Locally Explaining Prediction Behavior via Gradual Interventions and Measuring Property Gradients

Supplementary Material

Table of Contents

A. Method - Additional Details

In this section, we include additional details and discussions for our methodology. First, we describe the structural causal model (SCM), which is the foundation of our approach. Then, we comment on the second classifier-free guidance (CFG) scale [18], which recent image-to-image editing models implement, e.g., [4, 12]. While in our main paper, we provide arguments for input space interventions, in Section A.3, we discuss alternatives. To measure the impact of a property under gradual interventions, we propose to estimate the expected magnitude of the corresponding property gradients (Section 3.4). This score can be seen as an extension of the causal concept effect [15], which we show in Section A.4. Finally, we detail the corresponding shuffle hypothesis test and discuss some finer details in our approach. We also include example functional dependencies and a comparison of our score to the linear Pearson correlation coefficient [35].

A.1. SCM for Property Dependence

In our main paper, we provide a quick overview (Section 3.1 of the SCM that underpins our analysis (Fig. 2). Here, we provide a more detailed introduction from a different point of view.

First, to derive a structural causal model (SCM) or causal diagram after [34], we follow a related approach [39] and frame the structure of supervised learning. This then enables us to identify the variables determining the prediction behavior.

In supervised learning, the goal is to separate a latent data distribution based on task-specific reference annotations Y. This latent data distribution is an exogenous variable in a supervised learning system that we do not directly observe. Instead, we observe sampled data points or inputs, such as texts or images. We adopt the view that network inputs are a collection of properties, which collectively define the input sample [39].

In this work, we focus on two properties of the inputs: the task-dependent reference annotation Y and a property of interest X. These properties are not necessarily independent and can be causally related. Alternatively, they may correlate due to the sampling process or a confounding factor. This is particularly important because deep models are trained on finite, possibly biased samples of the true latent data distribution, where various spurious correlations can occur. Furthermore, the sampled training data $\mathcal{D}_{\text{train}}$ is strongly dependent on the task-related reference annotations Y and is not a random sample.

Given such a sample $\mathcal{D}_{\text{train}}$ of inputs with corresponding annotations and exogenous factors, such as optimizer and hyperparameter choices, we learn parameters θ for a model $\mathbb{F}$. For any input, $\mathbb{F}_\theta$ deterministically produces a corresponding output $\hat{Y}$. This output strongly depends on the learned weights θ, which in turn depends on $\mathcal{D}_{\text{train}}$. However, for an input sampled after training, i.e., during inference, we must consider a direct connection between Y and $\hat{Y}$ given that we directly optimize for this relationship. This connection should hold for any model outperforming random guessing on a held-out test dataset.

In contrast, we cannot be certain about the direct connection between the property of interest X and $\hat{Y}$, even if X and Y are correlated in $\mathcal{D}_{\text{train}}$, due to the automatic opti-

mization of $\mathbb{F}_\theta$. We summarize the described interactions in the causal diagram in Fig. 2.

To investigate the influence of a property X, [39] proposes to test conditional dependence in a collection of corresponding observations. While this approach can lead to global insights, they do not necessarily apply to individual local examples. To be specific, while the prediction behavior of a neural network may be influenced by properties such as hair color overall, other properties can locally dominate. Therefore, we propose gradually introducing changes in the property of interest X for an otherwise fixed input to investigate the influence on $\hat{Y}$ encoded in $\mathbb{F}_\theta$. We describe our approach in Section 3.3.

A.2. CFG scale for Image Alignment

In our main paper, we focus on the alignment with the image-edit instruction (Equation (1)) for modern generative models [4, 12]. However, in practice, these models implement a second condition: the alignment with the original image. In the terminology of [4], Equation (1) becomes

$$\begin{aligned}\bar{e}(z_t, c_T, c_I) =& e(z_t, \varnothing, \varnothing) \\ &+ s_I(e(z_t, c_T, \varnothing) - e(z_t, \varnothing, \varnothing)) \\ &+ s_T(e(z_t, c_T, c_I) - e(z_t, c_T, \varnothing)).\end{aligned} \tag{3}$$

Again, we omit the parameterization of e for brevity.

Equation (3) includes two guiding scales which together determine the intervention. In Fig. 11 in Appx. B.3, we perform a small ablation and find that the CFG text scale predominantly controls the intervention. Hence, in our work, we focus on s_T and fix s_I depending on the task.

A.3. Input Intervention Alternatives

Multiple options exist for intervening on a property X of interest. In principle, we could directly change the property value after extracting it. However, deep models are not designed to operate on property-level inputs. Instead, they expect inputs that conform to their learned input domain, such as images represented as pixel matrices for vision models. Therefore, we can explore two alternative possibilities: intervening in the input space or modifying the latent representations extracted by a model $\mathbb{F}_\theta$. In our main paper, we focus on input interventions. In this section, we discuss the alternative of intervening in a trained model's latent space.

Specifically, we identify multiple limitations of latent space interventions compared to our approach. Such latent representations are inherently non-interpretable, and while methods exist to ascribe meaning to changes in neurons, e.g., [22, 47, 58], the representations are polysemantic [10]. In other words, neurons often encode multiple different semantic properties or concepts simultaneously. This behavior is problematic because, in contrast to the input space, we cannot simply visualize the interventions. Hence, we could introduce unknown confounding.

Recent work focuses on disentangling latent representations into human interpretable representations often based on sparse autoencoders, e.g., [6, 8, 14, 30, 46]. However, these approaches are model-specific, require significant implementation overhead, and provide no guarantee that the property of interest X is learned or extracted by the model. To be specific, the latent representation contains no information about an unlearned property from an information-theoretic point of view. Hence, a latent vector does not uniquely map to one input. In fact, two inputs that only differ in an unlearned property can result in the same latent vector. In other words, it might be impossible to investigate many properties deemed important by the user via interventions in latent space. Among the advantages we list in the main paper, we believe that this last point is a crucial advantage of intervening in input space.

A.4. Connection to Causal Concept Effect for Binary Properties

In [15], the authors introduce the causal concept effect (CaCE), defined as

$$\begin{aligned}\text{CaCE}(\mathbb{F}, \mathsf{X}) = \mathbb{E}_g[\mathbb{F}(I)|do(\mathsf{X}=1)] \\ - \mathbb{E}_g[\mathbb{F}(I)|do(\mathsf{X}=0)],\end{aligned} \tag{4}$$

where g denotes the generative process to perform the intervention. Similarly to us, the authors discuss various options, including generative models [15]. Note that we exchange some of the original symbols by our notation and use $\mathbb{F}$ for the classifier and X for the concept or property of interest. Equation (4) describes it for binary properties, but [15] further extends it to N-wise categorical by pairwise comparisons against the observed state for an input. Nevertheless, we focus here on the version in Equation (4).

To show the connection to our expected gradient magnitude, we assume an intervention with binary property states. Without loss of generality let $\mathsf{x} \in \{0, 1\}$. Then, Equation (2) becomes

$$\mathbb{E}_{\mathsf{X}}[|\nabla_{\mathsf{X}}\mathbb{F}_\theta(I_{\mathsf{X}})|] = \frac{1}{2}\sum_{\mathsf{x}\in\{0,1\}} |\nabla_{\mathsf{x}}\mathbb{F}_\theta(I_{\mathsf{x}})|. \tag{5}$$

In this limited binary case, we approximate the gradient by calculating the difference between the model outputs for both property states. Specifically, we get

$$\frac{1}{2}(|\mathbb{F}_\theta(I_0) - \mathbb{F}_\theta(I_1)| + |\mathbb{F}_\theta(I_1) - \mathbb{F}_\theta(I_0)|), \tag{6}$$

for the two possible property states in the binary example. Here, the two gradient magnitudes are equal, finally resulting in $|\mathbb{F}_\theta(I_0) - \mathbb{F}_\theta(I_1)|$ as the measured effect.

Note that in our work, we focus on local inputs and interventions to not violate the causal hierarchy theorem for

image editing [32]. Nevertheless, by taking the expectation over a generative process for I_{x}, we get $\mathbb{E}_g[|\mathbb{F}_\theta(I_0) - \mathbb{F}_\theta(I_1)|]$. This expectation includes an absolute value. However, it is clearly a related quantity to the CaCE in Equation (4). To illustrate this further, assume $\mathbb{F}_\theta(I_0) > \mathbb{F}_\theta(I_1)$ and rewrite using the do operator notation, then applying the linearity of expectation, we arrive at the right-hand side of Equation (4).

In our work, we take a further step by analyzing gradual interventions, such as those enabled by [12]. Specifically, we utilize CFG scaling [18] to generate ordered variations, allowing us to approximate the respective gradients for various settings of the property of interest using [11]. Here, gradients, as shown above, extend CaCE [15] and enable a more nuanced understanding of the relationships between inputs and outputs. While our main focus is on local explanations, we note that a consequential approach to deriving global explanations would be to take a second expectation over a probing or test dataset of inputs. However, it is crucial to carefully scrutinize the specific interventions to ensure they do not violate [32] and to confirm variations in the property of interest. Notably, this approach is similar to our findings in our first experiment (see Section 4.1), where we investigated multiple images (see also Section B.3).

Why do we need property gradients? To illustrate the advantage of our approach over CaCE, we consider a classic example from the causal literature, as seen in [3]. In this scenario, we aim to estimate the causal effect of administering a drug, which would typically involve collecting interventional data through a randomized control trial. However, the dosage variable is often not binary, and varying dosages can have different effects. For instance, administering the drug in extremely high doses may cancel out its beneficial effects, resulting in a negligible or no treatment effect. In other words, the desired impact is only achieved for specific values of the dosage property.

Our property gradients capture these nuanced effects (see Fig. 9h). Furthermore, using visualizations similar to those in our experiments enables the identification of the desired property band, in this case, the optimal dosage range. While we use medicine as an example, similar behavior can occur for properties learned by neural networks. In fact, hair color is likely to exhibit a similar pattern in real life. Specifically, we note that gray hair is correlated with high age, whereas completely white or platinum blond hair is a common hair dye choice for younger individuals.

A.5. Expected Property Gradient Magnitude Estimates & Hypothesis Test

In our main paper, we propose estimating the expected gradient magnitude, as shown in Equation (2). To achieve this, we employ [11] with a finite discrete list of sampled inputs

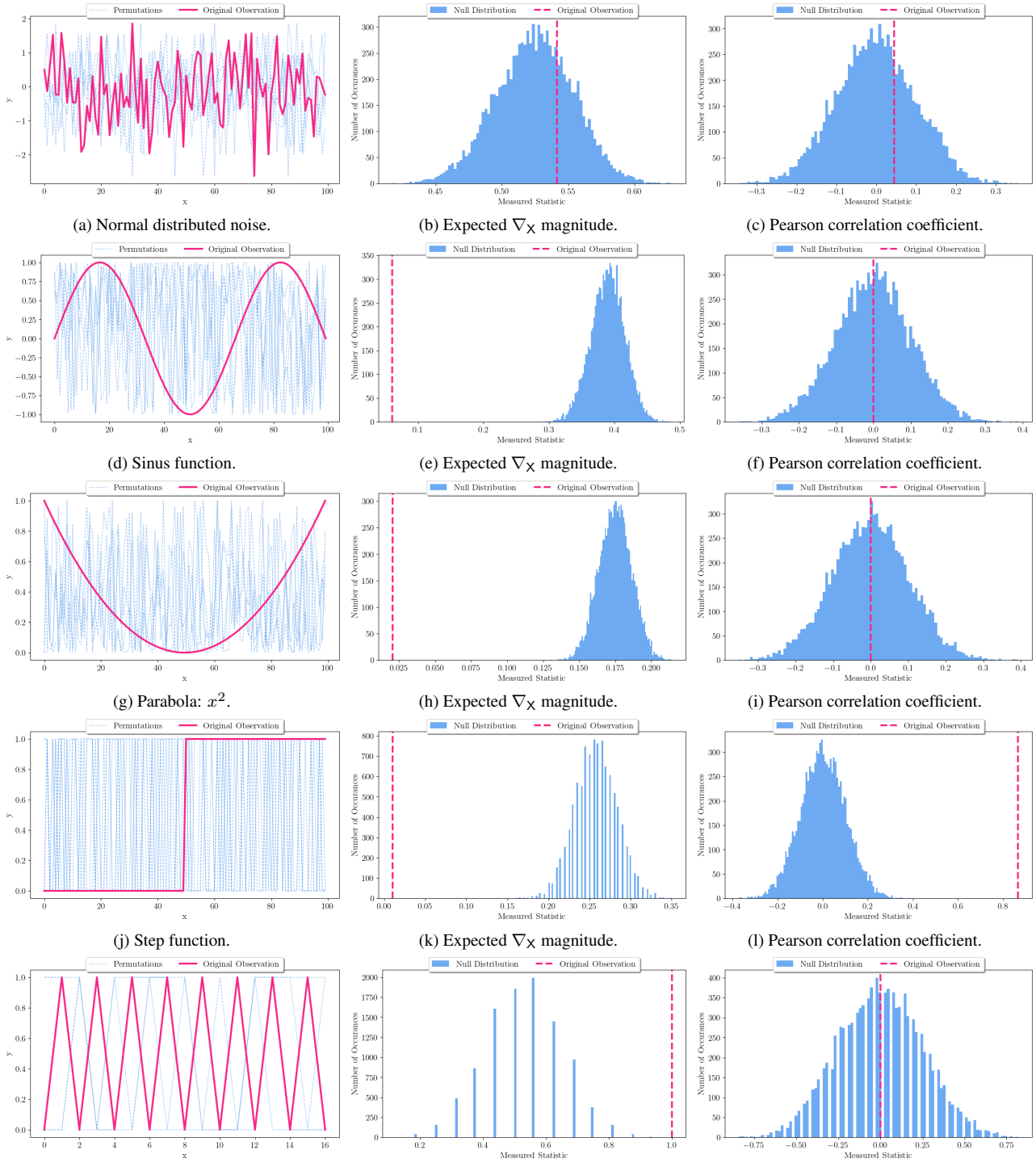
Algorithm 1 Hypothesis test for changes in prediction behavior for variations in a property X.

Require: ordered list of predictions $\mathbb{F}_\theta(I_{\mathsf{X}})$ ▷ N elements
Require: test statistic $\mathcal{S}$ ▷ for the outputs
Require: integer $K > 0$ ▷ Number of Permutations
Require: $\delta \in (0, 1)$ ▷ Significance Level
  $p \leftarrow 0.0$
  $\sigma_{orig.} \leftarrow \mathcal{S}(\mathbb{F}_\theta(I_{\mathsf{X}}))$ ▷ Estimate the original statistic
  for $i \in \{1, ..., K\}$ **do**
    $\mathbb{F}_\theta(I_{\mathsf{X}}^{(perm.)}) \leftarrow \texttt{permute}(\mathbb{F}_\theta(I_{\mathsf{X}}))$
    $\sigma_{perm.} \leftarrow \mathcal{S}(\mathbb{F}_\theta(I_{\mathsf{X}}^{(perm.)}))$
    if $\texttt{compare}(\sigma_{orig.}, \sigma_{perm.})$ **then**
      ▷ Comparison depends on $\mathcal{S}$ ($<$, $>$, two-sided, etc.)
      $p \leftarrow p + 1/K$ ▷ Increment the p-value
    end if
  end for
  if $p < \delta$ **then**
    `return` significant.
  else
    `return` not significant.
  end if

using an interventional strategy, such as [12]. To assess significance, we utilize the procedure outlined in Algorithm 1. In Fig. 9, we illustrate various example relationships, including independent random noise. Additionally, we visualize the estimated null distributions of Algorithm 1 using both Equation (2) and Pearson's correlation coefficient [35] as statistics. Notably, Equation (2) can detect periodic changes and changes with no linear trend while rejecting random noise, even with high effect strength. We highlight two key observations: First, as shown in Fig. 9n, our test statistic is two-sided. Second, we can investigate relationships where theoretically no gradient exists, i.e., ∇_{X} is infinite. Specifically, we consider step functions a relevant case, such as Fig. 9j, where models exhibit categorical changes in behavior upon reaching a certain threshold. However, by approximating gradients for discrete observations using [11], we can circumvent the issue of nonexisting gradients, enabling us to approximate Equation (2) even in such cases.

B. Cats vs. Dogs - Additional Details

In this section, we include additional details regarding our first experiment (Section 4.1). First, we detail the creation of the biased training and test splits of the Cats versus Dogs (CvD) [7] dataset. Next, we present the training details and hyperparameters of the classifiers under analysis. Additionally, we describe the interventional data generation process and extended results that reinforce the claims made in our main paper. Finally, we conduct a comparative analysis of



(a) Normal distributed noise. (b) Expected ∇_X magnitude. (c) Pearson correlation coefficient.

(d) Sinus function. (e) Expected ∇_X magnitude. (f) Pearson correlation coefficient.

(g) Parabola: x^2. (h) Expected ∇_X magnitude. (i) Pearson correlation coefficient.

(j) Step function. (k) Expected ∇_X magnitude. (l) Pearson correlation coefficient.

(m) Zigzag function. (n) Expected ∇_X magnitude. (o) Pearson correlation coefficient.

Figure 9. Visualization of the null distributions generated by Algorithm 1 for some example functions. The left-most column contains the observed functions versus five permuted instances. The middle column uses Equation (2) as a test statistic, while the right-most column employs the Pearson correlation coefficient [35]. Note that in the last row, for the zigzag function, we use the forward difference quotient instead of the central difference quotient to estimate ∇_X [11].

Table 4. Number of samples in the different splits for the CvD [7] dataset. With dark cat bias, we indicate dark-furred cats and light-furred dogs, while dark dog bias refers to the opposite.

Split	Class	Training Samples	Test Samples
Unbiased	cat	4000	1011
	dog	4005	1012
Dark cats bias	cat	1740	457
	dog	2141	546
Dark dogs bias	cat	2260	554
	dog	1864	466

multiple local and global XAI baselines in this scenario, highlighting the advantages of our local interventional approach in interpreting prediction behavior.

B.1. Creating a Biased Scenario

Our first experiment is based on a binary classification task between cats and dogs [7], where we intentionally introduce a correlation between the reference annotation and the fur color of the pictured animals. To create biased training and test splits, we leverage recent advances in multimodal models, specifically LLaVA 1.6 [24]. We prompt LLaVA with a yes/no question regarding the fur color of the reference annotation. Specifically, we use "Answer the question with yes or no: Does the {reference annotation} have dark fur?" and sort the images into corresponding biased splits based on the response. We assume that a "no" answer implies a light fur color, which is supported by our manual verification. The resulting sizes of the three training and test splits are summarized in Table 4. In the following section, we will detail the hyperparameter choices for the classification models.

B.2. Training Details

For the first experiment, we select the ConvMixer architecture [55], a simple yet effective convolution-only model class. Specifically, our ConvMixer configuration consists of an initial patch size of 5, a depth of 8, kernels with a width of 7, and a latent representation size of 256. For a detailed explanation of these hyperparameters, we refer the reader to the original paper [55].

During both training and inference, we preprocess the images by resizing them to an input size of 128×128 and normalizing the pixel values to the interval $[-1, 1]$. Furthermore, we apply the TrivialAugment data augmentation technique [31] with the wide augmentation space during training to enhance model robustness. We optimize the models using AdamW [28], setting the learning rate to 0.001, weight decay to 0.0005, and momentum to 0.9. After training for 100 epochs with batch size 64, we save the final model weights, which achieve the performances disclosed in Table 5.

Figure 10. Two failure cases we observed for the background interventions. In both, we tried various settings. Here, we report them using a 2.9 image guiding scale and 13.50 and 8.85 CFG text scale for the cat and dog, respectively. The left-hand side in both rows shows the original input image.

Table 5. Final test set accuracies in percent (%) achieved by our models trained to differentiate cats and dogs. Here, the columns signify the test data, and the rows denote the training data split.

	unbiased	dark cats split	dark dogs split
Unbiased	90.71	**92.52**	89.22
Dark Cats split	69.40	**93.92**	45.78
Dark Dogs split	70.93	48.45	**92.75**

B.3. Additional Results

Fur Color Intervention - 2D Results To perform the interventions on the fur color, we utilize a pre-trained version of [12], an image-to-image edit model that can be controlled using text prompts. This model utilizes a multimodal large language model (LLaVA [24]) together with an adapter network to provide expressive and focused edit instructions for instruct pix2pix [4]. The authors find that it performs especially well for local edits, meaning color changes, for example, do not change global illumination. Further, it includes two hyperparameters to control the alignment with the provided text instruction. Both are implemented and trained as classifier-free guidance (CFG) scales [18]. The first one, which we call the CFG text scale, generally controls the alignment with the instruction, while the CFG image scale

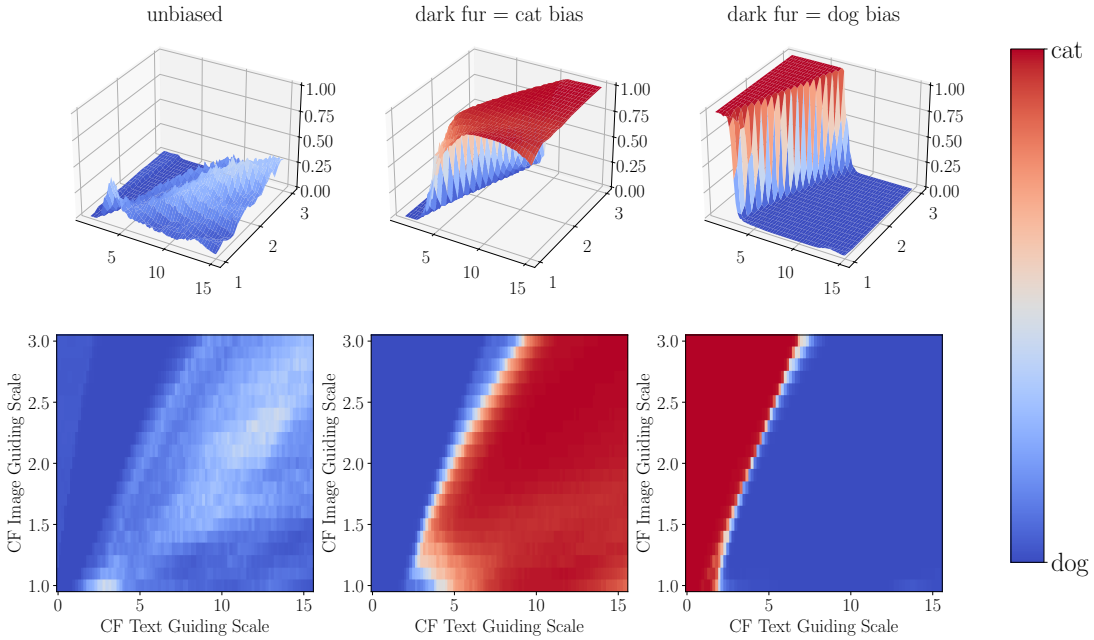


Figure 11. Changes in model predictions for an intervention on the fur color. Here, we display one ConvMixer model per column, specifically an unbiased one, one trained on only dark-furred dogs and light-furred cats ("dark fur = dog"), and one trained only on the opposite split ("dark fur = cat"). The x and y axes indicate the CF guiding scales for text and images, respectively. The white color (logit = 0.5) indicates the decision threshold between the two classes.

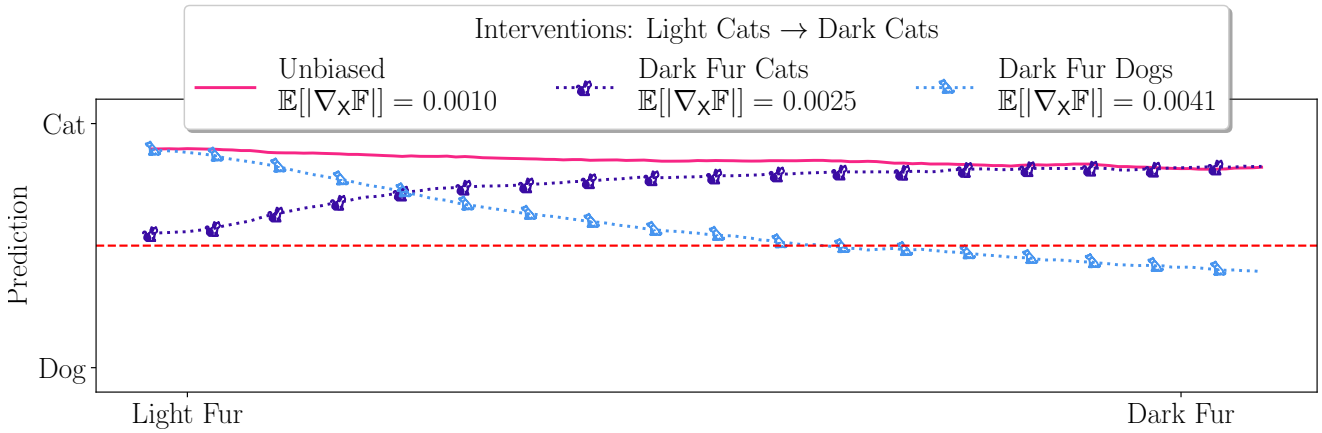
controls the similarity with the input image. We provide more details in Section A.2.

In our main paper (Fig. 1), we utilize a CFG image scale of 2.0 and interpolate the text scale between $[1.05, 14.7]$ using a stepsize of 0.15. We use "change the fur color to black" as our instruction. As an ablation, we visualize the results for the fur color intervention again in Fig. 11 for both guiding scales, varying the image scale between 1.0 and 3.0, following [12]. Notably, the CFG image scale primarily controls the point at which the prediction flip occurs. Specifically, we observe that the order of the predictions remains unchanged. The unbiased model does not exhibit a prediction flip but instead increases the activation for the cat class logit as the text guiding scale increases. In contrast, the model trained on dark dogs showcases the most abrupt change, while the dark cat model transitions slightly more gradually.

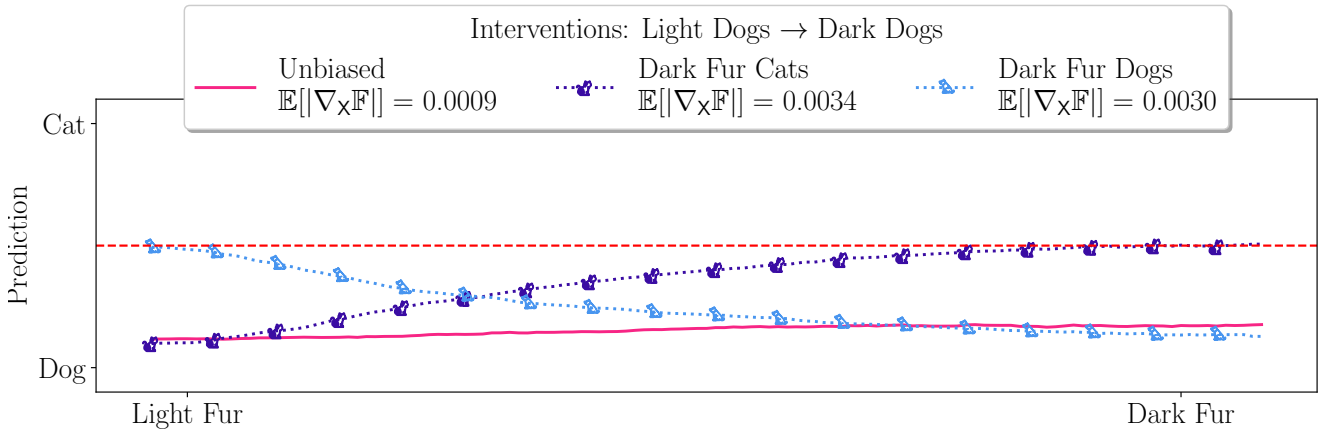
In summary, our results indicate that the actual intervention is controlled by the alignment with the image edit instruction. Therefore, in our subsequent experiments, we focus on the CFG text scale while selecting a suitable CFG image scale.

Additional Fur Color Interventions In our main paper, we present the average model predictions for all dark-furred cat images in the test dataset (see Fig. 3). Here we provide the remaining splits. To generate these images, we select the dog and cat images according to their initial fur color as described in Sec. 4.1. We use the following editing prompts: "change the fur color to dark black" for images of light-furred animals and "change the fur color to bright white" for images of dark-furred animals. Specifically, we employ an image guiding scale of 2.8 and vary the text guiding scale between 1.05 and 14.7. We provide examples of these edited images in Fig. 14. In all cases, we observe the lowest fur color impact for the unbiased model, while the models trained on the biased splits showcase the expected changes in prediction behavior.

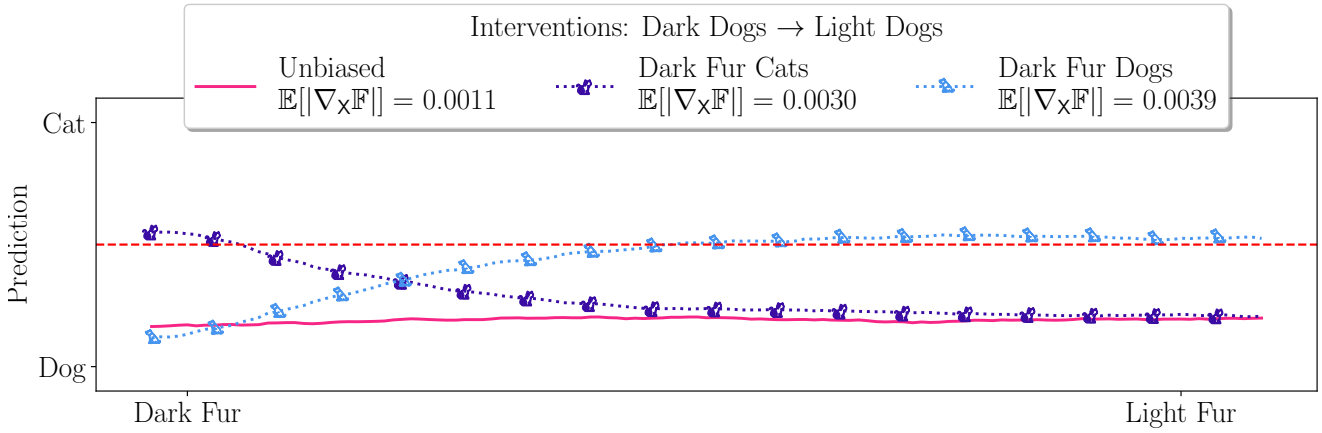
Further, in Fig. 13, we zoom in and visualize not only the mean but also the local behavior for ten selected images, where we verified correct interventions. While the overall trend remains consistent, indicating that the biased models flip their predictions on average, we occasionally observe a weaker relationship. We report the concrete measured $\mathbb{E}[|\nabla_{\mathsf{x}}|]$ values for these ten images in Table 6. Notably, here, we find generally higher $\mathbb{E}[|\nabla_{\mathsf{x}}|]$ scores compared to the average over all images, indicating that manually veri-



(a) Light furred cats → Dark furred cats.

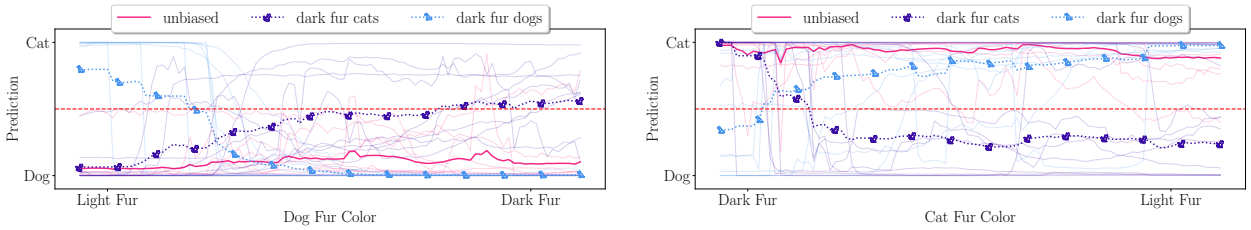


(b) Light furred dogs → Dark furred dogs.



(c) Dark furred dogs → light furred dogs.

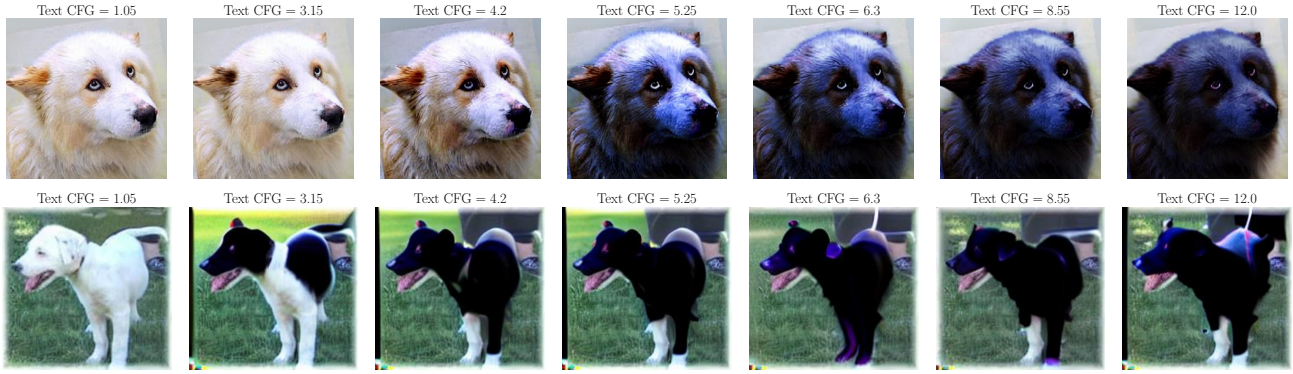
Figure 12. Average model output behavior for the property label combinations not contained in the main paper (compare to Fig. 3). Here, we use three ConvMixer models: an unbiased one, one trained on only dark-furred dogs and light-furred cats ("Dark Fur Dogs"), and one trained only on the opposite split ("Dark Fur Cats"). The red dotted line indicates the threshold where the model prediction flips. We include the average $\mathbb{E}[|\nabla_x|]$ per model in the legend. In all cases, we observe the lowest fur color impact for the unbiased model, while the models trained on the biased splits showcase the expected changes in prediction behavior. We provide **ten** local explanations for light-furred dogs and dark-furred cats in Fig. 13



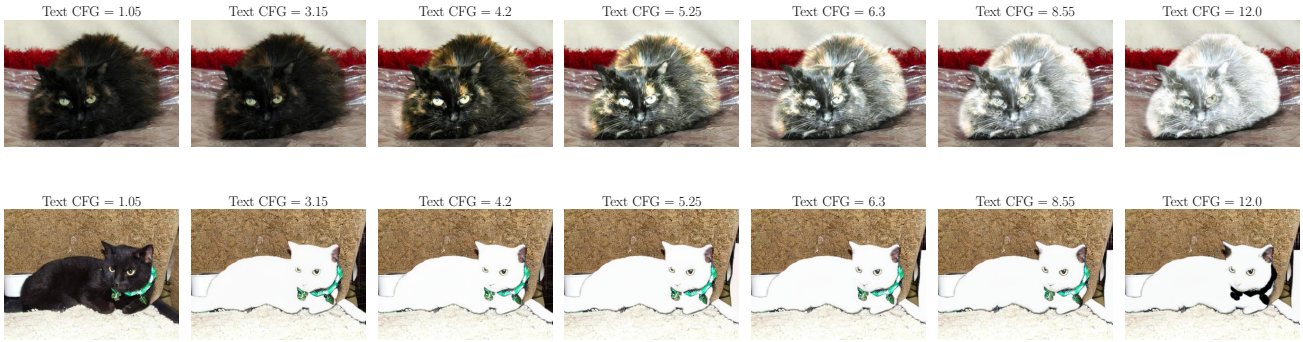
(a) Ten **dog** images with fur color interventions.

(b) Ten **cat** images with fur color interventions.

Figure 13. Changes in model predictions for an intervention on the fur color for ten respective images. Here, we use three ConvMixer models: an unbiased one, one trained on only dark-furred dogs and light-furred cats ("dark fur = dog"), and one trained only on the opposite split ("dark fur = cat"). The red dotted line indicates the threshold where the model prediction flips.



(a) Two additional example interventions for dogs with light fur.



(b) Two example interventions for cats with dark fur.

Figure 14. Interventions on the fur color for both cats and dogs. We display two additional examples per class. Note the difference in the onset for both examples per animal, given that we utilize the same hyperparameters and prompt for [12].

fied interventions benefit our measurements. Nevertheless, the ordering between the unbiased and the biased models persists, and we consistently measure a higher $\mathbb{E}[|\nabla_x|]$ for the two biased models. We find a stronger decrease in the intervention effect for light-furred dogs compared to dark-furred cats. Furthermore, the unbiased model (see Fig. 3) is the only model in our extended analysis where the averaged prediction never flips.

Finally, the discrepancies between the $\mathbb{E}[|\nabla_x|]$ values in the averaged case and the individual local case can be attributed to the varying onsets of the behavior changes. This phenomenon is also reflected in the variations observed in the interventions (Fig. 14). Specifically, as shown in Fig. 13a, the local flips occur at different CFG scales, highlighting the limitations of the behavior visualization. This observation underscores the non-linear mapping learned by [12] to intervene on the original image. Further, this mapping locally varies, which supports our claims made in the

Table 6. $\mathbb{E}[|\nabla_{\mathsf{X}}|]$ of the fur color property for our three CvD models. Here, we utilize the mean behavior (Fig. 3) over ten images. Additionally, we report significance ($p < 0.01$) and prediction flips. For examples of the interventional data, see Fig. 14.

		Model Behavior		
Data	Model	$\mathbb{E}[\|\nabla_{\mathsf{X}}\|]$	$p < 0.01$	Pred. Flips
Light Dogs	unbiased	0.00598	✓	✗
	dark cats	0.00863	✓	✓
	dark dogs	0.00888	✓	✓
Dark Cats	unbiased	0.00638	✓	✗
	dark cats	0.01207	✓	✓
	dark dogs	0.01084	✓	✓

Table 7. $\mathbb{E}[|\nabla_{\mathsf{X}}|]$ of the background property for our three CvD models. Here, we utilize the mean behavior (Fig. 16) over ten images. Additionally, we report significance ($p < 0.01$) and prediction flips. For examples of the interventional data, see Fig. 15.

		Model Behavior		
Data	Model	$\mathbb{E}[\|\nabla_{\mathsf{X}}\|]$	$p < 0.01$	Pred. Flips
Light Dogs	unbiased	0.00289	✓	✗
	dark cats	0.00015	✓	✗
	dark dogs	0.00193	✓	✗
Dark Cats	unbiased	0.00080	✓	✗
	dark cats	0.00004	✓	✗
	dark dogs	0.00188	✓	✗

main paper.

Additional Background Interventions To intervene on the background illumination, we utilize the following prompt with [12]: "[Darken/Brighten] the background color, keep the fur color unchanged". However, in contrast to the foreground interventions, we observe various failure cases. We illustrate two problems we encountered in Fig. 10.

In the first case, we observe minimal changes in the background, while instead, we notice more pronounced changes in the foreground, which we explicitly aim to avoid. In the second failure case, we witness a complete shift in the subject, in addition to the correct intervention in the background. We hypothesize that these observations are an expression of the causal hierarchy theorem for image edits [32]. Specifically, in [32], the authors demonstrate that even if a model correctly learns the training distributions, it does not guarantee that it learns the correct causal structure at a higher level of the PCH [3].

The backbone of [12], Instruct-Pix2Pix [4], was trained using synthetic interventional data produced using [17]. We believe that the background interventions are examples of image edits that are uncommon in the synthetic training data.

Nevertheless, these failure cases reinforce our conviction that human oversight is currently necessary to elevate explanations of local behavior to the interventional level, as discussed in the main paper in Section 5. Therefore, to further investigate the background interventions, we employ our second identified approach to generate interventional data and utilize image processing. Specifically, we assume a centered subject and reduce and utilize a centered Gaussian kernel to multiplicatively increase or decrease the pixels that are closer to the image edge. We provide examples in Fig. 15.

Fig. 16 illustrates the mean behavior changes, and we summarize the corresponding $\mathbb{E}[|\nabla_{\mathsf{X}}|]$ values in Table 7. Furthermore, Fig. 17 visualizes the local behavior for the different local images under the background interventions.

Our analysis largely confirms the local observations made in Section 4.1. Specifically, we observe significantly lower $\mathbb{E}[|\nabla_{\mathsf{X}}|]$ values for all models under the background interventions compared to the fur color interventions. Nevertheless, all measured changes in behavior are statistically significant, as determined by Algorithm 1. However, on average, none of the models exhibit flips in their predictions. Despite this, we observe in Fig. 17 that for some images, the models flip predictions for the stronger stages of the intervention. Notably, the unbiased model in Fig. 17a incorrectly predicts `cat` for some of the intervened dog images. We hypothesize that this is a consequence of our designed interventional strategy, which can also affect the foreground subjects. These results provide additional evidence to support the arguments raised in Section 5 and highlight the importance of visually inspecting the interventional data and corresponding model behavior.

B.4. Local Baseline XAI Results

In this section, we generate multiple other local explanations and highlight the challenges in extracting meaningful semantic insights from the results. In contrast, our approach presented in the main paper analyzes the changes in output for variations in a property, yielding actionable explanations. We also demonstrate how previous approaches can benefit from incorporating local interventions.

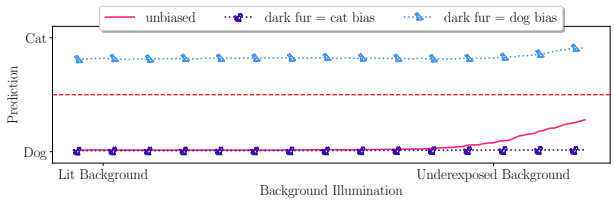
Setup We select the following methods to generate visual explanations for individual examples [29, 41, 49, 52, 53, 59]. We choose these methods due to their widespread usage and ease of accessibility. In any case, for all chosen methods, we utilize the implementations from [23] to ensure consistency and reproducibility.

These local methods highlight areas in the input that

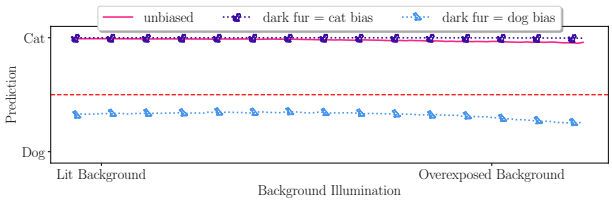
(a) Two additional example interventions for dogs with light fur.

(b) Two example interventions for cats with dark fur.

Figure 15. Interventions on the background for both cats and dogs. For both classes, we include two examples. After observing failure cases (see Fig. 10) using [12], we design the intervention using Gaussian kernels to scale pixels close to the image edges.



(a) Average of ten **dog** images with background interventions.



(b) Average of ten **cat** images with background interventions.

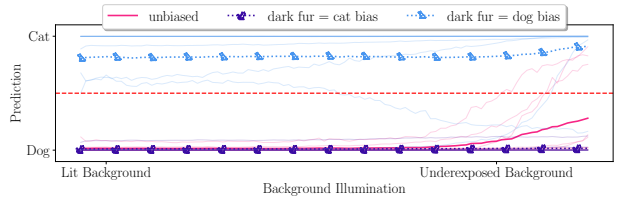
Figure 16. Changes in model predictions for an intervention on the background illumination for ten respective images. Here, we use three ConvMixer models: an unbiased one, one trained on only dark-furred dogs and light-furred cats ("dark fur = dog"), and one trained only on the opposite split ("dark fur = cat"). The red dotted line indicates the threshold where the model prediction flips.

speak for or against the selected class (sometimes both). Specifically, we select the `Cat` logit of our models to stay as comparable as possible to our main paper. We now detail specific hyperparameter choices to ensure reproducibility:

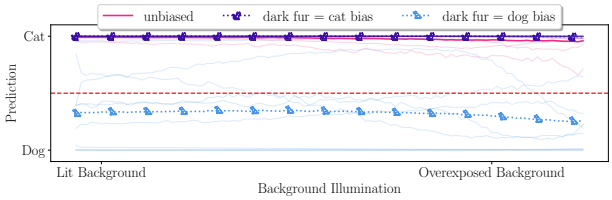
- or Guided Grad-CAM [49], we generate explanations with respect to the last convolutional layer of our models.
- For the occlusion-based method [59], we use the average gray scale value with a window size of 9×9. Additionally, we employ a stride length of four in both x and y directions.
- For LIME [41], and Kernel-SHAP [29], we utilize SLIC [2] super-pixels with 100 segments and a compactness of one as a basis.

For all other hyperparameters, we use the default settings provided by [23].

Results Fig. 18 presents visualizations of the selected local XAI baselines, including attribution maps for both the original image and an image with an intervention on the fur



(a) Ten **dog** images with background interventions.



(b) Ten **cat** images with background interventions.

Figure 17. Changes in model predictions for an intervention on the background illumination for ten respective images. Here, we use three ConvMixer models: an unbiased one, one trained on only dark-furred dogs and light-furred cats ("dark fur = dog"), and one trained only on the opposite split ("dark fur = cat"). The red dotted line indicates the threshold where the model prediction flips.

color.

Notably, in many cases, the most important regions identified by these methods align with the head of the dog. For gradient-based methods, such as [49, 52, 53], this suggests that changes in these pixels lead to significant changes in the `cat` class logit. Similarly, the occlusion-based approach [59] indicates that occluding the head has the highest impact. LIME [41] and Kernel-SHAP [29] also highlight regions that support or contradict the prediction locally, correctly marking the head in the respective color of the corresponding bias. Furthermore, investigating the intervened images reveals that the colors flip.

This observation suggests a change in behavior at the local level, which can be explained by the highlighted image regions. However, visual explanations require semantic interpretation to identify the human-understandable property driving this change. By examining only the left-hand side of Fig. 18, it is challenging to determine whether the head shape, ears, fur color, or another property is the causal factor. Interventional images can facilitate this interpretation. By providing both the original image and an image with a black fur intervention, we simplify the interpretation of the visualizations. In contrast, our gradual interventional approach yields a direct explanation with respect to a specific property. Further, estimating $\mathbb{E}[|\nabla_X|]$ provides a structured way to measure the corresponding local impact. However, our approach is not mutually exclusive with other local explanation methods but rather can complement them in future works, as seen in Fig. 18.

B.5. Global Baseline XAI Results

In this section, we evaluate our approach by comparing it to global explanations derived for the cats versus dogs model. To facilitate this comparison, we first outline the experimental setup and introduce the chosen methods for generating global explanations.

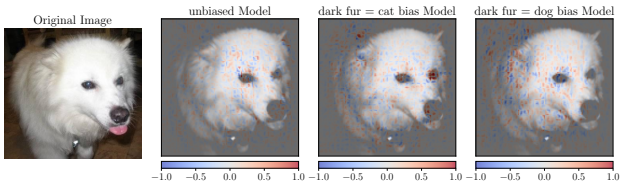
Setup For the comparison with global methods, we select three approaches: [39, 47, 58]. These methods provide a diverse range of techniques for generating global explanations, allowing us to evaluate our approach.

First, we utilize [58] to identify important concepts in the model. This method uses a probing dataset to identify concepts in a specific layer and orders them by importance using Shapley values (SHAP) [51]. Specifically, it finds concepts that maximize completeness for all classes. They then remove duplicates, where the concept activation vectors (CAVs) have a similarity of over 95%. To generate explanations for a specific class, the top-K image patches closest in the activation space are selected. We follow this approach and select the three CAVs with the highest SHAP values and show the top six images per CAV.

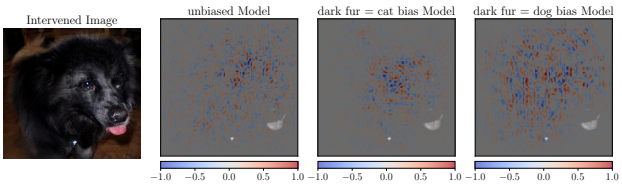
Next, we employ [47] to identify semantic concepts. This method builds on [58] and aims to identify textual descriptions for specific discovered concepts. Towards this goal, it performs a comparison to a set of texts in CLIP [38] latent space using cosine similarities. For both [58] and [47], we target the last convolutional layer of our trained cat versus dog networks and use the validation set of ImageNet [45] as the probing dataset. Additionally, for [47], we use the 20K most common Google terms [21] to calculate the textual descriptions.

Finally, we select [39] as a global explanation method because we deem it closely related to our approach for analyzing arbitrary properties. However, it is an associational approach that aims to uncover changes in behavior on the test dataset with respect to a selected property without interventions. Similar to our approach, [39] frames supervised learning as a Structural Causal Model (SCM). However, it investigates the dependence between the property and the output statistically. Crucially, the authors note that the reference annotation is often a common confounder when the label correlates with the property of interest. Hence, conditional independence (CI) is estimated with the reference annotations serving as the conditioning variable. The resulting explanation is the binary result of this hypothesis test.

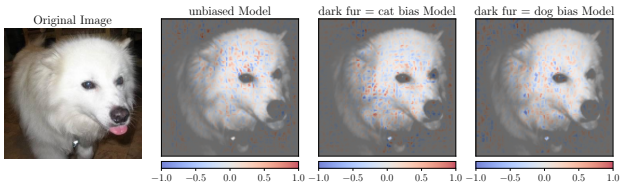
However, there is no universal non-parametric CI test [50]. Therefore, selecting a suitable CI test is the essential hyperparameter choice. We use partial correlation to cap-



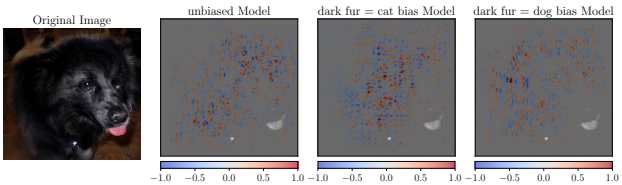
(a) Integrated gradients [53] for image of a dog.



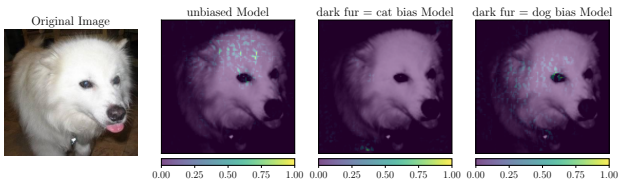
(b) Integrated gradients [53] for intervened image.



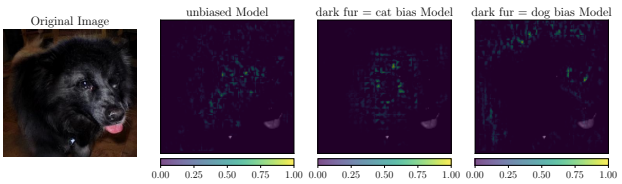
(c) DeepLIFT [52] attribution for image of a dog.



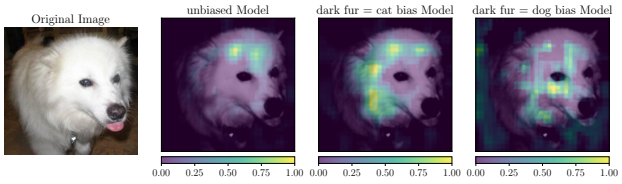
(d) DeepLIFT [52] attribution for intervened image.



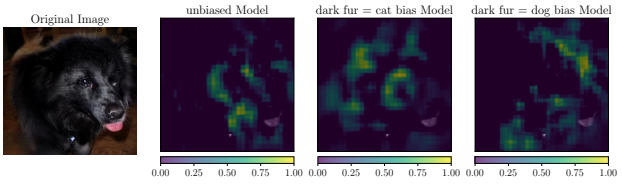
(e) Guided Grad-CAMs [49] for image of a dog.



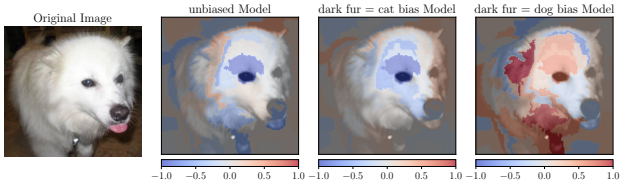
(f) Guided Grad-CAM [49] for intervened image.



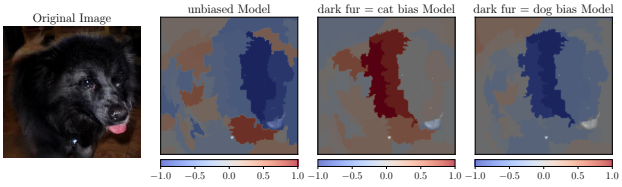
(g) Occlusion based attribution [59] for image of a dog.



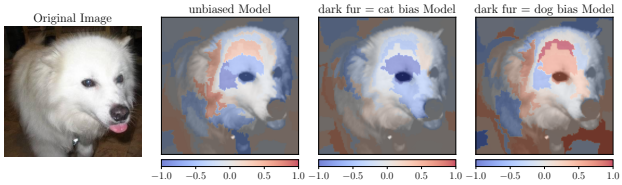
(h) Occlusion based attribution [59] for intervened image.



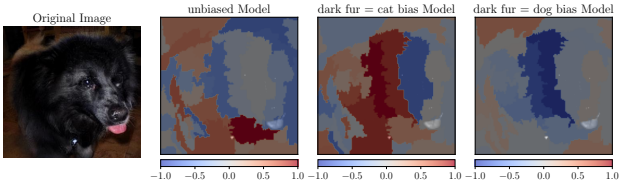
(i) LIME-based [41] attribution for the image of a dog.



(j) LIME-based [41] for intervened image.



(k) Kernel-SHAP based [29] attribution for image of a dog.



(l) Kernel-SHAP based [29] attribution for intervened image.

Figure 18. Multiple local XAI methods were applied to our dog image example. In all cases, we utilized the implementation as provided in [23]. We also exclusively visualize the results for the cat class logit, meaning positive values should correspond to regions important for the decision of the respective models.

ture linear relationships and two non-linear tests, namely CMIknn [44] and conditional HSIC [13]. For the properties in our test dataset, we select the fur color and background illumination to stay comparable to our experiments in the main paper. Specifically, we take the binary fur color as indicated by membership in our biased splits (see Section B.1). Regarding the background color, we calculate the mean of the pixel intensities over the four corner pixels for all images in our test set. Here, a low value indicates a dark background, while a high value corresponds to bright colors.

Results Fig. 19, Fig. 20, and Fig. 21 visualize the three CAVs with the highest SHAP values [51] for the `cat` class in the unbiased model, the dark cats biased model, and the dark dogs biased model, respectively. Each CAV is accompanied by the top six images with the highest similarity in the latent space.

Notably, the unbiased model exhibits relatively large positive SHAP values for all three visualized concepts (Fig. 19). In contrast, the biased models (Fig. 20 and Fig. 21) have a third-ranked CAV with a SHAP value close to zero. This suggests that two concepts are sufficient to reproduce the outputs of biased models according to [58].

However, interpreting the visual explanations semantically is a challenging task, similar to the local methods (Section B.4). For the unbiased model, we observe multiple cat images, particularly for the concept with the largest SHAP value. This may indicate that a broad cat concept is important for the model regarding the `cat` class. In contrast, the dark cats biased model (Fig. 20) does not exhibit any cat images. Instead, we observe many darker images, hinting at the underlying bias of the fur color. However, without knowledge of the training setup or additional steps, this is difficult to discover from just the visual explanations of the CAVs.

To gain a deeper understanding, we utilize [47] to find textual descriptions for the CAVs found by [58]. We list these descriptions in Table 8. Specifically, we provide the five closest words in CLIP [38] latent space and indicate the central word of the corresponding cluster [47]. These descriptions confirm our previous observations. The unbiased model and the dark dog-biased model learn a broad cat or kitten concept. In contrast, the most important CAV for the dark cats model is described as vehicles. Nevertheless, we highlight two findings. First, "darkness" is part of the five closest words in the third CAV of the dark cats biased model. Second, "white" is similarly discovered for the second CAV of the dark dogs model. These results hint again at the underlying bias.

While our local approach is not explorative, it allows us to directly test for specific properties or concepts. Hence, it provides an additional tool for investigating model behavior. Furthermore, we believe that combining our approach with explorative concept-based methods is promising, as discussed in Section 5.

Finally, [39] can detect global behavior changes with respect to a specific property similar to our approach. In Table 9, we summarize the binary results for both the fur color and background properties using different conditional independence (CI) tests. We find strong differences between the CI tests. Specifically, CMIknn [44] always rejects the null hypothesis (property is not used). In contrast, conditional HSIC [13] cannot identify the relationship for both properties. For partial correlation, we find that the biased models always change behavior for changes in both the fur color and background illumination. Taking the majority decision following [40], we overall find that all three models learn the fur color, while only the biased models additionally use the background.

While the approach described in [39] enables testing for the usage of arbitrary properties on a global level, we identify two differences to our approach. First, our approach allows a direct interpretation of how the outputs change given the local interventions. In other words, by utilizing gradual interventions, we can visualize the shift in behavior for specific variations in the property. Second, while the global insights in Table 9 tell us that background is learned, it is not clear whether it is important for individual inputs. With $\mathbb{E}[|\nabla_{\mathsf{X}}|]$, we develop a score to measure the impact of a property under interventions to provide local insights.

Overall, we find that our local approach complements global baselines to investigate how model outputs change for interventions in individual inputs.

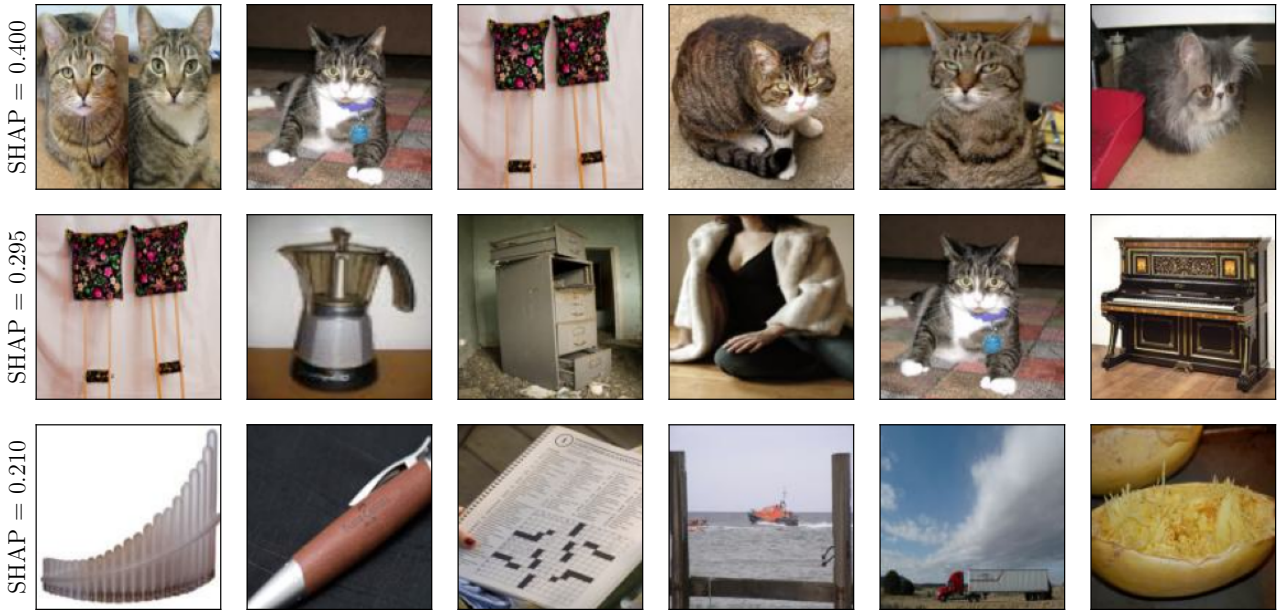


Figure 19. The top three CAVs for the unbiased model using [58]. Here we focus again on the cat class logit, to stay consistent with our other experiments.

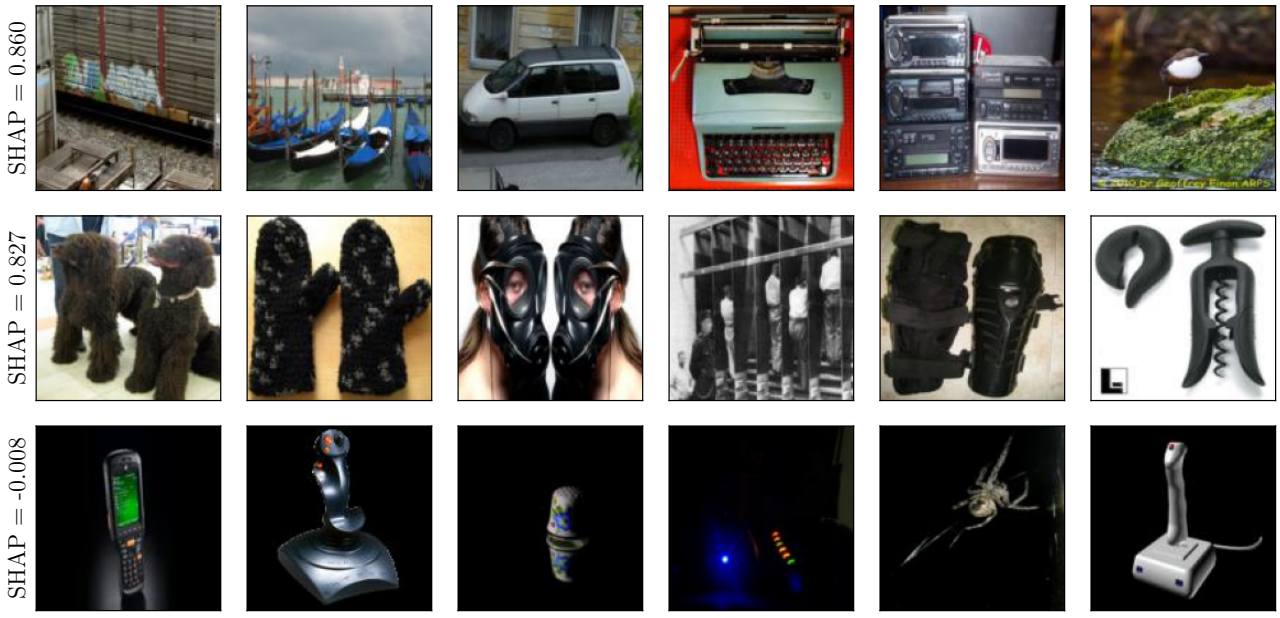


Figure 20. The top three CAVs for the dark cat biased model using [58]. Here we focus again on the cat class logit, to stay consistent with our other experiments.

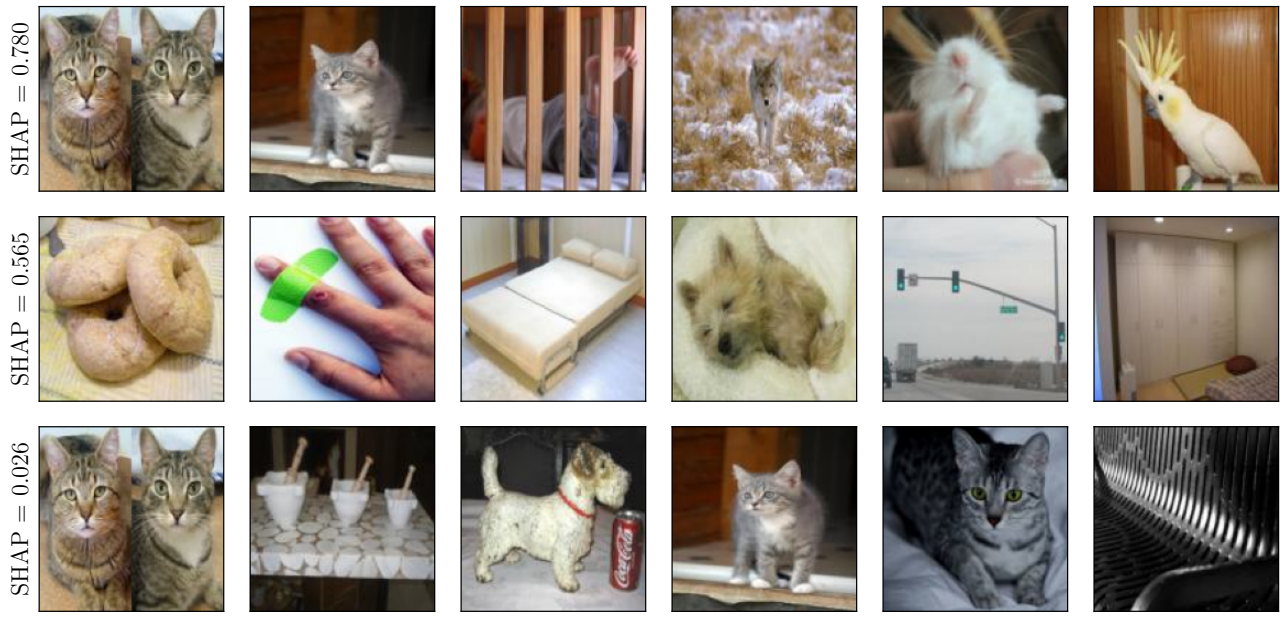


Figure 21. The top three CAVs for the dark dog biased model using [58]. Here we focus again on the cat class logit, to stay consistent with our other experiments.

Table 8. Concept descriptions for the CAVs visualized in Fig. 19, Fig. 20, and Fig. 21 generated using [47]. Here, Google20K [21] is used as the text probing dataset.

Model	SHAP	Central Word	Closest Words
Unbiased	0.400	cats	kitts, cats, katz, cat, lynx
	0.295	cabinets	cabinets, dresser, furniture, cabinet, furnishing
	0.210	products	products, shipment, casserole, product, coatings
Dark cats bias	0.860	vehicles	vehicles, vehicle, automobiles, wagon, decks
	0.827	armor	black, equipment, armor, exhibitor, overstock
	-0.008	lights	nightlife, candle, lights, flashlight, darkness
Dark dogs bias	0.780	kitten	beige, kitten, persian, cygwin, mimi
	0.565	room	room, bed, condosaver, white, resulting
	0.026	cat	cat, cats, kitten, kitts, persian

Table 9. Results for the global method described in [39] and the fur color and background color properties in the cats versus dogs test set. We use a significance level of 0.05 and provide the binary results of the hypothesis tests. Additionally, we mark significant results. Finally, the columns denote different conditional independence tests, the main hyperparameter of [39].

	Fur Color			Background Color		
Model	Partial Corr.	CMIknn [44]	cHSIC [13]	Partial Corr.	CMIknn [44]	cHSIC [13]
unbiased model	$p < 0.05$	$p < 0.05$	$p > 0.05$	$p > 0.05$	$p < 0.05$	$p > 0.05$
dark fur = cats	$p < 0.05$	$p < 0.05$	$p > 0.05$	$p < 0.05$	$p < 0.05$	$p > 0.05$
dark fur = dogs	$p < 0.05$	$p < 0.05$	$p > 0.05$	$p < 0.05$	$p < 0.05$	$p > 0.05$

C. ISIC Classification - Additional Details

C.1. Setup Details

To showcase the ability of our approach to measure systematic changes in model prediction behavior in complex settings, we select the real-world task of skin lesion classification. Specifically, we choose the binary problem to differentiate between healthy skin lesions (nevi) and dangerous melanomata. As a dataset, we sample an equal amount of both classes from the ISIC archive [1]. We train four different architectures: ResNet18 [16], EfficientNet-B0 [54], ConvNext-S [27], and ViT-B/16 [9]. For each of these datasets, we consider three training datasets: unbiased skin lesion data, biased skin lesion data, and the ImageNet [45] pre-trained weights.

Regarding the bias, consider that the ISIC archive is a collection of various skin lesion images collected by independent groups and medical researchers. Hence, there are strong variations depending on the respective data sources. This includes biases such as the spurious correlation of colorful patches with the class nevus introduced by [48]. Specifically, Scope et al. [48] study nevi in children and apply visually large and distinct colorful patches next to healthy skin lesions.

In our biased training setting, we sample half of the images of class nevus from the set of images containing colorful patches. In contrast, we exclude these images for our unbiased split. We proceed similarly for the respective test data, on which we perform our remaining investigation.

Training Hyperparameters: Again, we rely on ImageNet [45] pre-trained weights and normalization statistics. Hence, we resize the images to an input size of 224×224 during training and inference. We apply [31] with the wide augmentation space during training.

We optimize using AdamW [28], with a learning rate of 0.0001, weight decay of 0.0005, and momentum of 0.9. Finally, we train for 50 epochs with a batch size of 32. The performance of all architecture and training data combinations is contained in Table 10.

Colorful Patch Interventions: To test how strongly each of the models relies on the colorful patches to derive its prediction, we intervene in images containing melanomata. Specifically, we randomly sample ten melanoma images correctly classified by both the biased and unbiased networks. Next, we sample five random images containing colorful patches per melanoma image. We ensure that these patches are neither part of the training nor the test datasets. Finally, we use segmentations of the colorful patches provided in [42] to alpha blend them with the melanoma images. In our main paper, we include one example in Fig. 5 (bottom).

Table 10. Accuracy in percent (%) of various architectures for melanoma classification trained on data from [1]. We separate the test data into biased data containing colorful patches from [48] and unbiased data where we sample nevus images without. Similarly, the first rotated column indicates the training distribution.

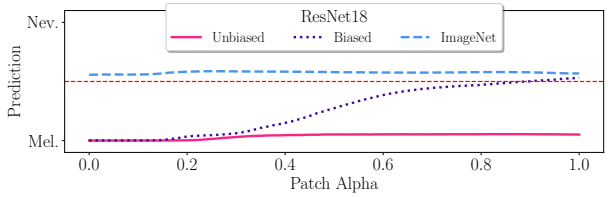
		Test Data	
	Model	Unbiased	Biased
Unbiased	ResNet18 [16]	86.70	85.89
	EfficientNet-B0 [54]	87.15	90.12
	ConvNeXt-S [27]	88.05	88.32
	ViT-B/16 [9]	86.25	84.37
Biased	ResNet18 [16]	82.84	89.13
	EfficientNet-B0 [54]	85.09	91.37
	ConvNeXt-S [27]	83.92	92.18
	ViT-B/16 [9]	83.47	87.42

We choose synthetic interventions to showcase the ability of our approach to work with diverse sources of interventional data. This especially holds for expert and domain knowledge, where specifically designed interventions can ensure the correct target, similar to this experiment or [5]. Hence, our approach facilitates the analysis of complex tasks where it is important to strictly apply the causal hierarchy theorem [32].

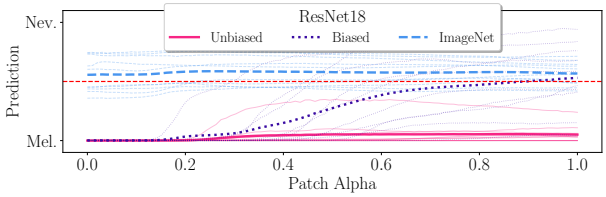
C.2. Additional Results

Table 10 contains the accuracies for the different training and test splits (biased/unbiased). There is a clear difference between models trained on the unbiased split and models trained in a biased scenario. Each of the two paradigms outperforms the other in the test split following their respective training distribution. However, the models trained on unbiased skin lesions also achieve similarly high performance on the data containing colorful patches. In contrast, the biased models seem to overfit the training domain and perform drastically worse on the unbiased split. Here, the ViT [9] achieves the lowest performance on the biased data, which is even outperformed by some of the models trained on the unbiased split. Nevertheless, the biased ViT [9] loses around 4% when evaluated on unbiased data, which is the lowest performance drop of all biased models.

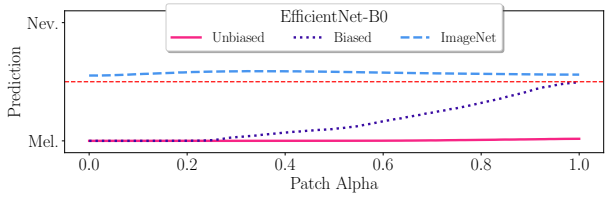
This observation is congruent with the estimated $\mathbb{E}[|\nabla_X|]$ in Table 3 in our main paper. Specifically, we measure the lowest expected property gradient magnitude of all biased models for the ViT [9]. All convolutional models show, on average, a higher impact during the colorful patch interventions. Fig. 22 visualizes the average for the ten melanoma images with patch interventions and confirms our observation. The vision transformer is the only architecture where the biased model does not, on average, change



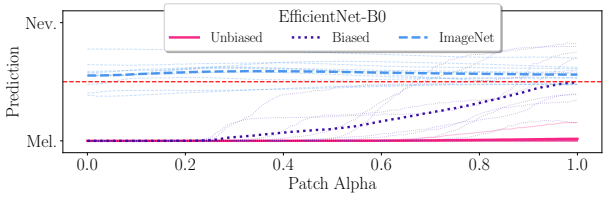
(a) Resnet18 [16]: Average of ten colorful patch interventions.



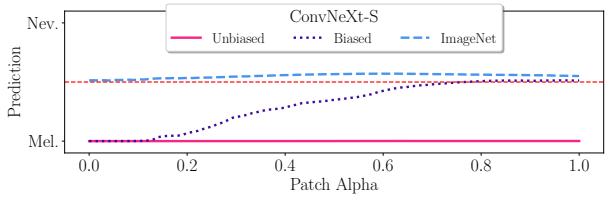
(b) Resnet18 [16]: Ten local colorful patch interventions.



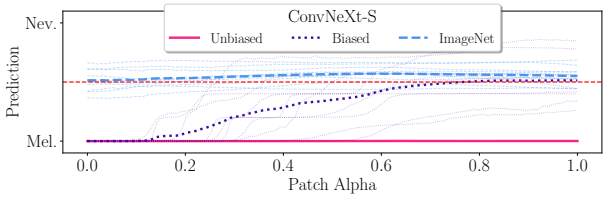
(c) EfficientNet-B0 [54]: Average of ten colorful patch interventions.



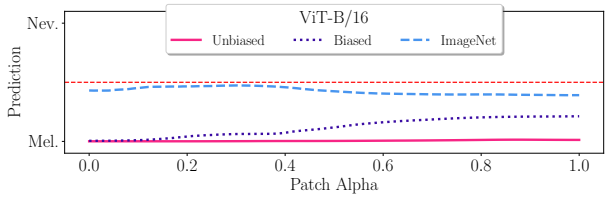
(d) EfficientNet-B0 [54]: Ten local colorful patch interventions.



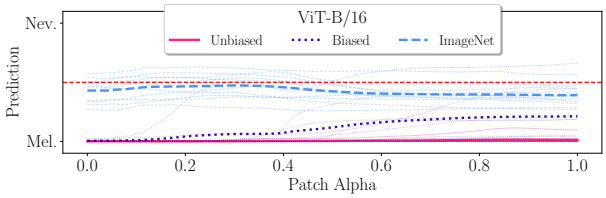
(e) ConvNext-S [27]: Average of ten colorful patch interventions.



(f) ConvNext-S [27]: Ten local colorful patch interventions.



(g) ViT-B/16 [9]: Average of ten colorful patch interventions.



(h) ViT-B/16 [9]: Ten local colorful patch interventions.

Figure 22. Changes in model predictions of skin lesion classifiers for the intervention of introducing colorful patches [48]. We include various architectures and split depending on the training data: unbiased skin lesion images, biased skin lesion images where spurious patches correlate with the class `nevus`, and ImageNet pre-trained weights. The red dotted line is the threshold where the predictions flip.

its prediction. This result is an indication that convolutional models learn this spurious visual bias more strongly. Previous works, e.g., [36, 40], analyze this behavior on an associational level and find significant changes in prediction behavior for colorful patches. Now, our proposed approach of interventional local explanation enables a more fine-grained analysis.

Next, both in Fig. 22 and Table 3, we see higher colorful patch impact for the ImageNet [45] pre-trained models compared to the models trained on unbiased skin lesions. This is an expected observation, given the large visual changes during the intervention. Further, the differentiation of large blocks of colors is a useful feature during the pre-training on general-purpose datasets. In contrast, the models trained on unbiased skin lesions, which are often centered, learn to focus on the actual lesion. Hence, they are only minimally influenced by the colorful patches. This result is not obvious from a pure performance analysis (Table 10). However, our interventional approach does provide insights into models trained for this complex scenario beyond benchmarking (Table 3).

Table 11. Final accuracies in percent (%) achieved by various models trained to differentiate young versus old in CelebA [25]. We split between ImageNet [45] pre-training ("PT") and random initialization ("RI") and calculate the performance delta (Δ).

Model	PT	RI	Perf. Δ
ConvMixer [55]	86.08	81.49	-4.59
ResNet18 [16]	85.37	84.38	-0.99
EfficientNet-B0 [54]	86.61	84.14	-2.46
MobileNetV3-L [19]	85.94	83.05	-2.88
DenseNet121 [20]	85.75	84.20	-1.55
ConvNeXt-S [27]	85.98	83.63	-2.35
ViT-B/16 [9]	85.51	72.49	-13.02
SwinT-S [26]	85.72	50.25	-35.47

D. CelebA - Additional Details

In this section, we provide more details regarding our second experiment, where we study the training dynamics of eight architectures (Section 4.3). First, Section D.1 discusses the corresponding hyperparameters and training details. Then, we provide additional visualizations in Section D.2 before finally investigating another local example.

D.1. Setup Details

In our second experiment, we investigate the local training dynamics of eight different architectures: ConvMixer [55], ResNet18 [16], EfficientNet-B0 [54], MobileNetV3-L [19], DenseNet121 [20], ConvNeXt-S [27], ViT-B/16 [9], and SwinTransformer-S [26]. We select these architectures to cover a range of model families and design choices and investigate random initialization versus pre-trained weights.

For the ConvMixer model, we use an initial patch size of 14, a depth of 20, kernels with a width of 9, and a latent representation size of 1024. These hyperparameters are specifically chosen to utilize the ImageNet pre-trained weights included in [57]. For specifics regarding these parameters, we refer the reader to the original paper [55]. For all other architectures, we rely on the standard PyTorch [33] implementation and parameterizations.

During training and inference, we resize the images to an input size of 224×224. For pre-trained models, we use the ImageNet [45] statistics for normalization. In contrast, for random initializations, we normalize the values in the interval of $[-1, 1]$. Additionally, we utilize [31] with the wide augmentation space during training, irrespective of the initialization. We optimize the models using AdamW [28], setting the learning rate to 0.0001, weight decay to 0.0005, and momentum to 0.9. For all models, we employ a batch size of 32.

After each of the 100 training epochs, we save the model weights, with the final weights achieving the performances disclosed in Table 11. Note that the attention-based models show a stronger decrease in performance for randomly initialized weights. Especially, the SwinTransformer-S [26] diverges to random guessing capabilities. This observation confirms other works, e.g., [9, 36], which find that transformer architectures depend heavily on pre-training. The divergence is also visible in our property analysis (see Fig. 25 and Table 12), which we will discuss in the next section.

D.2. Additional Results

We provide the visualizations of the interventional data corresponding to the results in the main paper in Fig. 23 (top row). Specifically, we again use [12] with a CFG image scale of 2.5 and increase the corresponding text scale starting from 1.05 up to 9.75. Here, we find that higher scales result in unwanted artifacts beyond the targeted intervention. As an editing phrase, we employ "change the hair to gray-white color". In this section, we provide the corresponding analysis for the other model architectures omitted in the main text and concrete measurements for the average $\mathbb{E}[|\nabla_{\mathsf{X}}|]$.

In Fig. 24 and Fig. 25, we present the development of the $\mathbb{E}[|\nabla_{\mathsf{X}}|]$ during training for the local hair color intervention across all eight architectures in our analysis. We split the visualizations between ImageNet [45] pre-trained (Fig. 24) and randomly initialized weights (Fig. 25). These results strongly support our previous observations.

Notably, there is a stark difference between the pre-trained models and the randomly initialized versions. The former strongly learn the hair color property and change their behavior based on the intervention, whereas the latter show mostly lower $\mathbb{E}[|\nabla_{\mathsf{X}}|]$ scores. We highlight the different scales for the respective y-axes. Additionally, we draw attention to the results for the SwinTransformer-S [26]. While the pre-trained model often strongly learns the hair color property, the randomly initialized variant diverges completely (Table 11). This observation is further corroborated by the average $\mathbb{E}[|\nabla_{\mathsf{X}}|]$ in Table 12.

We also calculate the corresponding Pearson correlation coefficients [35], which we include in Table 12. Note that in all cases where we can calculate the correlation, we measure lower effect strength for the randomly initialized models. This provides further evidence of the effectiveness of our approach in capturing the strength of the behavior changes under gradual interventions.

To gain further insight into the learned behavior, we provide visualizations similar to our behavior plots (e.g., Fig. 1), where we show the output changes under the interventions after every epoch during training. In Fig. 26 and Fig. 27, we show the pre-trained and randomly initialized models, respectively.

We highlight two key observations. First, for both model variants, we observe that the selected example is nearly always correctly classified. For example, as noted in our main

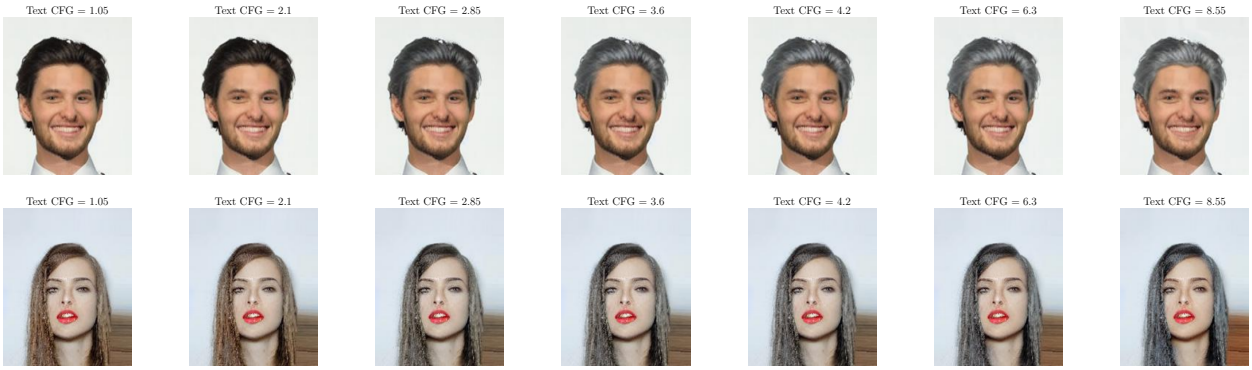


Figure 23. Hair color interventions for two people labeled as young. Here, we use a pre-trained version of [12]. Increasing the CFG text scale indicates a higher alignment with the edit instruction, here "change the hair to gray-white color". The top row is extensively discussed in the main section of the paper. We include additional results about the bottom row in the supplementary material.

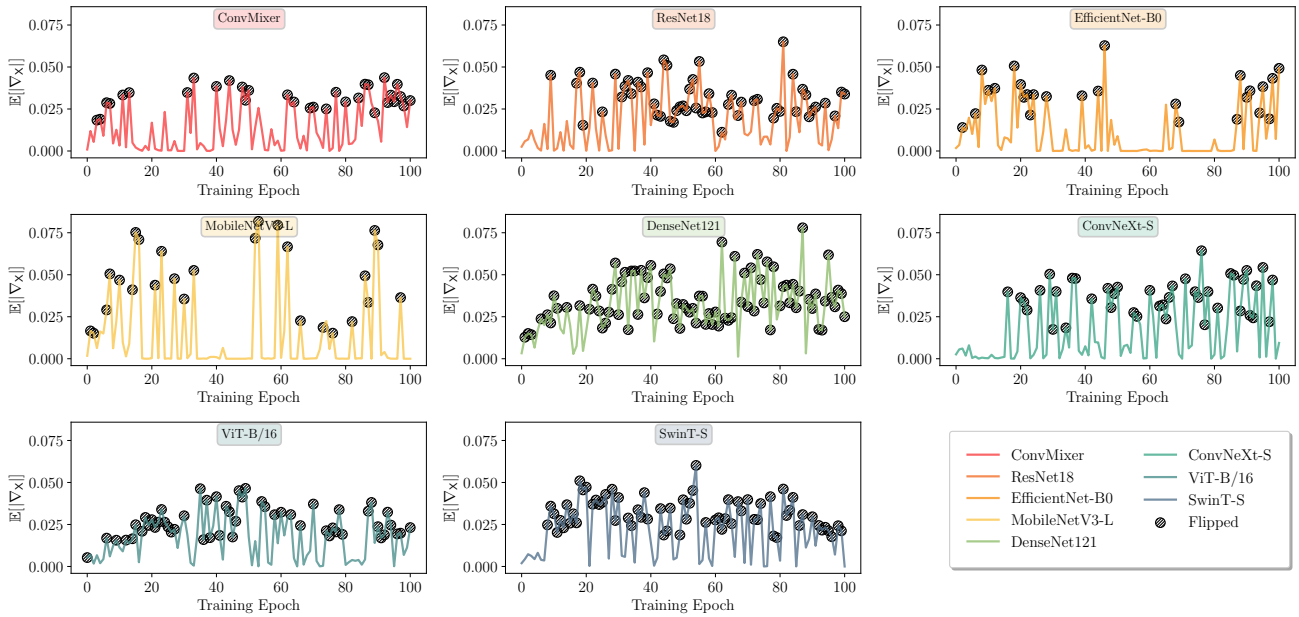


Figure 24. Visualization of how the impact of gray hair changes over the training for all models in Table 11, and top row in Fig. 23. Here, we focus on ImageNet pre-trained [45] weights, marking epochs where the network prediction flips during the intervention.

paper, the DenseNets [20] both classify the original image correctly in all cases but differ in their behavior under the intervention. An exception is the randomly initialized SwinTransformer-S [26], which achieves random guessing accuracy (Table 11) and shows outputs nearly independent of the inputs for the complete training.

Second, given that most models correctly classify the original image, we note that changes in behavior often lead to incorrect predictions during the intervention. Specifically, for grayer hair colors (according to Fig. 23, top row), we observe lower activations in the `Young` logits of our classifiers. In fact, we often find a rapid decline after a certain state in the gradual intervention. However, the specific threshold varies depending on the epoch. This behavior is, for example, visible for the ConvMixer [55] in Fig. 27.

In general, we can confirm the lower average $\mathbb{E}[|\nabla_X|]$ for the randomly initialized models in Table 12 using Fig. 27. Many models only show slight deviations under the hair color interventions. This again highlights the difference between effect size and significance of our $\mathbb{E}[|\nabla_X|]$ scores. Further, these results show that pre-trained and randomly initialized models differ on the level of properties they employ for decisions on a local level. Additionally, this change in local behavior is not directly connected to the classifica-

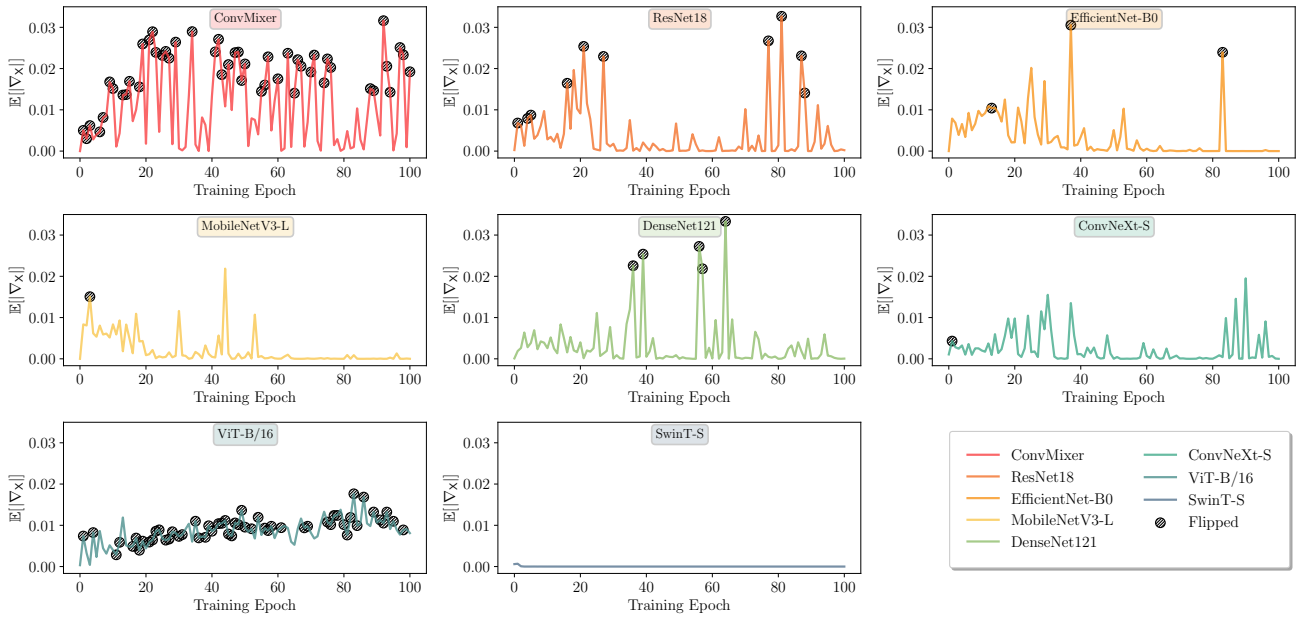


Figure 25. Visualization of how the impact of gray hair changes over the training for all models in Table 11, and top row in Fig. 23. Here, we focus on randomly initialized weights, mark epochs, where the network prediction flips during the intervention.

Table 12. Average $\mathbb{E}[|\nabla_X|]$ for the gray hair color feature over the training process. We list both ImageNet [45] pre-trained and randomly initialized models. In **bold**, we highlight a greater measured $\mathbb{E}[|\nabla_X|]$ for any architecture. Similarly, we use underline for more flipped predictions. We also denote the number of significant p-values ($p < 0.01$) over the training following Algorithm 1. Note that the maximum is 101 because we test the initial model and one model after each of the 100 epochs. Lastly, we determine the average Pearson correlation coefficient [35] (ρ) over the training. However, ρ is not defined for constants, leading to missing values for some models.

	Pre-trained				Random Init			
Model	$\mathbb{E}[\lvert\nabla_X\rvert]$	#Flips	#Sig.	ρ	$\mathbb{E}[\lvert\nabla_X\rvert]$	#Flips	#Sig.	ρ
ConvMixer [55]	**0.01453**	32	101	-0.73037	0.01144	49	100	—
ResNet18 [16]	**0.02006**	55	101	-0.71550	0.00376	10	101	-0.55702
EfficientNet-B0 [54]	**0.01153**	26	95	-0.63042	0.00336	3	98	—
MobileNetV3-L [19]	**0.01379**	26	78	-0.32585	0.00208	1	95	—
DenseNet121 [20]	**0.03194**	86	101	-0.81263	0.00329	5	101	-0.43834
ConvNeXt-S [27]	**0.01801**	42	101	-0.67621	0.00244	1	101	-0.16601
ViT-B/16 [9]	**0.01780**	54	101	-0.74494	0.00853	55	101	-0.75000
SwinT-S [26]	**0.02268**	67	101	-0.78029	0.00001	0	19	—

tion of the original sample.

Although the relationship between hair color and age is not causal, with gray hair not necessarily indicating older age, we would expect well-performing classifiers to capture the statistical correlation present in the CelebA dataset [25]. Indeed, the results in Table 11 show that pre-trained models achieve higher predictive performance, likely due to their ability to exploit such correlations. Our local analysis supports this finding, revealing a stronger dependence on the hair color property for pre-trained models. However, it is crucial to note that these results may not generalize to other local inputs, as other factors can influence the model's behavior and lead to different observations. To illustrate this, we provide an additional example and re-examine the local model outputs, highlighting again how global correlations can be misleading when analyzing local model behavior.

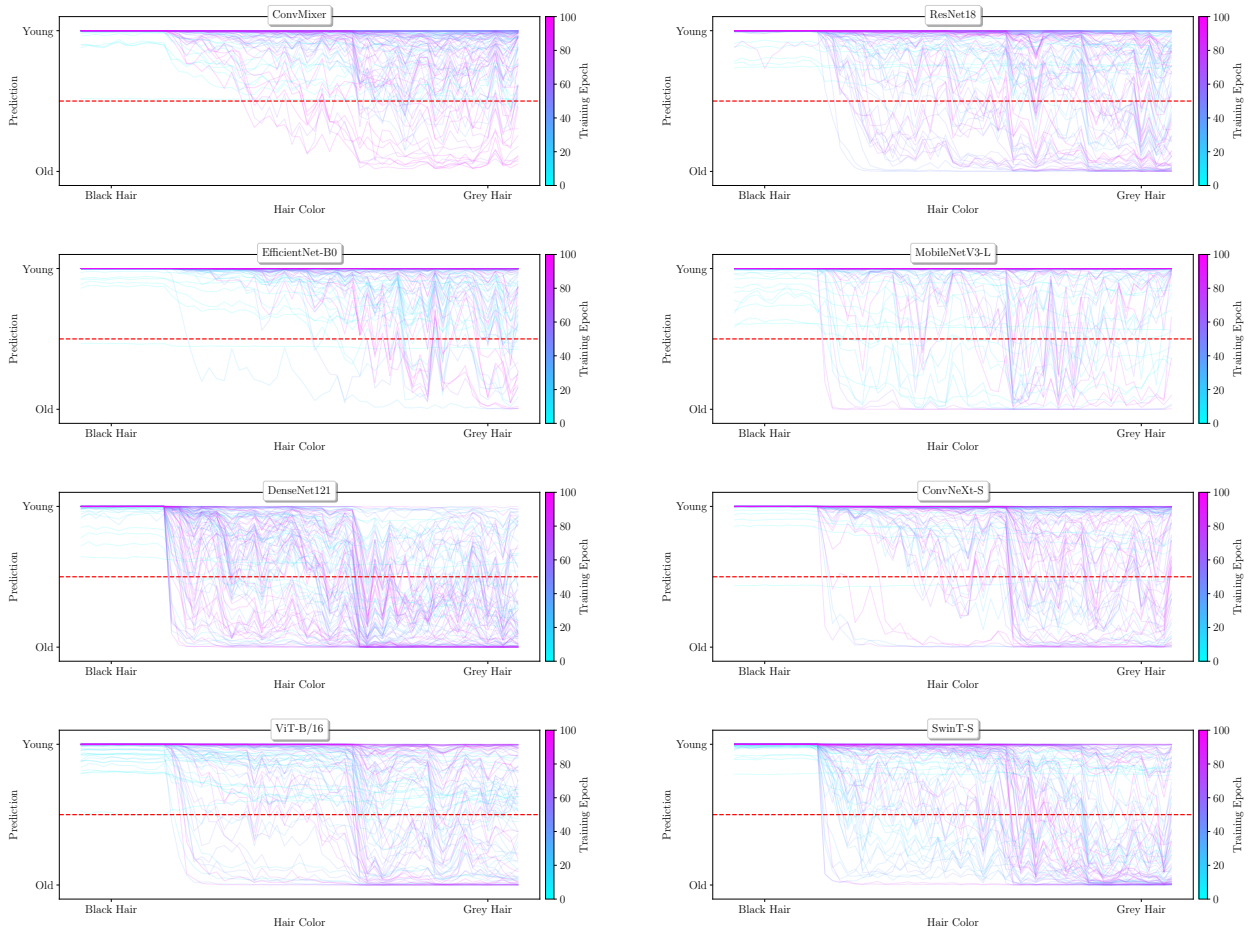


Figure 26. Model behavior per epoch visualized for the hair color intervention shown in the top row of Fig. 23 and discussed in the main part of the paper. Here, we only show the behavior for **pre-trained models** using ImageNet [45] weight. The models include ConvMixer [55], ResNet18 [16], EfficientNet-B0 [54], MobileNetV3-L [19], DenseNet121 [20], ConvNeXt-S [27], ViT-B/16 [9], and SwinTransformer-S [26]. The red dotted line indicates, in all cases, the threshold where the model prediction flips.

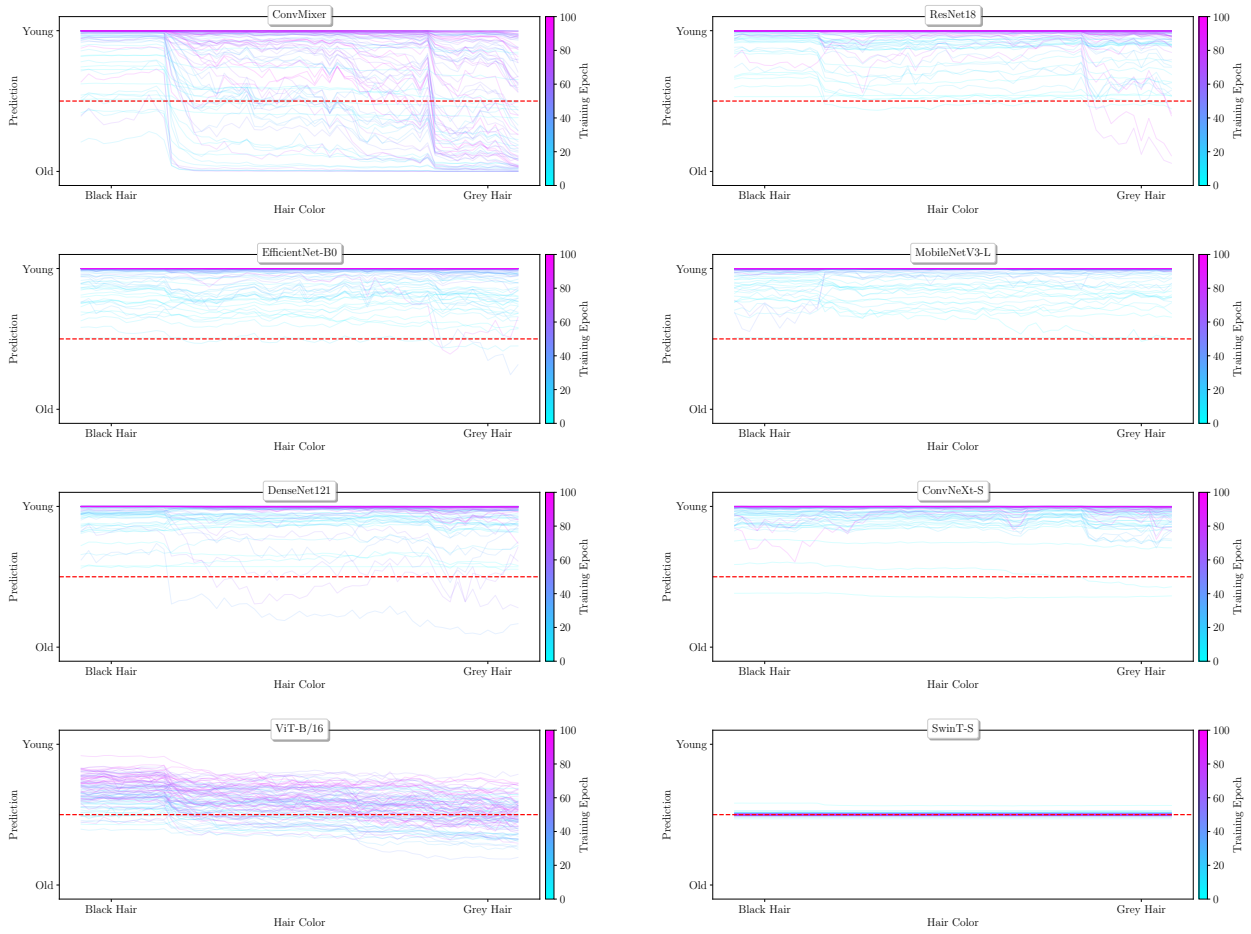


Figure 27. Model behavior per epoch visualized for the hair color intervention shown in the top row of Fig. 23 and discussed in the main part of the paper. Here, we only show the behavior for **randomly initialized** models. The models include ConvMixer [55], ResNet18 [16], EfficientNet-B0 [54], MobileNetV3-L [19], DenseNet121 [20], ConvNeXt-S [27], ViT-B/16 [9], and SwinTransformer-S [26]. The red dotted line indicates, in all cases, the threshold where the model prediction flips.

Additional Sample with Confounding Properties In Fig. 23, we present a second example used for local training analysis (bottom row), where we employ the same prompt and hyperparameters as before. Notably, many properties, such as makeup, perceived gender, and hair length, differ between the two individual samples. We visualize the development of the local $\mathbb{E}[|\nabla_{\mathsf{X}}|]$ for the hair color intervention over training in Fig. 28 and Fig. 29 for the pre-trained and randomly initialized models, respectively. The corresponding changes in behavior are showcased in Fig. 30 and Fig. 31. In all cases, we observe very small $\mathbb{E}[|\nabla_{\mathsf{X}}|]$ scores, indicating that the models are not locally influenced by the gray hair color for this individual sample. Although minor exceptions exist, such as the pre-trained EfficientNet early in the training or the randomly initialized ViT [9], we find only minimal differences between the initializations. The visualizations in Fig. 30 and Fig. 31 provide an explanation, showing that the models, with rare exceptions, correctly classify the sample throughout the complete intervention and training.

We hypothesize that other properties correlated with age may lead to this phenomenon. For instance, makeup is strongly correlated with the `Young` label in CelebA [25]. The visible makeup in Fig. 23 (bottom row) could potentially be more influential for this specific input. Future work should investigate the local interactions of properties to further interpret local prediction behavior.

Our observations underscore the importance of local interventional explanations, which can provide additional insights for individual inputs that go beyond global insights.

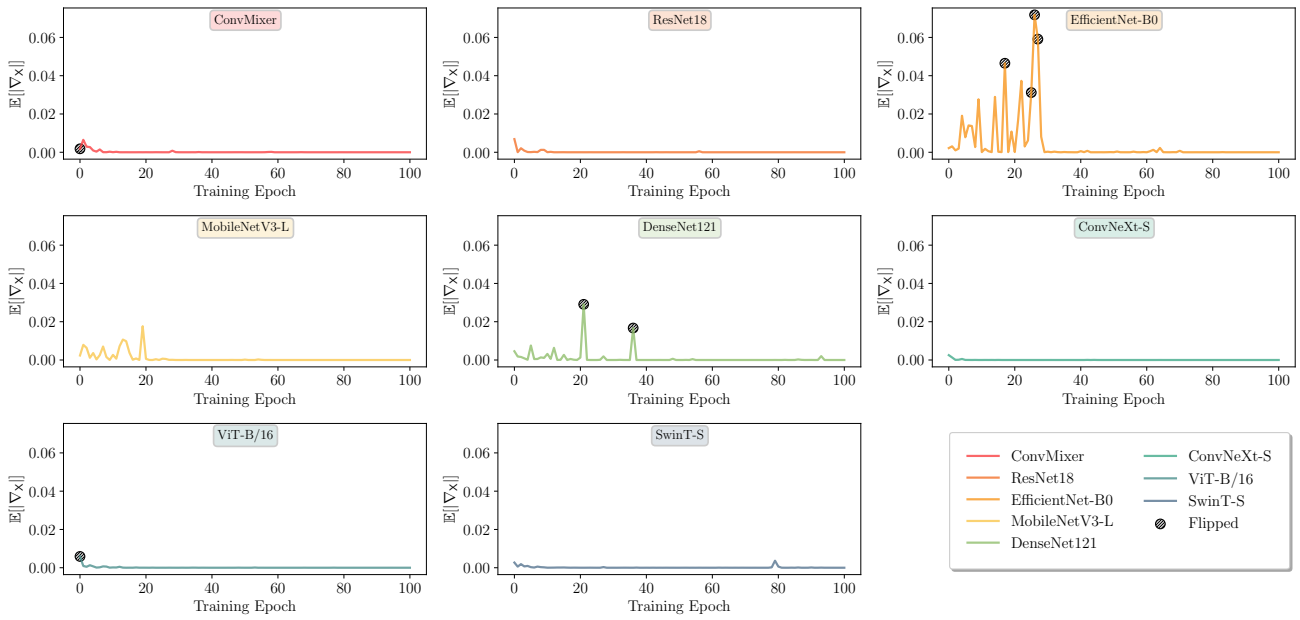


Figure 28. Visualization of how the impact of gray hair changes over the training for various pre-trained architectures. Here, we utilize the bottom row of Fig. 23 and find little impact of the hair color, hinting at confounding properties.

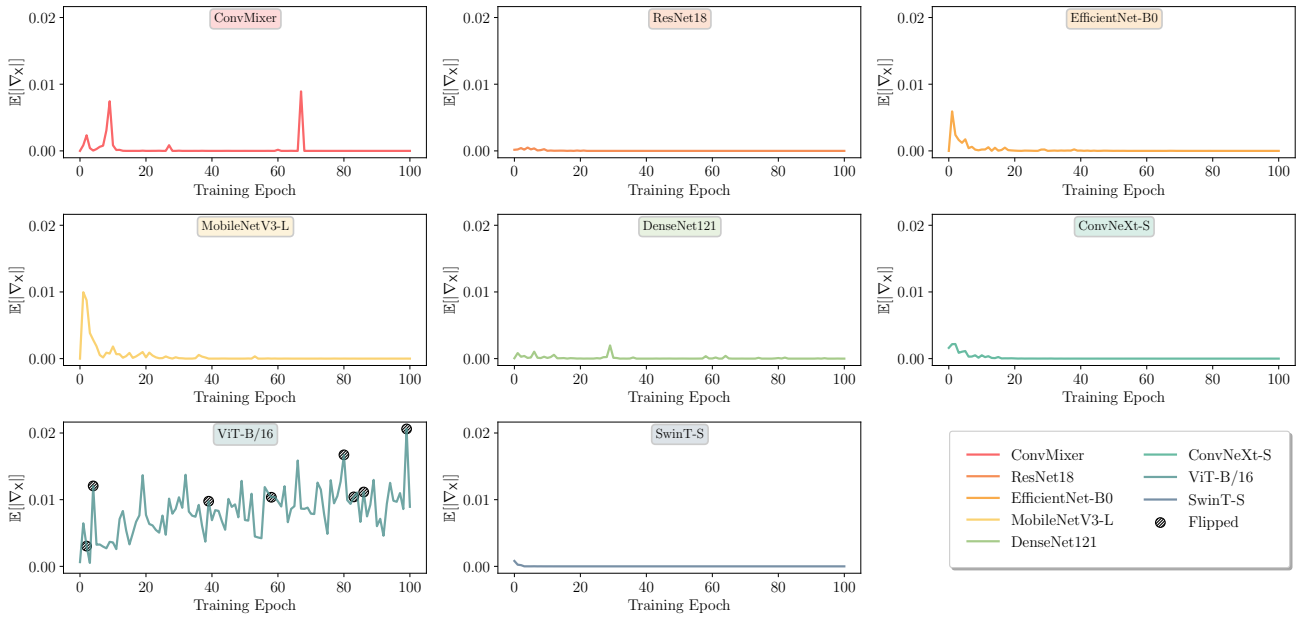


Figure 29. Visualization of how the impact of gray hair changes over the training for various randomly initialized architectures. Here, we utilize the bottom row of Fig. 23 and find little impact of the hair color, hinting at confounding properties.

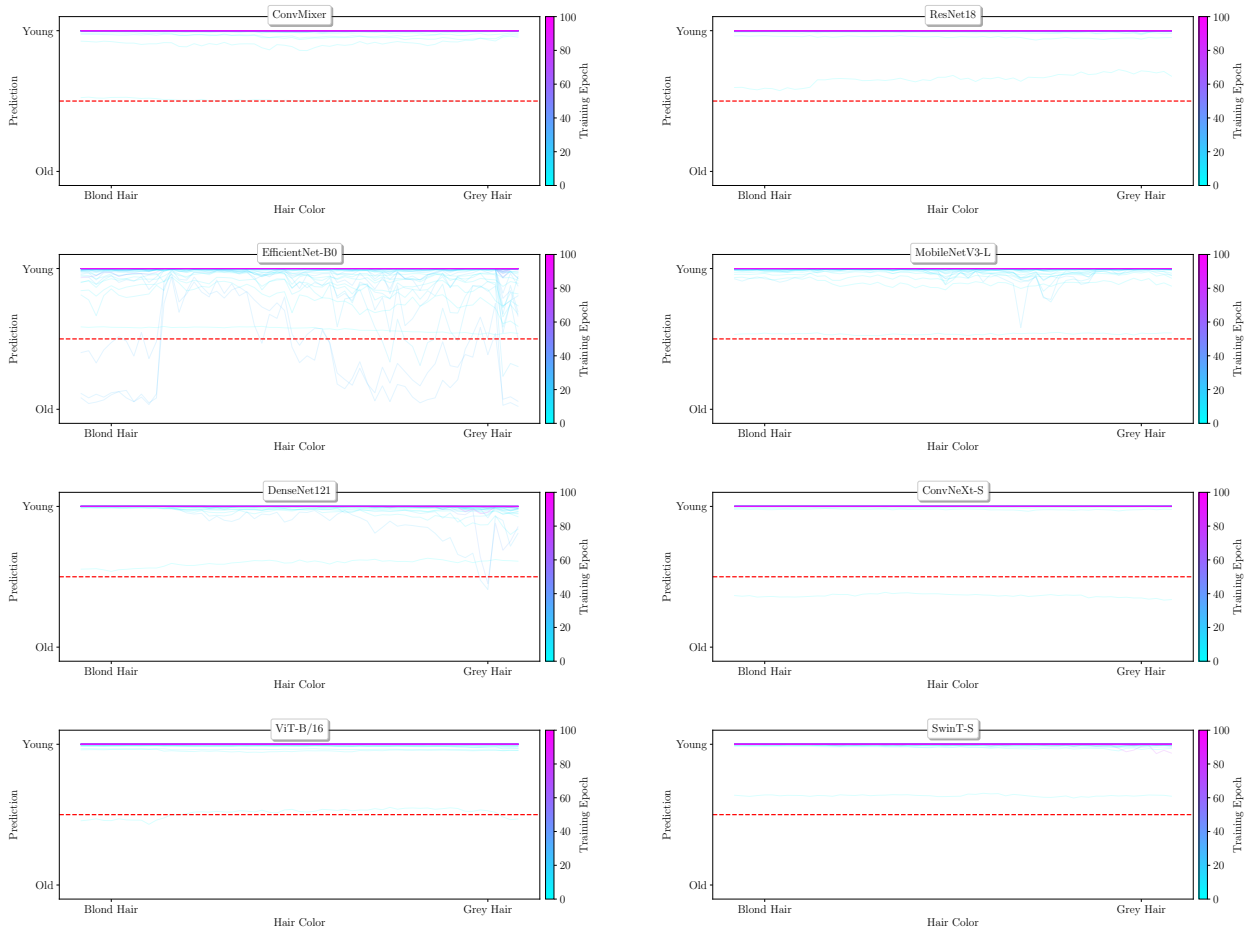


Figure 30. Model behavior per epoch visualized for the hair color intervention shown in the bottom row of Fig. 23. Here, we only show the behavior for **pre-trained models**. The red dotted line indicates, in all cases, the threshold where the model prediction flips. Note the near-constant model outputs, which are also reflected in the low impacts of Fig. 28, meaning the networks do not change behavior for hair color interventions for the selected example.

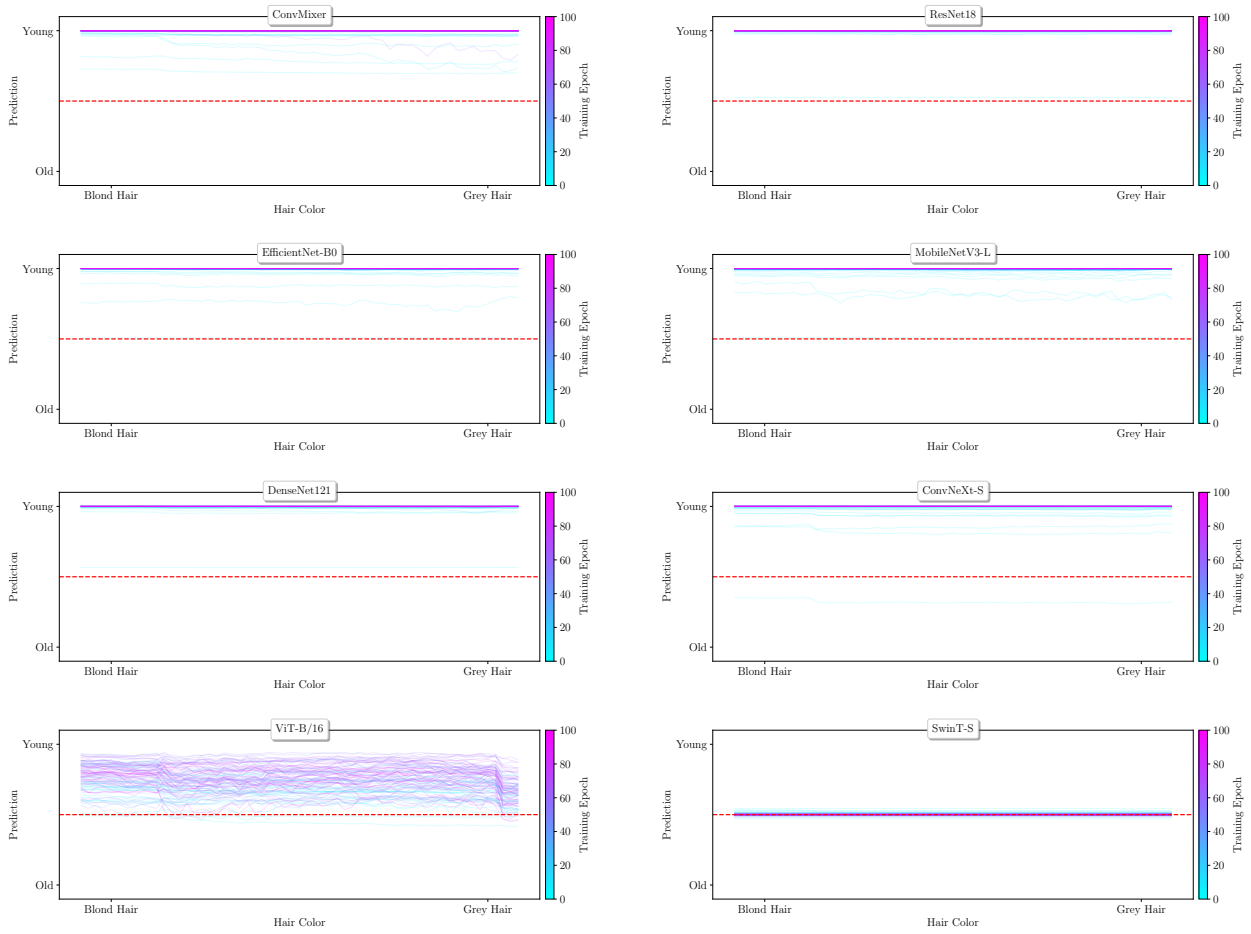


Figure 31. Model behavior per epoch visualized for the hair color intervention shown in the bottom row of Fig. 23. Here, we only show the behavior for **randomly initialized** models. The red dotted line indicates, in all cases, the threshold where the model prediction flips. Note the near-constant model outputs, which are also reflected in the low impacts of Fig. 29, meaning the networks do not change behavior for hair color interventions for the selected example.

E. CLIP Analysis - Additional Details

In this section, we structure the content as follows. First, we provide the full details of our inference setup for zero-shot classification using a pre-trained CLIP [38] model. Then, we highlight additional results and ablations.

Table 13. Recently, researchers utilize multiple text descriptions when performing zero-shot classification with CLIP [38] models, see, for example, [37, 43]. Here, we list the text descriptors used for the corresponding objects in our real-life and virtual interventional data to calculate the cosine similarities.

Obj.	Text Descriptors
Toy Elephant	`toy elephant, elephant, african elephant, picture of an elephant, gray elephant, standing elephant, elephant model, small elephant, indian elephant, elephant tusk`
Toy Giraffe	`toy giraffe, giraffe, african giraffe, picture of a giraffe, spotted giraffe, standing giraffe, giraffe model, small giraffe,tall giraffe, giraffe bull`
Toy Stegosaurus	`toy stegosaurus, stegosaurus, dinosaur, picture of a stegosaurus, toy dinosaur, standing stegosaurus, stegosaurus model, small stegosaurus, green stegosaurus, stegosaurus plates`
3D Frog Model	`frog model, toy frog, frog, frog rendering, picture of a frog, 3d frog model, green frog, photo of a large frog, sitting frog, still frog`

E.1. Setup Details

As mentioned in our main paper, we focus on real-life interventional data. Additionally, we separately perform a virtual rotation of a 3D frog model[2] around all three axes, rendering one image per degree to also exclude photon noise. This setup enables us to systematically evaluate the local impact of orientation on the CLIP [38] model's outputs.

Specifically, we construct a zero-shot classification scenario by capturing images of three toy figures: an elephant, a giraffe, and a stegosaurus. For real-life interventions, these objects do not change. Therefore, the ground truth remains fixed. Nevertheless, we intervene in the input orientation using a turn table and capture one complete rotation of the toy figures in front of a neutral background. We do this specifically because we expect animals to be most often photographed upright and facing the camera. In other words, we expect pre-trained models to show behavioral changes for uncommon object positions. We visualize parts of the data in Fig. 32

[2] https://skfb.ly/6USP7

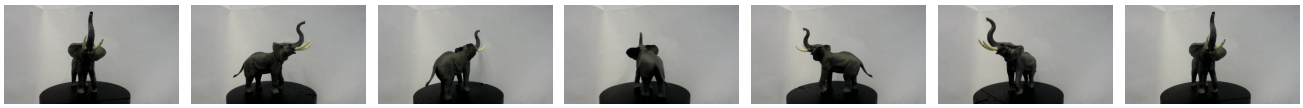
Table 14. The approximated $\mathbb{E}[|\nabla_x|]$ scores for the selected CLIP [38] model using the three real-life rotation interventions (see Fig. 32). The columns signify the text embeddings we use to calculate the cosine similarities in the latent space. To be specific, row one contains the scores pertaining to Fig. 36a, and rows two and three correspond to Fig. 36b and Fig. 8, respectively. Note that all scores are statistically significant following Algorithm 1.

	Texts Similarity $\mathbb{E}[\|\nabla_x\|]$		
Interv. Data	Elephant	Giraffe	Stegosaurus
Elephant	0.00141	0.00141	0.00178
Giraffe	0.00096	0.00086	0.00118
Stegosaurus	0.00157	0.00142	0.00140

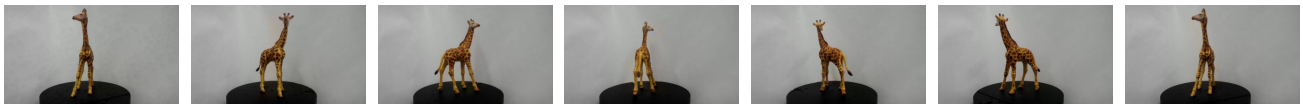
The rotation of the turn table is an in-plane rotation around the z-axis, where the figurines do not flip upside down. Hence, we strengthen our analysis by additionally incorporating virtual interventions of a 3D animal model. Specifically, we choose a model of a frog and perform interventions by rotating it around all three axes, rendering one image per degree. We visualize the resulting data in Fig. 33.

To demonstrate the cyclic nature of both the real-life and virtual interventional data, we utilize t-SNE [56] to display the CLIP [38] model latent vectors as generated by the visual encoder. Specifically, we perform a dimensionality reduction to 2D using t-SNE and show the results in Fig. 34 and Fig. 35, respectively. We also include a corresponding text phrase to provide an intuition of the latent space for our cosine similarity analysis. While these visualizations are inherently limited in the insights they provide due to the strong reduction in dimensionality, the periodicity of the data is visible, particularly for the virtual interventions. This observation is consistent with the behavior of the cosine similarities in our main paper (see Section 4.4).

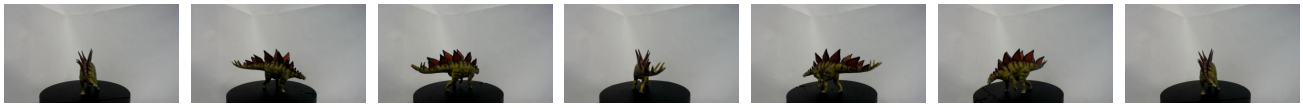
For the actual zero-shot classification, we follow recent approaches, e.g., [37, 43], and compare them to multiple text phrases or, rather, the corresponding embeddings. The specific text descriptors are listed in Table 13.



(a) Toy elephant.



(b) Toy giraffe.



(c) Toy stegosaurus.

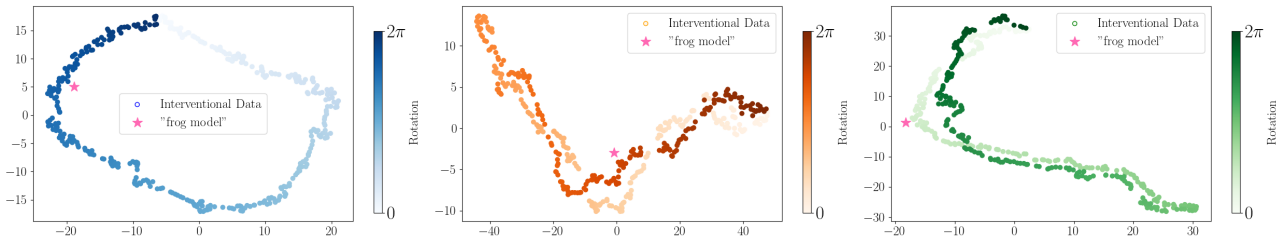
Figure 32. Real-life interventions on the position of toy animals. The interventions here use a turn table, meaning we intervene in the rotational position compared to the fixed camera.

(a) Rotation around the x-axis.

(b) Rotation around the y-axis.

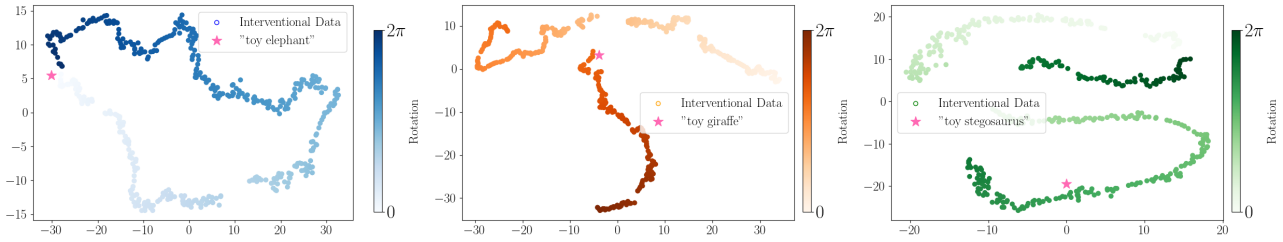
(c) Rotation around the z-axis.

Figure 33. Virtual interventions on the position of a rendered frog model. Here, we intervene by rotating around all three axes of orientation while keeping the camera fixed.



(a) Frog model, rotation around the x-axis. (b) Frog model, rotation around the y-axis. (c) Frog model, rotation around the z-axis.

Figure 34. CLIP [38] latents visualized in 2D using t-SNE [56]. In all cases, we add the embedding for the corresponding description. We encode the rotation angle using color in all three visualizations.



(a) Toy elephant. (b) Toy giraffe. (c) Toy stegosaurus.

Figure 35. CLIP [38] latents visualized in 2D using t-SNE [56]. In all cases, we add the embedding for the corresponding description. We encode the rotation angle using color in all three visualizations.

Table 15. The approximated $\mathbb{E}[|\nabla_X|]$ scores for the selected CLIP [38] model using the three virtual rotation interventions (see Fig. 33). The rows contain interventional data pertaining to the three axes of rotation. Specifically, they correspond to the scores achieved by the visualized means in Fig. 39. Note that all scores are statistically significant following Algorithm 1.

Interv. Data	Frog Texts Similarity $\mathbb{E}[\lvert\nabla_X\rvert]$
x-axis	0.00161
y-axis	0.00156
z-axis	0.00095

E.2. Additional Results

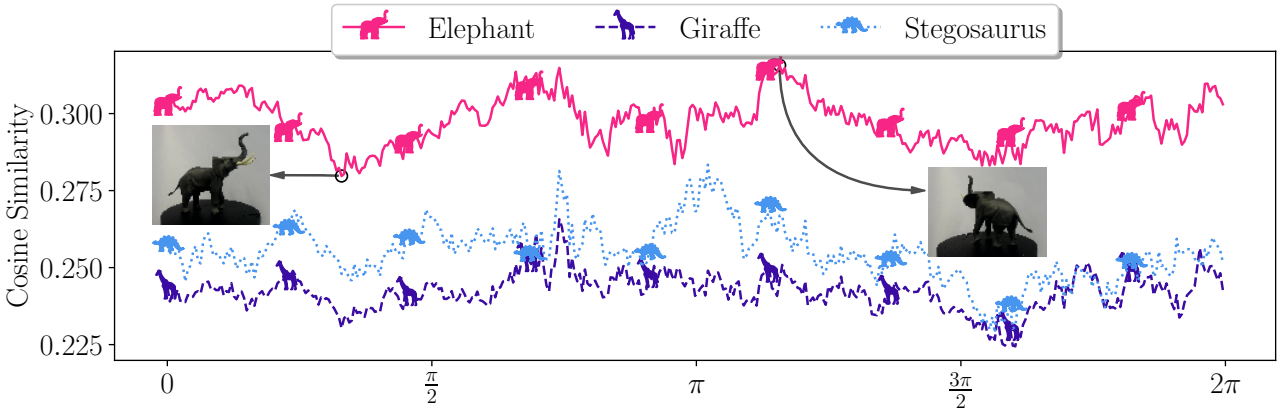
Real-Life Interventions We present additional visualizations that complement the results shown in Fig. 8, focusing on the other interventional data. Specifically, Fig. 36a and Fig. 36b display the average changes in cosine similarities for the elephant and giraffe figures, respectively, along with the minimum and maximum cosine similarity for the ground truth class.

Furthermore, we provide standard deviation estimates for the interventional data in Fig. 37. The visualization is split according to the three real-life interventions depicted in Fig. 32. In Fig. 37a, Fig. 37b, and Fig. 37c, we compare the visual embeddings to the three sets of phrases listed in Table 13. Particularly, Fig. 37c corresponds to Fig. 8 in the main paper, while Fig. 37a and Fig. 37b correspond to the visualizations in Fig. 36a and Fig. 36b, respectively.

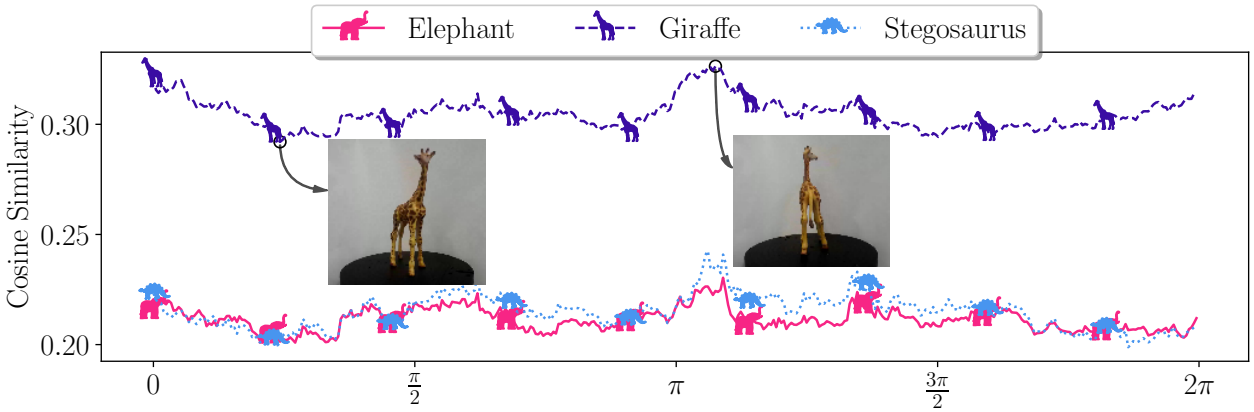
As observed in the main paper, the standard deviations are consistent during the intervention. However, they differ between text descriptors, as evident in the comparison between elephant and giraffe descriptors in Fig. 37a and Fig. 37b. Nevertheless, for specific combinations of interventional data and text phrases, we observe consistent variations. Moreover, consistent with the results in the main paper, the CLIP model [38] accurately predicts the shown images throughout the intervention, highlighting its robustness in our zero-shot classification setting.

In all three examples, we observe an approximately periodic behavior, which is a consequence of our experimental setup. However, the exact behavior varies significantly between the different toy figures. While the stegosaurus model exhibits a maximum similarity during a sideways orientation, this is not the case for the other figures. In contrast, the elephant and giraffe models achieve maximum similarity shortly after a rotation angle of π. Here, the giraffe's maximum corresponds to a sideways position of the head, which we believe is a crucial feature. Notably, the elephant's trunk remains visible during the complete intervention

We calculate the corresponding $\mathbb{E}[|\nabla_X|]$ for all combinations of interventional data and text descriptions in Table 14. The results show that the images containing the giraffe model have the smallest expected property gradient magnitudes, which aligns with the visualization in Fig. 36b and the low standard deviations in Fig. 37b. Additionally,



(a) Average cosine similarities in CLIP [38] latent space for images of a toy elephant with rotational interventions.



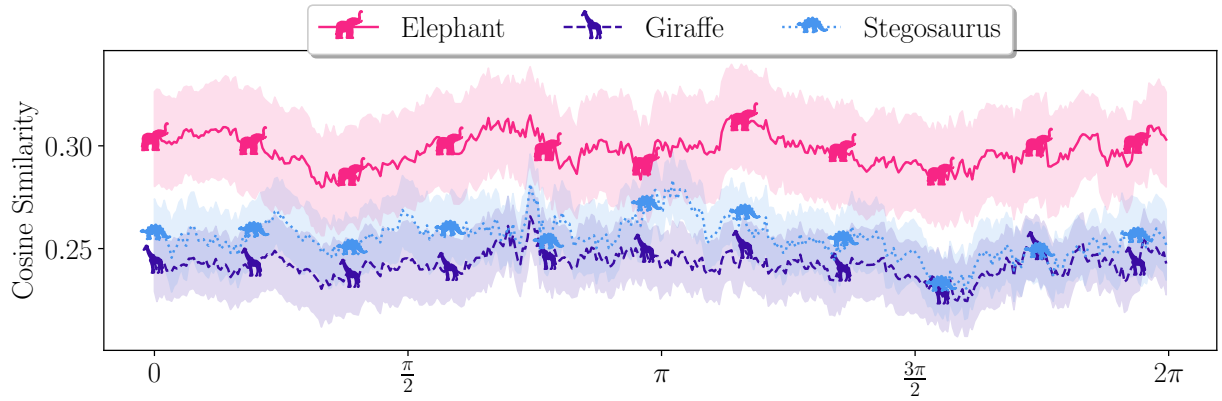
(b) Average cosine similarities in CLIP [38] latent space for images of a toy giraffe with rotational interventions.

Figure 36. Changes in CLIP [38] latent space cosine similarities for an intervention on the object position. We showcase real-life interventions (Fig. 8), where we rotate around the z-axis using a turntable. Here, we compare always images of the same toy model against description of all three classes. Additionally, we visualize the minimum and maximum similarity together with the corresponding image.

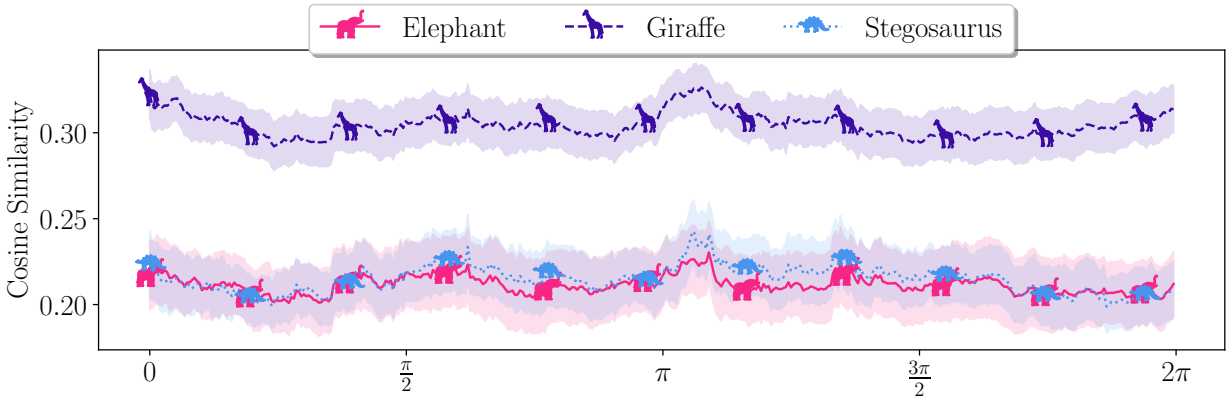
these low scores correspond to the highest average difference between the predicted class and the rejected classes. We also observe that the lowest scores in Table 14 are concentrated along the diagonal, indicating that the rotation impacts the cosine similarities empirically the least when the similarities are high. In contrast, we observe higher $\mathbb{E}[|\nabla_{\mathsf{X}}|]$ for the rejected classes. We will further investigate these observations using virtual interventions.

Virtual Interventions Fig. 38 visualizes the influence of different rotation axes for the virtual interventions. We find stark differences between the axes, with the most pronounced drop in similarity occurring for the rotation around the x-axis. Additionally, we note that the low point for the y-axis is achieved at approximately the same rotation angle. We visualize these minima and confirm our previous hypothesis that the model locally struggles with upside-down object orientations, resulting in lower similarity. The smaller decrease in similarity for the y-axis rotation may be attributed to the frontal view during rotation, which provides a more familiar perspective. The in-plane rotation around the z-axis provides additional evidence that upside-down positions are problematic. Specifically, we observe only smaller deviations in comparison. Ultimately, our analysis reveals a consistent pattern, underscoring the model's local vulnerability to upside-down orientations.

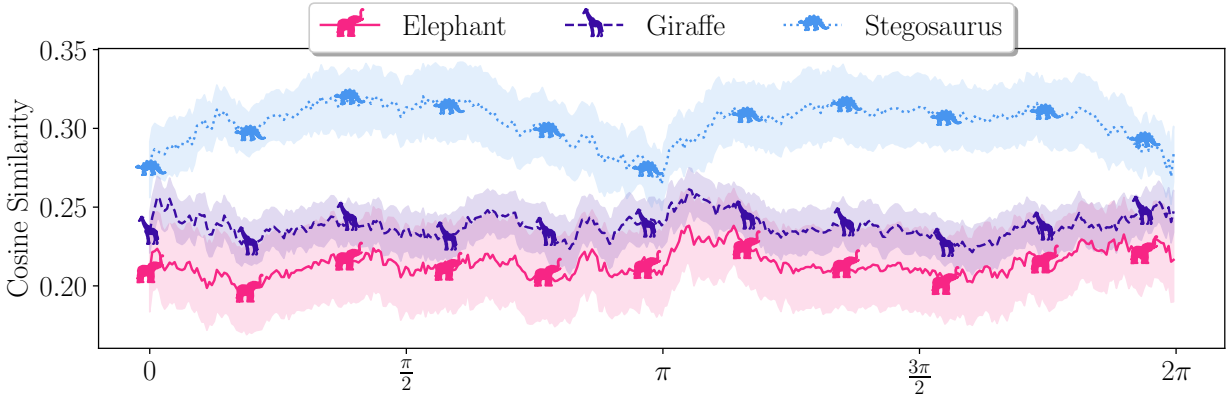
Similar to the real-life interventions, we observe consistent standard deviations throughout the complete interventions in Fig. 39. We visualize the mean behavior and the respective minima and maxima for all three interventions. Notably, the x-axis and y-axis rotation interventions exhibit



(a) Cosine similarities in CLIP [38] latent space for rotations of a toy elephant.



(b) Cosine similarities in CLIP [38] latent space for rotations of a toy giraffe.



(c) Cosine similarities in CLIP [38] latent space for rotations of a toy stegosaurus.

Figure 37. Changes in CLIP [38] latent space cosine similarities for an intervention on the object position. We showcase real interventions, where we rotate three different toys. We then compare against the three classes that are posed by these toys: elephant, giraffe, and stegosaurus. The standard deviations are calculated over ten different text descriptions each (Table 13).

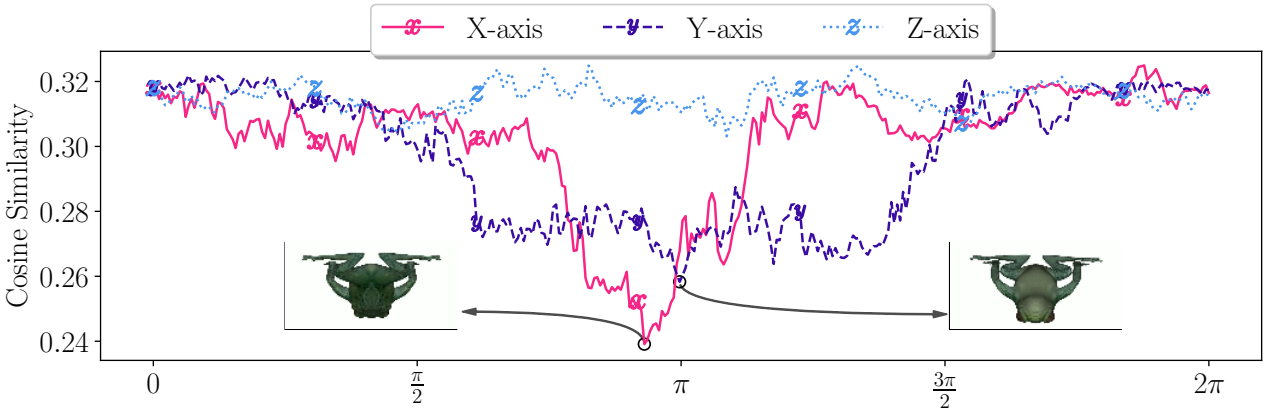
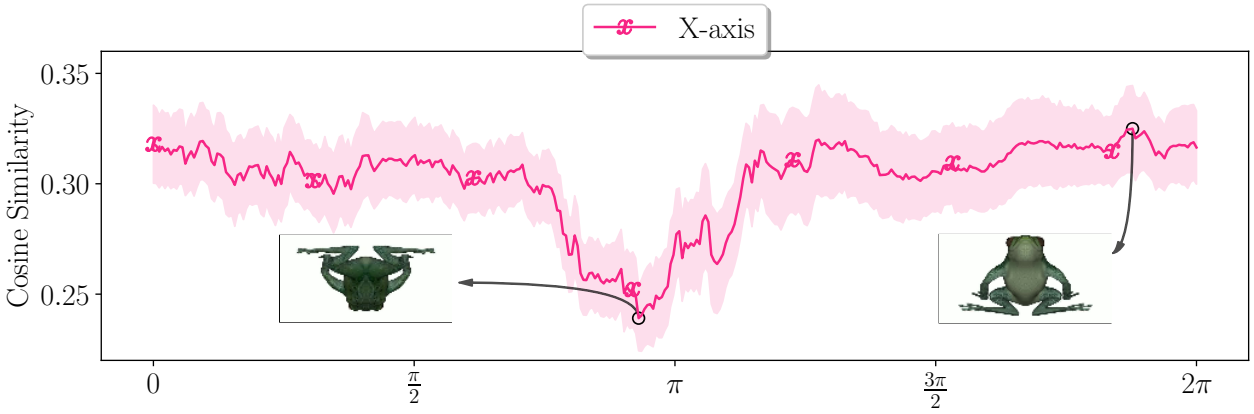


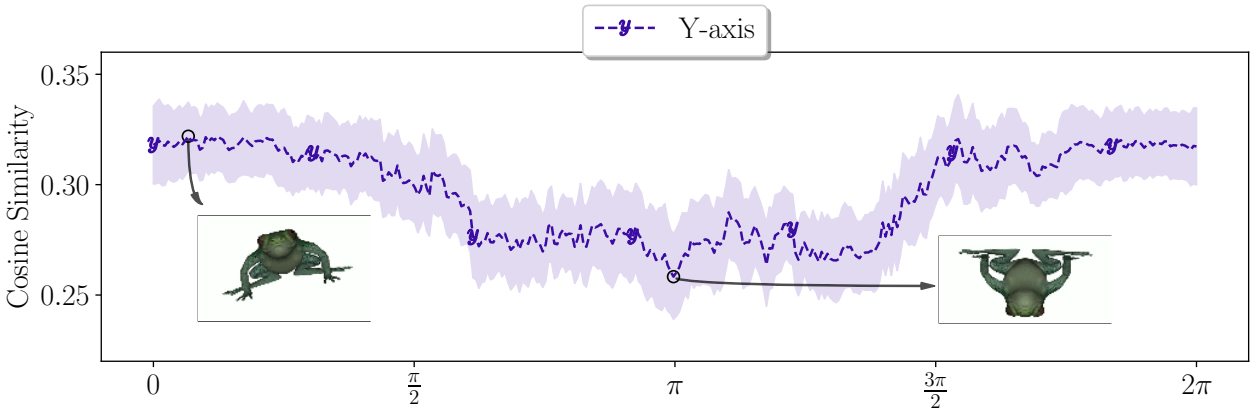
Figure 38. Average cosine similarities in CLIP [38] latent space for **virtual** interventional data. We mark the minima for the x and y-axis rotations.

high points for the upright position, with dips in cosine similarity observed for more uncommon upside-down orientations. Furthermore, we observe similar behavior to Fig. 8 under the inplane rotation around the z-axis. Specifically, the minimum cosine similarity is achieved around π, while the maximum is observed in a sideways orientation. In Table 15, we summarize the approximated $\mathbb{E}[|\nabla_{\mathsf{X}}|]$ scores for all three virtual rotations compared to the text embeddings in Table 13. These scores confirm our previous notion that the dependence on rotation is higher when average cosine similarities are lower. Specifically, we see the highest $\mathbb{E}[|\nabla_{\mathsf{X}}|]$ for the x-axis, which also exhibits the lowest observed cosine similarity in our virtual interventional data. Conversely, we find a small $\mathbb{E}[|\nabla_{\mathsf{X}}|]$ for the inplane rotation (z-axis), which aligns with the visualizations in Fig. 39.

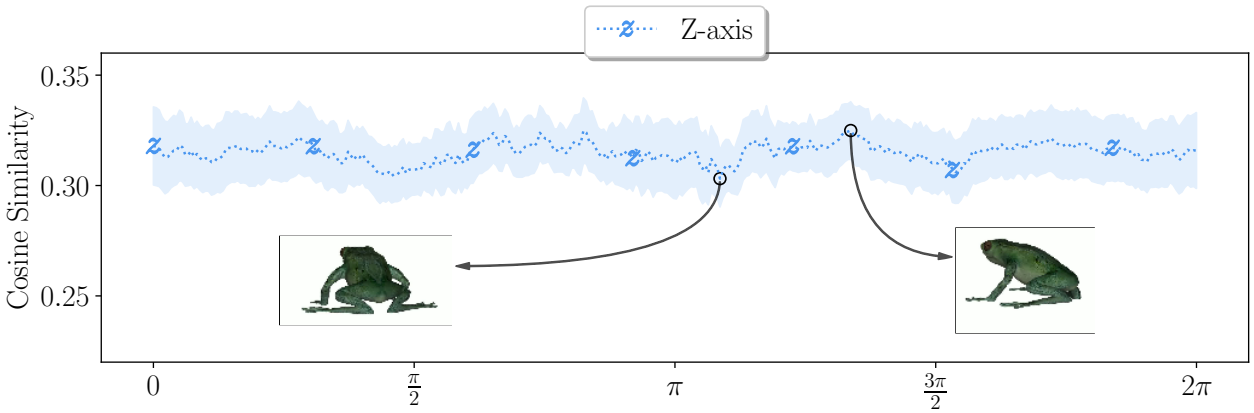
Overall, our additional visualizations and $\mathbb{E}[|\nabla_{\mathsf{X}}|]$ results provide further evidence to support the claims made in our main paper. Moreover, they demonstrate that our approach can effectively handle diverse sources of interventional data.



(a) Cosine similarities in CLIP [38] latent space for rotations around the x-axis.



(b) Cosine similarities in CLIP [38] latent space for rotations around the y-axis.



(c) Cosine similarities in CLIP [38] latent space for rotations around the z-axis.

Figure 39. Changes in CLIP [38] latent space cosine similarities for an intervention on the object position. We showcase virtual interventions, where we rotate a frog model around all three axes. Here, we additionally visualize the minimum, the maximum, and the standard deviation for ten text phrases used for the similarity calculations.

References

[1] International skin imaging collaboration, ISIC Archive. https://www.isic-archive.com/. 6, 16

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. 10

[3] Elias Bareinboim, Juan David Correa, Duligur Ibeling, and Thomas F. Icard. On pearl's hierarchy and the foundations of causal inference. *Probabilistic and Causal Inference*, 2022. 3, 9

[4] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 1, 3, 4, 8, 2, 5, 9

[5] Tim Büchner, Niklas Penzel, Orlando Guntinas-Lichius, and Joachim Denzler. Facing asymmetry–uncovering the causal link between facial symmetry and expression classifiers using synthetic interventions. *arXiv preprint arXiv:2409.15927*, 2024. 2, 3, 4, 16

[6] Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards Automated Circuit Discovery for Mechanistic Interpretability, 2023. 1, 2

[7] Will Cukierski. Dogs vs. cats. `https://kaggle.com/competitions/dogs-vs-cats`, 2013. Kaggle. 1, 5, 6, 3

[8] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models, 2023. 1, 2, 8

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 6, 7, 16, 17, 18, 20, 21, 22, 23

[10] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022. 2

[11] Bengt Fornberg. Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation*, 51: 699–706, 1988. 4, 3

[12] Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding Instruction-based Image Editing via Multimodal Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2024. 1, 2, 3, 4, 5, 7, 8, 6, 9, 10, 18, 19

[13] Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf. Kernel measures of conditional dependence. *Advances in neural information processing systems*, 20, 2007. 13, 15

[14] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders, 2024. 1, 2

[15] Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. Explaining classifiers with causal concept effect (cace). *arXiv preprint arXiv:1907.07165*, 2019. 1, 2, 3, 4, 8

[16] 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*, pages 770–778, 2016. 6, 7, 16, 17, 18, 20, 21, 22

[17] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 8, 9

[18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 1, 2, 3, 4, 8, 5

[19] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 7, 18, 20, 21, 22

[20] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 7, 18, 19, 20, 21, 22

[21] Josh Kaufmann. Google-10000-english: A list of the 10,000 most common english words. `https://github.com/first20hours/google-10000-english/tree/master`. 11, 15

[22] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. 1, 2, 8

[23] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*, 2020. 9, 10, 12

[24] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 5

[25] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 7, 18, 20, 23

[26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 7, 18, 19, 20, 21, 22

[27] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the

2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 6, 7, 16, 17, 18, 20, 21, 22

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 5, 16, 18

[29] 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*, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. 1, 2, 5, 6, 9, 10, 11, 12

[30] Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse Feature Circuits: Discovering and Editing Interpretable Causal Graphs in Language Models, 2024. 1, 2, 8

[31] Samuel G Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 774–782, 2021. 5, 16, 18

[32] Yushu Pan and Elias Bareinboim. Counterfactual image editing. In *Forty-first International Conference on Machine Learning*, 2024. 3, 4, 7, 8, 9, 16

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 18

[34] Judea Pearl. *Causality*. Cambridge University Press, 2009. 3, 1

[35] Karl Pearson. Notes on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895. 4, 1, 3, 18, 20

[36] Tristan Piater, Niklas Penzel, Gideon Stein, and Joachim Denzler. When medical imaging met self-attention: A love story that didn't quite work out. *arXiv preprint arXiv:2404.12295*, 2024. 17, 18

[37] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15691–15701, 2023. 8, 27

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 4, 8, 11, 13, 27, 29, 30, 31, 32, 33

[39] Christian Reimers, Jakob Runge, and Joachim Denzler. Determining the relevance of features for deep neural networks. In *European Conference on Computer Vision*, pages 330–346. Springer, 2020. 1, 2, 3, 5, 8, 11, 13, 15

[40] Christian Reimers, Niklas Penzel, Paul Bodesheim, Jakob Runge, and Joachim Denzler. Conditional dependence tests reveal the usage of abcd rule features and bias variables in automatic skin lesion classification. In *CVPR ISIC Skin Image Analysis Workshop (CVPR-WS)*, pages 1810–1819, 2021. 13, 17

[41] 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*, pages 1135–1144, 2016. 1, 2, 5, 6, 9, 10, 11, 12

[42] Laura Rieger, Chandan Singh, William Murdoch, and Bin Yu. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *International conference on machine learning*, pages 8116–8126. PMLR, 2020. 6, 16

[43] Karsten Roth, Jae Myung Kim, A Koepke, Oriol Vinyals, Cordelia Schmid, and Zeynep Akata. Waffling around for performance: Visual classification with random words and broad concepts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15746–15757, 2023. 8, 27

[44] Jakob Runge. Conditional independence testing based on a nearest-neighbor estimator of conditional mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 938–947. Pmlr, 2018. 13, 15

[45] 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. *International journal of computer vision*, 115:211–252, 2015. 6, 7, 11, 16, 17, 18, 19, 20, 21

[46] Adam Scherlis, Kshitij Sachan, Adam S. Jermyn, Joe Benton, and Buck Shlegeris. Polysemanticity and Capacity in Neural Networks, 2023. 1, 2

[47] Laines Schmalwasser, Jakob Gawlikowski, Joachim Denzler, and Julia Niebling. Exploiting text-image latent spaces for the description of visual concepts. In *International Conference on Pattern Recognition (ICPR)*, 2024. (accepted at ICPR). 2, 5, 11, 13, 15

[48] Alon Scope, Michael A Marchetti, Ashfaq A Marghoob, Stephen W Dusza, Alan C Geller, Jaya M Satagopan, Martin A Weinstock, Marianne Berwick, and Allan C Halpern. The study of nevi in children: Principles learned and implications for melanoma diagnosis. *J. Am. Acad. Dermatol.*, 75 (4):813–823, 2016. 6, 16, 17

[49] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: visual explanations from deep networks via gradient-based localization. *International journal of computer vision*, 128:336–359, 2020. 1, 2, 5, 6, 9, 10, 11, 12

[50] Rajen D. Shah and Jonas Peters. The hardness of conditional independence testing and the generalised covariance measure. *The Annals of Statistics*, 48(3):1514 – 1538, 2020. 11

[51] L. S. Shapley. *A Value for n-Person Games*, pages 307–318. Princeton University Press, Princeton, 1953. 11, 13

[52] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMlR, 2017. 1, 2, 5, 6, 9, 11, 12

[53] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017. 1, 2, 5, 6, 9, 11, 12

[54] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 6, 7, 16, 17, 18, 20, 21, 22

[55] Asher Trockman and J Zico Kolter. Patches are all you need? *Transactions on Machine Learning Research*, 2022. 5, 6, 7, 18, 19, 20, 21, 22

[56] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9 (86):2579–2605, 2008. 27, 29

[57] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 18

[58] Chih-Kuan Yeh, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in neural information processing systems*, 33: 20554–20565, 2020. 1, 2, 5, 8, 11, 13, 14, 15

[59] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing. 2, 3, 5, 6, 9, 10, 11, 12