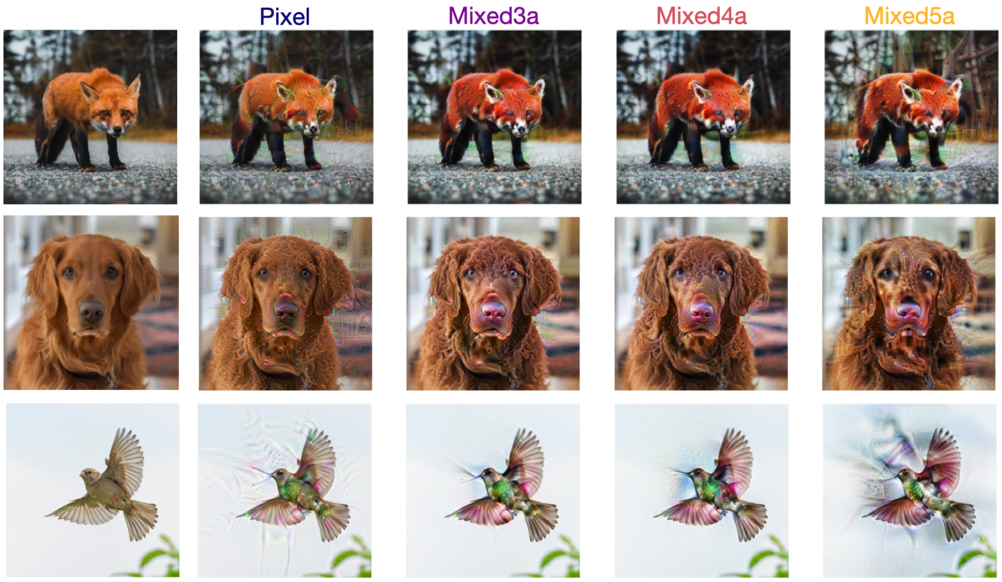


Figure 17: Additional regularization layer examples