

# DiG-IN: Diffusion Guidance for Investigating Networks - Uncovering Classifier Differences, Neuron Visualisations, and Visual Counterfactual Explanations

Maximilian Augustin

Yannic Neuhaus

Matthias Hein

Tübingen AI Center – University of Tübingen

## Abstract

*While deep learning has led to huge progress in complex image classification tasks like ImageNet, unexpected failure modes, e.g. via spurious features, call into question how reliably these classifiers work in the wild. Furthermore, for safety-critical tasks the black-box nature of their decisions is problematic, and explanations or at least methods which make decisions plausible are needed urgently. In this paper, we address these problems by generating images that optimize a classifier-derived objective using a framework for guided image generation. We analyze the decisions of image classifiers by visual counterfactual explanations (VCEs), detection of systematic mistakes by analyzing images where classifiers maximally disagree, and visualization of neurons and spurious features. In this way, we validate existing observations, e.g. the shape bias of adversarially robust models, as well as novel failure modes, e.g. systematic errors of zero-shot CLIP classifiers. Moreover, our VCEs outperform previous work while being more versatile.*

## 1. Introduction

Deep learning-based image classifiers suffer from several failure modes such as non-robustness to image corruptions [32, 40], spurious features and shortcuts [28, 54, 76], overconfidence on out-of-distribution inputs [31, 33, 55], adversarial examples [50, 82] or biases [27], among others.

While there has been a lot of work on detecting these failure modes, there remain two important problems that are addressed in this paper: i) systematic high-confidence predictions of classifiers, e.g. due to harmful spurious features [54], often occur on subgroups of out-of-distribution data. It is inherently difficult to find these subgroups as no data is available for them; ii) the visualization of the semantic meaning of concepts, e.g. of single neurons, or counterfactual explanations for image classifiers is extremely challenging as one has to optimize on the set of natural images and the optimization in pixel space leads to adversarial samples.

In this paper, we tackle these problems by leveraging recent progress in generative models [13, 61, 64, 67]. Our goal is to visualize properties of one or multiple image classifiers by optimizing on the approximation of the “natural image manifold” given by a latent diffusion model like Stable Diffusion [64]. This allows us to search for “unknown unknowns”, *i.e.* failure cases that correspond to a subpopulation of natural images which is neither easy to find in existing datasets nor allows for a textual description and is thus not amenable to direct prompting. We achieve this by using a generic framework for optimizing the inputs to a latent diffusion model to create realistic-looking images that minimize a loss function  $L$ , e.g. for the generation of images maximizing classifier disagreement, and VCEs and neuron visualizations, see Fig. 1 for an overview.

Using our DiG-IN framework we detect systematic failure cases of a zero-shot CLIP ImageNet classifier by maximizing the difference in the predicted probability for a given class, produce realistic visual counterfactuals for any image classifier outperforming [5], and provide neuron visualizations for a SE-ResNet and introduce Neuron Counterfactuals and evaluate them for neurons labeled as spurious in [76] of a ResNet50 ImageNet classifier.

## 2. Related Work

**Detection of systematic errors:** [26] develop a pipeline to iteratively retrieve real images from LAION-5B and label failure cases where the retrieval is refined based on the labels and additional LLM captions. [44] use a 3D simulator to generate and evaluate controlled scenes containing class objects to find systematic model vulnerabilities and validate these synthetic scenes in the real world by manual reconstruction of the scenes, whereas [73] try to find transformations which leave one classifier invariant but change another classifier. [22] leverage an error-aware mixture model on a multi-modal embedding to discover systematic errors in data subsets. [14, 51, 85] use a fixed set of attributes or properties of objects to search for systematic errors for subpopulations by generating corresponding user-interpretable prompts with

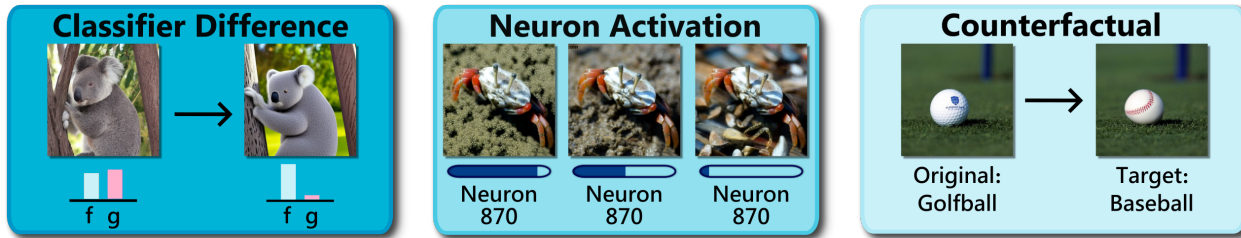


Figure 1. **Illustration of three tasks for debugging image classifiers.** **Left:** we generate images where one classifier is highly confident in a class and the other is not and recover the shape bias of adversarially robust models compared to a standard model; **Middle:** we generate images when maximizing or minimizing a neuron. We identify one neuron labeled as spurious for “fiddler crab” in [76] as associated to sand; **Right:** we produce visual counterfactual explanations for arbitrary image classifiers and outperform [5].

a fixed template structure. [12] use patch-attacks on a pixel level or restricted attacks on the latent space of an image generator to construct perturbations which are then pasted into images. As the added patches are not coherent with the original image, the resulting image is typically unrealistic. While some of these methods use generative models to search for systematic errors this is done with a fixed search pattern. Thus, problematic cases can be missed if not included in the pre-defined attribute set. In contrast, we optimize over the prompt/latent space and thus can find any problematic case as long as the diffusion model can generate it.

**Spurious features** are a particular failure mode where out-of-distribution images including the spurious features are confidently classified as a corresponding class, e.g. graffiti as “freight car” due to graffiti often appearing on training images of “freight car” in ImageNet. [54] label a spurious feature as harmful if it can mislead the classifier to classify the image as the corresponding class without the class object being present. Most existing methods are limited to smaller datasets or subsets of ImageNet [4, 58, 74, 75], only [52, 54, 76, 77] do a full search on ImageNet. [76] label neurons of a ResNet50 as “core” or “spurious” features by inspecting Grad-Cam images and feature attacks. We show that our prompt-based optimization allows for a much easier identification of spurious features by generating realistic images that maximize or minimize the neuron activation.

**Interpretability methods** are often motivated by detecting failure modes of a classifier. Very popular ones are, for example, attribution methods such as GradCAM [72], Shapley values [49], Relevance Propagation [8], and LIME [63]. These methods were analyzed with mixed success regarding the detection of spurious features in [1, 2]. Counterfactual explanations [86, 87] have recently become popular but are difficult to generate for images as the optimization problem is very similar to that of adversarial examples [82]. Visual counterfactual explanations are generated via manipulation of a latent space [71], using a diffusion model [5, 24] or in image space [6, 10, 68] for an adversarially robust classifier.

### 3. Method

#### 3.1. Background: Latent Diffusion Models

Score-based diffusion models [39, 78, 80] generate new samples from a data distribution  $p(x)$  by progressively denoising a latent vector drawn from a prior distribution. In this work, we focus on latent diffusion models (LDMs) [64, 83] that generate new samples in the latent space of a variational auto-encoder (VAE), where  $\mathcal{D}$  denotes the de- and  $\mathcal{E}$  the corresponding encoder. We use  $x$  to denote images in pixel- and  $z$  for images in VAE-latent space. During sampling, a random latent  $z_T$ , where  $T$  corresponds to the total number of sampling steps, is drawn from the prior distribution. We then produce less and less noisy samples  $z_{T-1}, z_{T-2}, \dots$  until we reach a noise-free VAE latent  $z_0$ , which can be transformed into pixel space using  $\mathcal{D}$  to produce the final image. The exact sequence  $(z_t)_{t=0}^T$  depends on the specific solver. While diffusion models initially used stochastic samplers [39], it has been shown that one can generate high-quality samples with deterministic solvers like DDIM [79], where the entire randomness lies in the initial latent  $z_T$ . The sequence of latents  $(z_t)_{t=0}^T$  for DDIM is then defined via:

$$z_{t-1} = \frac{\sqrt{\alpha_{t-1}}}{\sqrt{\alpha_t}} z_t - \frac{\sqrt{1 - \alpha_t} \epsilon(z_t, t, C)}{\sqrt{\alpha_t}} + \sqrt{1 - \alpha_{t-1}} \epsilon(z_t, t, C). \quad (1)$$

Here  $(\alpha_t)_{t=1}^T$  defines the noise schedule and  $\epsilon$  is the denoising model which is trained to predict the noise that was added to a noisy sample, see Appendix A for details.  $\epsilon$  is typically parameterized using a U-Net [65] where an additional conditioning signal can be employed to give the user control over the outcome of the diffusion process by sampling from a conditional distribution  $p(z|C)$ . In this work, we use the text-to-image Stable Diffusion [64] (SD) model where the conditioning signal  $C$  is a text encoding from a CLIP [60] text encoder which is fed into the U-Net via cross-attention layers. The SD model is trained on a large set of image-text pairs [70] and covers a variety of naturally occurring images. In practice, to amplify the impact of the conditioning, it is often necessary to employ classifier-free guidance [35],

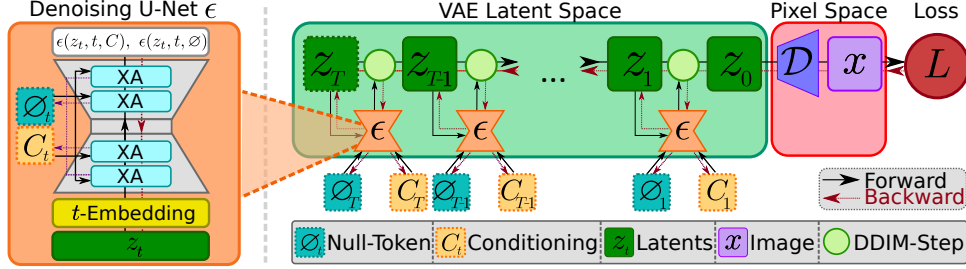


Figure 2. Illustration of the forward diffusion process (black arrows) from the initial latent  $z_T$  into the loss function  $L$  and the gradient flow during backpropagation (purple arrows). The optimization variables  $z_T$ ,  $(\emptyset_t)_{t=1}^T$  and  $(C_t)_{t=1}^T$  are marked with a dashed border. On the left, we illustrate the conditioning mechanism inside the denoising U-Net via cross-attention (XA) layers.

where  $\epsilon(z_t, t, C)$  in Eq. (1) is replaced with a combination of the conditional  $\epsilon(z_t, t, C)$  and an unconditional prediction  $\epsilon(z_t, t, \emptyset)$  with a null-text token  $\emptyset$ .

### 3.2. DiG-IN: Diffusion Guidance Framework for Investigating Neural Networks

Text-guided diffusion models have shown great success in generating highly realistic images. Several recent approaches for the detection of systematic errors leverage large text-to-image models [14, 51, 85] for the generation of images. They use fixed prompt templates describing specific properties of the desired input. However, these approaches are restricted to the variability of images encoded by their prompt templates and text guidance is often not precise enough. Our goal is an optimization framework where the image generation is directly guided by one or multiple classifiers (classifier disagreement and VCEs) or their properties (maximizing and minimizing neuron activations). Finding a text prompt that captures these tasks is just as hard as solving the task itself, *e.g.* if we want to find out what semantic concept maximizes a certain neuron we do not have access to a text description. While methods such as ControlNet [92] have shown great success at fine-grained conditioning of diffusion models, they require training samples that are not available for the tasks we want to solve and in addition, would require retraining for every vision classifier we want to explain.

However, it is easy to formulate our tasks as an optimization problem using a loss function  $L$  on the generated image. For example, we can easily calculate the activation of the target neuron from our previous example and search for highly activating images. Using the fact that the DDIM solver from the previous Section is non-stochastic, the output of the entire diffusion process is a deterministic function of the initial latent  $z_T$ , the conditioning  $C$  and the null-text token  $\emptyset$ . This allows us to formulate all our explanation tasks as optimization problems of the following form:

$$\max_{z_T, (C_t)_{t=1}^T, (\emptyset_t)_{t=1}^T} -L\left(\mathcal{D}(\mathbf{z}_0(z_T, (C_t)_{t=1}^T, (\emptyset_t)_{t=1}^T))\right). \quad (2)$$

Here, we use  $\mathbf{z}_0(z_T, (C_t)_{t=1}^T, (\emptyset_t)_{t=1}^T)$  to denote the

noise-free latent which is obtained by running the diffusion process from the initial latent  $z_T$ . Additionally, we use a separate conditioning  $C_t$  and null-text  $\emptyset_t$  for each time-step  $t \in \{1, \dots, T\}$  (see Figure 2). Intuitively, we search for a starting latent and conditioning that generates an image that optimizes our loss  $L$  without the need for manual prompt tuning or other forms of human supervision. We call this diffusion guidance framework DiG-IN. In the following Sections, we provide the corresponding loss function for each task. We want to highlight that this optimization framework is completely plug-and-play, *i.e.* it can be used with any vision model without requiring finetuning of the generative model. In practice, storing the entire diffusion process in memory for gradient computations is not possible due to VRAM limitations and we use gradient checkpointing [17] to compute the intermediate activations as required. See Algorithm 1 for pseudo-code.

### 4. Maximizing Classifier Disagreement

We generate maximally disagreeing images for a pair of two classifiers. This is a valuable tool to highlight differences caused by different training types, architectures, or pre-training and is particularly interesting for identifying subgroups where one classifier performs worse than the other. Forcing disagreement shifts the focus from prototypical examples of a class and makes this approach especially suitable for discovering unexpected failure modes on out-of-distribution images. Assume we are given two classifiers  $f, g$  and want to generate a realistic image that is predicted as target class  $y$  by  $f$  and not recognized by  $g$ . As objective we use the difference of confidences in the target class  $y$ :

$$\max_{z_T, (C_t)_{t=1}^T, (\emptyset_t)_{t=1}^T} p_f\left(y|\mathcal{D}(\mathbf{z}_0(z_T, (C_t)_{t=1}^T, (\emptyset_t)_{t=1}^T))\right) - p_g\left(y|\mathcal{D}(\mathbf{z}_0(z_T, (C_t)_{t=1}^T, (\emptyset_t)_{t=1}^T))\right). \quad (3)$$

We initialize the optimization with a random latent and the prompt: "a photograph of a <CLASSNAME>".

**Results:** Maximally disagreeing images are useful to explore subpopulations that capture classifier-specific biases and





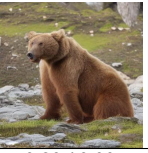
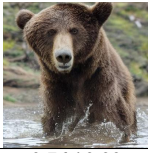





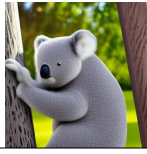


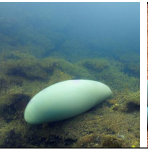

		$p_f$ : Confidence Robust ViT-S $\uparrow$ vs $p_g$ : Confidence ViT-S $\downarrow$							
		Head Cabbage ( $p_f / p_g$ )		Koala ( $p_f / p_g$ )		Brown Bear ( $p_f / p_g$ )		Dugong ( $p_f / p_g$ )	
		0.57 / 0.95	0.70 / 0.95	0.79 / 0.96	0.76 / 0.97	0.76 / 0.96	0.67 / 0.96	0.01 / 0.01	0.14 / 0.92
SD Init.									
		0.82 / 0.00	0.79 / 0.00	0.86 / 0.00	0.92 / 0.06	0.80 / 0.00	0.76 / 0.00	0.66 / 0.02	0.78 / 0.00
	$p_f \uparrow - p_g \downarrow$								

Figure 3. **Classifier disagreement: shape bias of adversarially robust models.** For a given class  $y$ , the first row shows the output of Stable Diffusion for “a photograph of  $y$ ”. The images in the second row have been optimized to maximize the confidence of an adversarially robust ViT-S while minimizing the one of a standard ViT-S. The resulting images retain the same shape but with smooth surfaces and little texture.

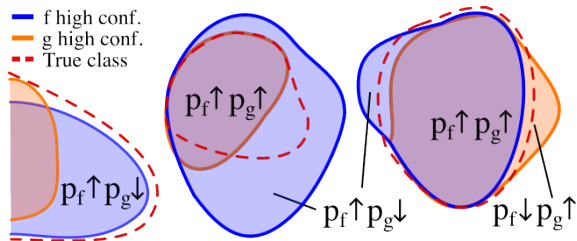


Figure 4. **Classifier Disagreement:** Images maximizing the disagreement between two classifiers  $f$  and  $g$  can reveal biases and failure modes of one or both classifiers. The three different variants we observe are: In the case of shape bias of robust models, the generated subpopulation has a schematic appearance but is still part of the true class (left). The zero-shot CLIP classifier extends the original class to a much larger set of out-of-distribution samples which causes unexpected failure modes (middle). When comparing the ViT and the ConvNext models, we find different biases by generating images inside as well as outside of the true class (right).

failure modes (Fig. 4). To demonstrate the versatility of this approach, we visualize the shape bias of adversarially robust models, failure cases due to the text embedding of zero-shot CLIP, and differences between a ViT and ConvNeXt.

**Shape bias of adv. robust models:** In Fig. 3 we show the difference between an adversarially trained ViT-S and a standard ViT-S. Both variants mostly give the correct prediction with high confidence on the initial Stable Diffusion outputs. Maximizing the predicted probability of the robust model while minimizing that of the standard ViT-S, produces visible changes in the texture, e.g. smooth cartoon-like surfaces, while retaining the shapes of the objects as well as their class. The standard classifier assigns zero confidence to the generated images, whereas the confidence of the robust one increases. This verifies the shape bias of adversarially trained models which was already observed in [15, 29, 93].

**Failure cases of zero-shot CLIP:** Next, we consider the maximally disagreeing images for an ImageNet classifier

(ConvNeXt-B) and a corresponding zero-shot CLIP (ViT-B-16 trained on LAION-2B) classifier (see Fig. 5). Here, we observe several failure modes specific to the properties of the zero-shot classifier which classifies based on the cosine similarity to a text embedding of the class name. In the first two examples, an image corresponding to only parts of the class name (“waffle” for “waffle iron”, “arch bridge” for “steel arch bridge”) achieves a high similarity for the CLIP model but low confidence for the ConvNeXt. The latter is even a misclassification, as an “arch bridge” made of stone is a “viaduct” which is another ImageNet class (we further investigate this error in Fig. 8). The generated images for the classes “wooden spoon” and “space bar” show a related pattern. In these cases, the composition of individual parts of the class name achieves a high score for the CLIP model but does not resemble the intended class objects in the training set. A spoon on a wooden table is classified as “wooden spoon” and the words “space bar” in front of a “space” background are classified as “space bar”. To verify these findings, we queried the LAION-5B image retrieval API for the text embeddings of “an image of waffle”, “an image of arch bridge”, “an image of a spoon on a wooden table”, and “an image of a bar in space”. These real images produce the same results (see the second row of Fig. 5).

**Comparing biases: ViT vs ConvNeXt:** We investigate the differences between a ViT-B and a ConvNeXt-B. We generate two images by maximizing the confidence of one while minimizing the other and vice versa (see Fig. 16 in App. B). We discover subtle biases when maximizing the ConvNeXt confidence for “goblet”: we generate empty wine glasses classified as “goblet” by the ConvNeXt and “red wine” by the ViT. Both of them are wrong, as the image does not contain an ImageNet object. Nevertheless, insights about such consistent behavior can help to detect failure modes that would occur after the release of the model and cannot be noticed by inspecting the training or test dataset.


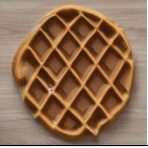
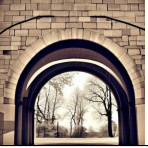

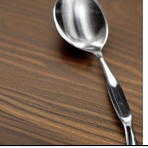
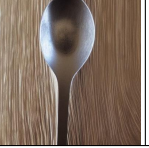
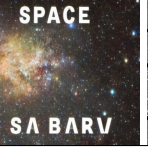
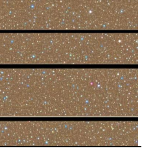








		$p_f$ : Confidence Zero-shot CLIP ImageNet classifier $\uparrow$				vs. $p_g$ : Confidence ConvNeXt-B $\downarrow$			
		Waffle Iron ( $p_f / p_g$ )		Steel Arch Bridge ( $p_f / p_g$ )		Wooden Spoon ( $p_f / p_g$ )		Space Bar ( $p_f / p_g$ )	
		1.00 / 0.01	1.00 / 0.00	1.00 / 0.00	1.00 / 0.00	0.98 / 0.00	0.92 / 0.04	1.00 / 0.00	0.99 / 0.00
$p_f \uparrow - p_g \downarrow$									
		Validation of CLIP zero-shot errors on <b>real</b> images from LAION-5B with retrieval query "an image of ..."							
		".. a waffle"		".. an arch bridge"		".. a spoon on a wooden table"		".. a bar in space"	
		1.00 / 0.18	1.00 / 0.02	0.98 / 0.00	0.99 / 0.00	0.94 / 0.00	0.99 / 0.07	0.81 / 0.00	0.40 / 0.00
Real Images									

Figure 5. **Detection of errors of the zero-shot CLIP model (ImageNet):** we generate a SD image with the prompt “a photograph of <CLASSNAME>”. Starting from this image, we maximize the difference between the predicted probability for the target class of a zero-shot CLIP ImageNet model and a ConvNeXt-B trained on ImageNet (first row). We find subpopulations of images that are systematically misclassified by the CLIP model: waffles are classified as “waffle iron”, stone bridges as “steel arch bridges”, spoons on a wooden table as “wooden spoon”, and images with space and bar as “space bar”. In the second row we validate these errors by finding similar real images in LAION-5B (see App. C). The errors of CLIP are most likely an artefact of the text embeddings due to the composition of the class name.

## 5. Visual Counterfactual Explanations

Counterfactual reasoning has become a valuable tool for understanding the behavior of models. For image classifiers, a Visual Counterfactual Explanation (VCE) [5, 10] for input  $\hat{x}$ , target class  $y$  and classifier  $f$  is a new image  $x$ , that **i)** is classified as  $y$  by  $f$  (actionable), **ii)** looks realistic (on the natural image manifold), **iii)** contains minimal changes to the input  $\hat{x}$ . In particular, that the VCE  $x$  is actionable distinguishes it from other explanation techniques. Prior methods that generate VCEs for ImageNet require an additional dataset-specific adversarially robust model [5]. In contrast, our method is training-free and produces VCEs for *any* classifier trained on *any* dataset containing natural images. We thus refer to our generated counterfactuals as *Universal VCE (UVCE)*.

VCE generation is a challenging image-to-image task. The loss for VCE generation has to include a similarity measure to the original image in addition to the predicted probability of  $f$  in class  $y$ . As the optimization problem is highly non-convex, we need a good initialization for better performance and convergence. We describe our method in the following (see Appendix D for pseudo-code and details).

**VCE Initialization:** As the VCE should be similar to the original image, random initialization is suboptimal. To find a latent  $z_T$  that reproduces the image  $\hat{x}$ , we use Null-Text inversion [53] which, on top of the latent  $z_T$  optimizes a per-time step null-token  $(\emptyset_t)_{t=1}^T$  to improve reconstruction. As the inversion is dependent on the text conditioning and we want a fully automated pipeline, we need a text description  $\hat{P}$  of  $\hat{x}$ . We use Open-Flamingo [3, 7] to extend the generic caption “an image of a <ORIGINAL CLASSNAME>” with additional details and then decode this caption using the

CLIP encoder in SD to get an initial conditioning  $\hat{C}$ . By doing so, we can find  $(z_T, \hat{C}, (\emptyset_t)_{t=1}^T)$  that closely reconstruct the original image. In order to get an even better initialization, we make use of the extensive knowledge contained in SD. We replace the original class name with the name of the target class in the prompt  $\hat{P}$  to get a modified prompt  $P$ , so “an image of a dog at the beach” becomes “an image of a cat at the beach”. This prompt can be decoded into a new conditioning  $C$  that contains the target class. Due to the change from  $\hat{C}$  to  $C$ , reconstructing the image with the new conditioning  $C$  yields images with different overall structure. We thus use a modified version of Prompt-to-Prompt from [34], who found that one can preserve structure by injecting cross-attention (XA) maps. This style of editing often results in a good initialization, but several issues prevent it from being a VCE method on its own. Most importantly, as  $f$  is not involved, the resulting images often have low confidence and secondly, it induces more changes than necessary, see Figure 18a. To overcome those issues, we propose to jointly optimize the confidence and distance to the starting image.

**VCE Optimization:** To ensure the similarity of the VCE  $x$  to the starting image  $\hat{x}$ , we want to change the class object while preserving the background. Prior works [5, 10] use  $L_p$  regularization between  $x$  and  $\hat{x}$  to keep the changes minimal. However,  $L_p$  distances between images depend heavily on the size of the foreground object. If the class object is small, we only want to allow minimal changes in the image, while for larger class objects we need to allow larger changes. As the XA maps encode the locations that are most influenced by a specific text token, we can use them to produce point prompts for computing segmentation maps in the VAE-latent

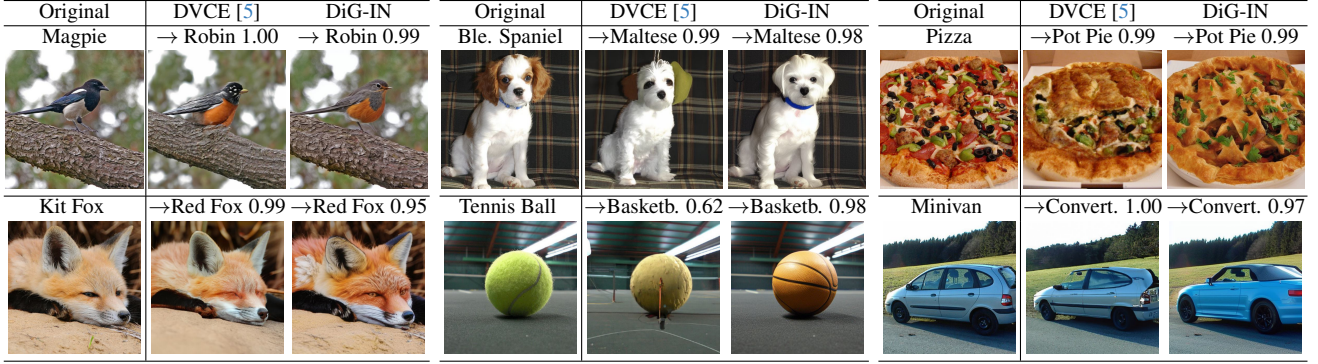


Figure 6. **ImageNet VCEs**: We show the original ImageNet validation image as well as DVCEs [5] and our DiG-IN UVCEs with the corresponding confidence in the target class. Our UVCEs are more realistic looking and produce fine-grained texture changes ("Red Fox", "Basketb.") as well as more complex geometric transformations ("Pot Pie", "Convertible") where DVCE can fail to create a coherent object.

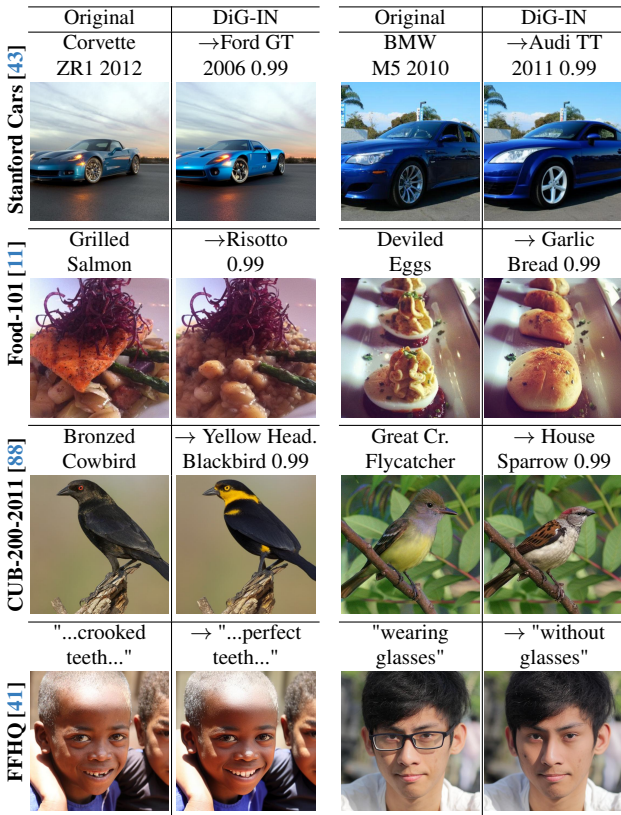


Figure 7. **UVCEs** for various datasets. DiG-IN is the first training-free method that can generate highly realistic VCEs for any dataset containing natural images without requiring a dataset-specific generative model or an adversarially robust classifier.

( $S_{VAE}$ ) and pixel space ( $S_{PX}$ ) using HQ-SAM [42], where  $S_{i,j} \approx 1$  if location  $(i, j)$  corresponds to the foreground object. We define our foreground aware distance regularization that penalizes background changes to the original image  $\hat{x}$  and its VAE encoding  $\mathcal{E}(\hat{x})$  while simultaneously allowing for large changes in color and shape in the foreground:

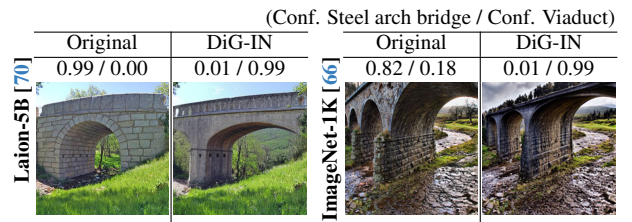


Figure 8. **Zero-shot CLIP UVCEs**: 14% of the ImageNet validation images of class "viaduct" are misclassified as "steel arch bridge" by zero-shot CLIP (Fig. 5). We generate UVCEs for **wrongly** classified images with the correct class "viaduct" as target. The classifier seems to distinguish the two classes based on the shape of the arch. This shows that the CLIP model has learned a wrong decision boundary and how UVCEs can be used to understand systematic misclassifications, e.g. narrow stone bridges that are classified as "steel arch bridge" instead of "viaduct".

$$d(z, \hat{x}) = w_{VAE} \|(1 - S_{VAE}) \odot (z - \mathcal{E}(\hat{x}))\|_2^2 + w_{PX} \|(1 - S_{PX}) \odot (\mathcal{D}(z) - \hat{x})\|_2^2. \quad (4)$$

The final loss for the VCE generation is then given by:

$$\max_{z_T, (C_t)_{t=1}^T, (\emptyset_t)_{t=1}^T} -d\left(\mathbf{z}_0(z_T, (C_t)_{t=1}^T, (\emptyset_t)_{t=1}^T), \hat{x}\right) + \log p_f\left(y | \mathcal{D}(\mathbf{z}_0(z_T, (C_t)_{t=1}^T, (\emptyset_t)_{t=1}^T))\right). \quad (5)$$

**Evaluation**: We compare our DiG-IN UVCEs to DVCEs [5] which is the most recent VCE method that works on ImageNet. We emphasize that, unlike DVCEs, we do not require a robust classifier or a dataset-specific diffusion model. We generate counterfactuals into classes that are close in the ImageNet hierarchy and show qualitative results in Fig. 6. While DVCEs work well for some images, they often produce unrealistic results. For example, for "Basketball" or "Convertible", DVCEs contain some features of the target class but the method fails to create a coherent object. In other

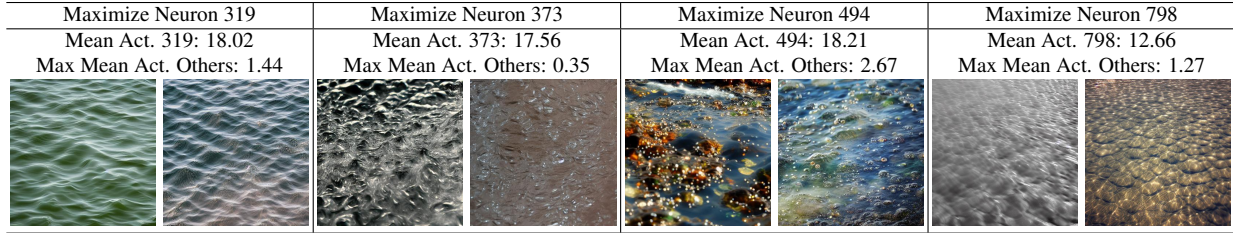


Figure 9. **Neuron visualization for a SE-ResNet-D 152 [90] trained on ImageNet:** Our neuron visualization allows to identify subtle differences between four neurons which are all activated by some kind of “water”. Interestingly, the individual neurons are maximally activated only for a specific type of “water” and show no strong activations for the images generated where the other neurons are maximized.

cases, some parts of the generated class seem artificial or illogical like the ear of the dog and the basketball texture. In contrast, our approach consistently produces more realistic changes. To validate our method, we did a user study on randomly selected images where we asked the participants to rate if “the counterfactual image ” **Q1**) “... is realistic” **Q2**) “... shows meaningful features of the target class” **Q3**) “... changes mainly the class object”. We also asked the participants to directly rate whether the DVCE or the UVCE counterfactual is better or if both are equal. Results are in Table 1 and further details and the images of the study are in Appendix E. Users rated our DiG-IN UVCEs as more realistic and as better showing the features of the target class. Our UVCEs were preferred over DVCEs in 59.5% of cases, 18.1% preferred DVCEs and 22.5% rated both equal.

	Q1	Q2	Q3	Better?
DVCE[5]	40.4%	63.7%	73.8%	18.1 %
UVCE	76.0%	81.3%	89.1%	59.5%

Table 1. **User Study.** Our UVCEs are rated as more realistic (Q1), showing better features of the target class (Q2), and overall better.

We emphasize that, unlike previous approaches like DVCE, we can generate our UVCEs for *any* image classifier (no robustness or specific diffusion model required) on *any* natural image dataset and we show examples for Cars, CUB, and Food as well as zero-shot attribute classification on FFHQ in Fig. 7 and additional examples in Appendix D. In addition, Fig. 8 contains an error analysis of CLIP using DiG-IN where we visualize what a *wrongly* predicted image would have to look like to be correctly classified and we present more UVCEs for images misclassified by a EVA02 [23] in Fig. 25 in the Appendix D.

## 6. Neuron Activation

In the next task, we want to visualize the semantic meaning of specific neurons in the last layer of a classification model. While the neurons in earlier layers of DNNs and convolutional NNs in particular, are thought to capture low-level image features like corners and edges, neurons in the last

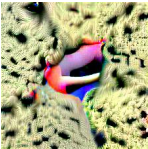



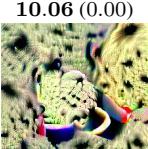



layer are meant to capture more semantically meaningful concepts [21]. For this task, assume we are given a classifier  $f$  and let  $\phi : \mathbb{R}^D \rightarrow \mathbb{R}^N$  denote the function that maps an input image into its feature representation at the final layer before the linear classification head. Let  $n$  be the target neuron  $n \in \{1, \dots, N\}$  we want to visualize. Our objective is to maximize the activation of that neuron using the objective:

$$\max_{z_T, (C_t)_{t=1}^T, (\emptyset_t)_{t=1}^T} \phi \left( \mathcal{D} \left( \mathbf{z}_0 \left( z_T, (C_t)_{t=1}^T, (\emptyset_t)_{t=1}^T \right) \right) \right)_n. \quad (6)$$

We demonstrate two visualization methods, one that generates synthetic prototypical images that highly activate a target neuron and introduce *Neuron Counterfactuals*.

**Synthetic Neuron Visualizations:** Our goal is to generate prototypical examples that visualize the target neuron  $n$ . A common way to identify the concepts captured by a neuron is to inspect highly active training images. However, such subpopulations usually differ in many aspects which makes this analysis ambiguous. For our optimization, we need an initial conditioning  $C$  which ideally relates to the objects that maximize this neuron. To get this, we use CogAgent [37] to list the objects in the most activating train images for that neuron. For each object, we use SD to generate images for the prompt: “a photograph of a <OBJECT>” and use the one with the highest mean activation for our initial conditioning  $C$  and optimize Eq. (6). We show results for 4 different “water” neurons in Fig. 9. Additional results and details can be found in Appendix F, where we also demonstrate the advantages over inspecting maximally active train images and prompt-based approaches (Fig. 29).

**Neuron Counterfactuals:** It has been shown that the neurons that are the most impactful for a classifier’s decision are often activated by the image background instead of the class object [54, 76]. To visualize this, we max- or minimize the activation of a potentially spurious neuron starting from the same Null-Text inversion of a *real* image we used in Sec. 5. Unlike for UVCEs, we now want to allow for background changes to insert or remove the spurious feature while preserving the class object. To achieve this, we use the distance term Eq. (4) without inverting the foreground mask.

Neuron 870 (Conf. class Fiddler crab)			
[76] Max. Neuron 870	Maximize Neuron 870	← Test Image	→ Minimize Neuron 870
9.76 (0.00)	5.74 (0.99)	2.24 (0.93)	0.02 (0.04)
			
10.06 (0.00)	3.10 (0.95)	1.31 (0.86)	0.17 (0.16)
			


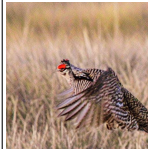
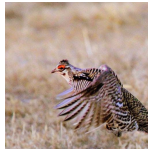
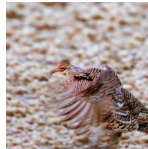
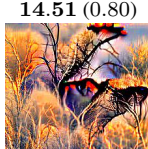
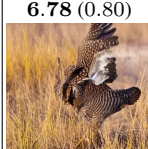
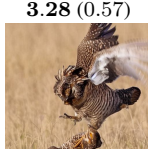
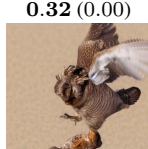
Neuron 565 (Conf. class Prairie chicken)			
[76] Max. Neuron 565	Maximize Neuron 565	← Test Image	→ Minimize Neuron 565
14.07 (0.62)	5.88 (0.97)	3.23 (0.87)	0.08 (0.01)
			
14.51 (0.80)	6.78 (0.80)	3.28 (0.57)	0.32 (0.00)
			

Figure 10. **Neuron Counterfactuals:** We visualize neurons marked as spurious in [76]. Starting from a test image, we max- and minimize the value of the corresponding spurious neuron. As comparison, we show the result of the feature attack maximizing the neuron of [76]. Our resulting images convey the semantic meaning of the neuron, whereas the feature attack is too extreme. For the class “fiddler crab”, maximizing the spurious neuron enhances the sandy background in the image, whereas minimizing the neuron removes the sand. Similarly, the semantic feature “dry gras” is amplified or removed in the “prairie chicken” images when the spurious neuron is maximized or minimized.









	Class 2 - NPCA Comp. 1 (Conf. class Great White Shark)		Class 554 - NPCA Comp. 2 (Conf. class Fireboat)	
	Fireboat	American Alligator	Grey Whale	Pirate
	-2.85 (0.02)	-3.53 (0.09)	-1.02 (0.00)	-1.20 (0.22)
Test Image				
	3.38 (0.29)	1.01 (0.41)	5.10 (0.97)	2.63 (0.99)
Max. NPCA				

Figure 11. **Validating harmful spurious features:** [54] identify NPCA components of certain classes as *harmful* spurious features, i.e. their presence alone is sufficient to trigger prediction of the class, by searching maximally activating images. We validate this property directly by maximizing the NPCA component (details in G) starting from images of other classes (top row). Left: Maximizing NPCA comp. 1 of great white shark changes the water surface and yields prediction ‘great white shark’ even though the ‘fireboat’ and ‘American alligator’ are still visible and no features of a shark are generated. Right: Same for the NPCA comp. 2 of fireboat.

Generated images that maximize individual neurons have already been used to detect spurious features [76]. In Fig. 10, we compare our approach to their “Feature attack”. Their procedure achieves a higher neuron activation but the resulting images lack realism as they show mostly artificial patterns. In addition, they mostly reduce the confidence in the spuriously correlated class. On the other hand, our results convey a clearer interpretation of the corresponding semantic concept: Maximizing the neurons amplifies the presence

of the corresponding spurious concepts (“sand” for “fiddler crab” and “dry gras” for “prairie chicken”), whereas minimizing removes them completely. Due to our regularization, the class object shows only minimal changes, however, we see that the confidence into the class changes dramatically depending on the activation of that neuron. This strongly suggests that both of them are cases of harmful spurious features, i.e. their presence in images that do not contain the actual class already triggers the prediction of the class. We specifically validate the harmful spurious features found by [54] in Fig. 11 where we start from the image of a *different* class and maximize the NPCA component [54], see Fig. 35 and Appendix G for details. We show more neuron counterfactuals in Appendix F and provide a quantitative evaluation of core and spurious neurons of [76] in Appendix F.3.

## 7. Conclusion

In this work, we have introduced a framework for analyzing and explaining *any* differentiable image classifier via diffusion guidance. We demonstrated that it enables flexible detection of systematic biases on in- and out-of-distribution data. Additionally, our work improves the understanding of classifier decisions by creating realistic and interpretable visualizations of individual neurons as well as better and more universal visual counterfactual explanations. See Appendix H for limitations and failure cases.

## Acknowledgements

We are grateful for support by the DFG, Project number 390727645, and the Carl Zeiss Foundation, project “Certification and Foundations of Safe Machine Learning Systems in Healthcare” and thank the IMPRS-IS for supporting YN.



## References

- [1] Julius Adebayo, Michael Muelly, Hal Abelson, and Been Kim. Post hoc explanations may be ineffective for detecting unknown spurious correlation. In *ICLR*, 2022. 2
- [2] Julius Adebayo, Michael Muelly, Ilaria Lliccardi, and Been Kim. Debugging tests for model explanations. In *NeurIPS*, 2020. 2
- [3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 5, 18
- [4] Christopher J. Anders, Leander Weber, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. Finding and removing clever hans: Using explanation methods to debug and improve deep models. *Information Fusion*, 77:261–295, 2022. 2
- [5] Maximilian Augustin, Valentyn Boreiko, Francesco Croce, and Matthias Hein. Diffusion visual counterfactual explanations. In *NeurIPS*, 2022. 1, 2, 5, 6, 7, 12, 22, 25
- [6] Maximilian Augustin, Alexander Meinke, and Matthias Hein. Adversarial robustness on in- and out-distribution improves explainability. In *ECCV*, 2020. 2, 15
- [7] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023. 5, 18
- [8] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *JMLR*, 2010. 2
- [9] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. *arXiv preprint arXiv:2302.07121*, 2023. 13
- [10] Valentyn Boreiko, Maximilian Augustin, Francesco Croce, Philipp Berens, and Matthias Hein. Sparse visual counterfactual explanations in image space. In *GCPR*, 2022. 2, 5
- [11] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014. 6, 26
- [12] Stephen Casper, Max Nadeau, Dylan Hadfield-Menell, and Gabriel Kreiman. Robust feature-level adversaries are interpretability tools. In *NeurIPS*, 2022. 2
- [13] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 1
- [14] Muxi Chen, Yu Li, and Qiang Xu. Hibus: On human-interpretable model debug. In *NeurIPS*, 2023. 1, 3
- [15] Peijie Chen, Chirag Agarwal, and Anh Nguyen. The shape and simplicity biases of adversarially robust imagenet-trained cnns. *arXiv preprint arXiv:2006.09373*, 2020. 4
- [16] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *NeurIPS*, 2018. 13
- [17] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016. 3
- [18] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 13
- [19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 14, 25, 26
- [20] Rachel Lea Draelos and Lawrence Carin. Use hirescam instead of grad-cam for faithful explanations of convolutional neural networks. *arXiv preprint arXiv:2011.08891*, 2020. 31
- [21] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations, 2019. 7
- [22] Sabri Eyuboglu, Maya Varma, Khaled Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. Domino: Discovering systematic errors with cross-modal embeddings. In *ICLR*, 2022. 1
- [23] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*, 2023. 7, 22, 24, 28
- [24] Karim Farid, Simon Schrodi, Max Argus, and Thomas Brox. Latent diffusion counterfactual explanations. *arXiv preprint arXiv:2310.06668*, 2023. 2
- [25] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruva Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *arXiv preprint arXiv:2304.14108*, 2023. 23
- [26] Irena Gao, Gabriel Ilharco, Scott Lundberg, and Marco Tulio Ribeiro. Adaptive testing of computer vision models. In *ICCV*, 2023. 1
- [27] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019. 1
- [28] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 1
- [29] Robert Geirhos, Kantharaju Narayanappa, Benjamin Mitzkus, Tizian Thieringer, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Partial success in closing the gap between human and machine vision. *NeurIPS*, 2021. 4

- [30] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 22
- [31] M. Hein, M. Andriushchenko, and J. Bitterwolf. Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *CVPR*, 2019. 1
- [32] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. 1
- [33] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017. 1
- [34] 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. 5, 19, 20, 21
- [35] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop*, 2021. 2
- [36] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 13
- [37] Wenyi Hong, Weihang Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and Jie Tang. Cogagent: A visual language model for gui agents. *arXiv preprint arXiv:2312.08914*, 2023. 7, 23, 32
- [38] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. 23
- [39] Pieter Abbeel Jonathan Ho, Ajay Jain. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2, 12, 13
- [40] Oğuzhan Fatih Kar, Teresa Yeo, Andrei Atanov, and Amir Zamir. 3d common corruptions and data augmentation. In *CVPR*, 2022. 1
- [41] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, 2020. 6, 23
- [42] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. In *NeurIPS*, 2023. 6, 20, 22, 31
- [43] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, 2013. 6, 27
- [44] Guillaume Leclerc, Hadi Salman, Andrew Ilyas, Sai Vemprala, Logan Engstrom, Vibhav Vineet, Kai Xiao, Pengchuan Zhang, Shibani Santurkar, Greg Yang, Ashish Kapoor, and Aleksander Madry. 3db: A framework for debugging computer vision models. In *NeurIPS*, 2022. 1
- [45] Wei Li, Xue Xu, Xinyan Xiao, Jiachen Liu, Hu Yang, Guohao Li, Zhanpeng Wang, Zhifan Feng, Qiaoqiao She, Yajuan Lyu, et al. Upainting: Unified text-to-image diffusion generation with cross-modal guidance. *arXiv preprint arXiv:2210.16031*, 2022. 13
- [46] Xianhang Li, Zeyu Wang, and Cihang Xie. An inverse scaling law for clip training. In *NeurIPS*, 2023. 23, 27
- [47] Shanchuan Lin, Anran Wang, and Xiao Yang. Sd-xl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024. 38
- [48] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 22, 24
- [49] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NeurIPS*, 2017. 2
- [50] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 1
- [51] Jan Hendrik Metzen, Robin Huttmacher, N. Grace Hua, Valentyn Boreiko, and Dan Zhang. Identification of systematic errors of image classifiers on rare subgroups. In *ICCV*, 2023. 1, 3
- [52] Mazda Moayeri, Sahil Singla, and Soheil Feizi. Hard ImageNet: Segmentations for objects with strong spurious cues. In *NeurIPS Datasets and Benchmarks Track*, 2022. 2
- [53] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022. 5, 14, 18
- [54] Yannic Neuhaus, Maximilian Augustin, Valentyn Boreiko, and Matthias Hein. Spurious features everywhere – large-scale detection of harmful spurious features in imagenet, 2023. 1, 2, 7, 8, 12, 31, 37
- [55] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015. 1
- [56] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022. 13
- [57] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *ICML*, 2022. 15
- [58] Gregory Plumb, Marco Tulio Ribeiro, and Ameet Talwalkar. Finding and fixing spurious patterns with explanations. *Transactions on Machine Learning Research (TMLR)*, 2022. 2
- [59] 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 *ICML*, 2021. 14
- [60] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2
- [61] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1
- [62] Yongming Rao, Guangyi Chen, Jiwen Lu, and Jie Zhou. Counterfactual attention learning for fine-grained visual categorization and re-identification. In *ICCV*, 2021. 22, 27

- [63] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *KDD*, 2016. 2
- [64] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 2, 12, 13
- [65] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2, 14
- [66] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 6, 25
- [67] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 2022. 1, 14
- [68] Shibani Santurkar, Dimitris Tsipras, Brandon Tran, Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Image synthesis with a single (robust) classifier. In *NeurIPS*, 2019. 2
- [69] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023. 38
- [70] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022. 2, 6, 22, 24
- [71] Kathryn Schutte, Olivier Moindrot, Paul Hérent, Jean-Baptiste Schiratti, and Simon Jégou. Using stylegan for visual interpretability of deep learning models on medical images. In *NeurIPS Workshop*, 2020. 2
- [72] 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. In *ICCV*, 2017. 2
- [73] Harshay Shah, Sung Min Park, Andrew Ilyas, and Aleksander Madry. ModelDiff: A framework for comparing learning algorithms. In *ICML*, volume 202. PMLR, 2023. 1
- [74] Rakshith Shetty, Bernt Schiele, and Mario Fritz. Not using the car to see the sidewalk—quantifying and controlling the effects of context in classification and segmentation. In *CVPR*, 2019. 2
- [75] Krishna Kumar Singh, Dhruv Mahajan, Kristen Grauman, Yong Jae Lee, Matt Feiszli, and Deepti Ghadiyaram. Don't judge an object by its context: learning to overcome contextual bias. In *CVPR*, 2020. 2
- [76] Sahil Singla and Soheil Feizi. Salient imagenet: How to discover spurious features in deep learning? In *ICLR*, 2022. 1, 2, 7, 8, 31, 34, 35, 36
- [77] Sahil Singla, Mazda Moayeri, and Soheil Feizi. Core risk minimization using salient imagenet. *arXiv:2203.15566*, 2022. 2
- [78] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 2, 12, 13
- [79] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 2, 12, 13, 18
- [80] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 2, 12, 13
- [81] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. In *TMLR*, 2022. 22, 24, 25, 26
- [82] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 1, 2
- [83] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *NeurIPS*, 2021. 2, 12
- [84] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 14
- [85] Joshua Vendrow, Saachi Jain, Logan Engstrom, and Aleksander Madry. Dataset interfaces: Diagnosing model failures using controllable counterfactual generation. *arXiv:2302.07865*, 2023. 1, 3
- [86] Sahil Verma, John P. Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review. *arXiv preprint, arXiv:2010.10596*, 2020. 2
- [87] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 2018. 2
- [88] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, California Institute of Technology, 2011. 6, 26
- [89] Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-end diffusion latent optimization improves classifier guidance. *arXiv preprint arXiv:2303.13703*, 2023. 15
- [90] Ross Wightman, Hugo Touvron, and Herve Jegou. Resnet strikes back: An improved training procedure in timm. In *NeurIPS Workshop on ImageNet: Past, Present, and Future*, 2021. 7, 32, 33
- [91] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *CVPR*, 2023. 22, 24
- [92] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 3
- [93] Tianyuan Zhang and Zhanxing Zhu. Interpreting adversarially trained convolutional neural networks. *ICML*, 2019. 4