Discriminative Class Tokens for Text-to-Image Diffusion Models

Idan Schwartz[1]* Vésteinn Snæbjarnarson[2]* Hila Chefer[1]
Serge Belongie[2] Lior Wolf[1] Sagie Benaim[2]
[1]Tel Aviv University [2]University of Copenhagen

Abstract

Recent advances in text-to-image diffusion models have enabled the generation of diverse and high-quality images. While impressive, the images often fall short of depicting subtle details and are susceptible to errors due to ambiguity in the input text. One way of alleviating these issues is to train diffusion models on class-labeled datasets. This approach has two disadvantages: (i) supervised datasets are generally small compared to large-scale scraped text-image datasets on which text-to-image models are trained, affecting the quality and diversity of the generated images, or (ii) the input is a hard-coded label, as opposed to free-form text, limiting the control over the generated images.

In this work, we propose a non-invasive fine-tuning technique that capitalizes on the expressive potential of free-form text while achieving high accuracy through discriminative signals from a pretrained classifier. This is done by iteratively modifying the embedding of an added input token of a text-to-image diffusion model, by steering generated images toward a given target class according to a classifier. Our method is fast compared to prior fine-tuning methods and does not require a collection of in-class images or retraining of a noise-tolerant classifier. We evaluate our method extensively, showing that the generated images are: (i) more accurate and of higher quality than standard diffusion models, (ii) can be used to augment training data in a low-resource setting, and (iii) reveal information about the data used to train the guiding classifier. The code is available at `https://github.com/idansc/discriminative_class_tokens`.

1. Introduction

Text-to-image diffusion models [37, 10] have shown remarkable success in generating diverse and high-quality images conditioned on input text. However, they often fall short when the input text contains lexical ambiguity or when generating fine-grained details. For instance, one might wish to render an image of a clothes 'iron', but could instead be presented with an image of the elemental metal.

To alleviate these issues, pretrained classifiers have been used to guide the denoising process. One such method mixes the score estimate of a diffusion model with the gradient of the log probability of a pre-trained classifier [10]. However, this approach has the downside of requiring a classifier that works on both real and noisy data. Others have conditioned the diffusion on class labels using a curated dataset [24] . While effective, this approach does not lead to the full expressive power of models trained on huge collections of web-scale image-text pairs.

A different line of work fine-tunes a diffusion model, or some of its input tokens, using a small (~5) collection of images [15, 26, 38]. These methods have the following drawbacks: (i) training new concepts can be slow, taking upwards of a few hours, (ii) the method may change the distribution of the generated images (as compared to the original diffusion model) to fit only the new label or concept, and (iii) the generated images are based on features from a small group of images and may not capture the diversity of the entire class.

This work introduces a method that more accurately captures the desired class, avoiding lexical ambiguity while more accurately portraying fine-grained details. It does so while retaining the full expressive power of the original pretrained diffusion model without the above-mentioned drawbacks. Instead of guiding the diffusion process or updating the entire model with the classifier, we only update the representation of a single added token, corresponding to each class of interest. We do this without tuning the model on labeled images. To learn the token representation corresponding to a given target class, we iteratively generate new images with a higher class probability according to the pretrained classifier. At each iteration, feedback from the classifier steers the designated discriminative class token to this end. Our optimization process uses a new technique, *gradient skipping*, which only propagates the gradient through the final stage of the diffusion process. The optimized token is then used as part of the conditioning text-input to generate images using the original diffusion model.

[*] Equal contribution.

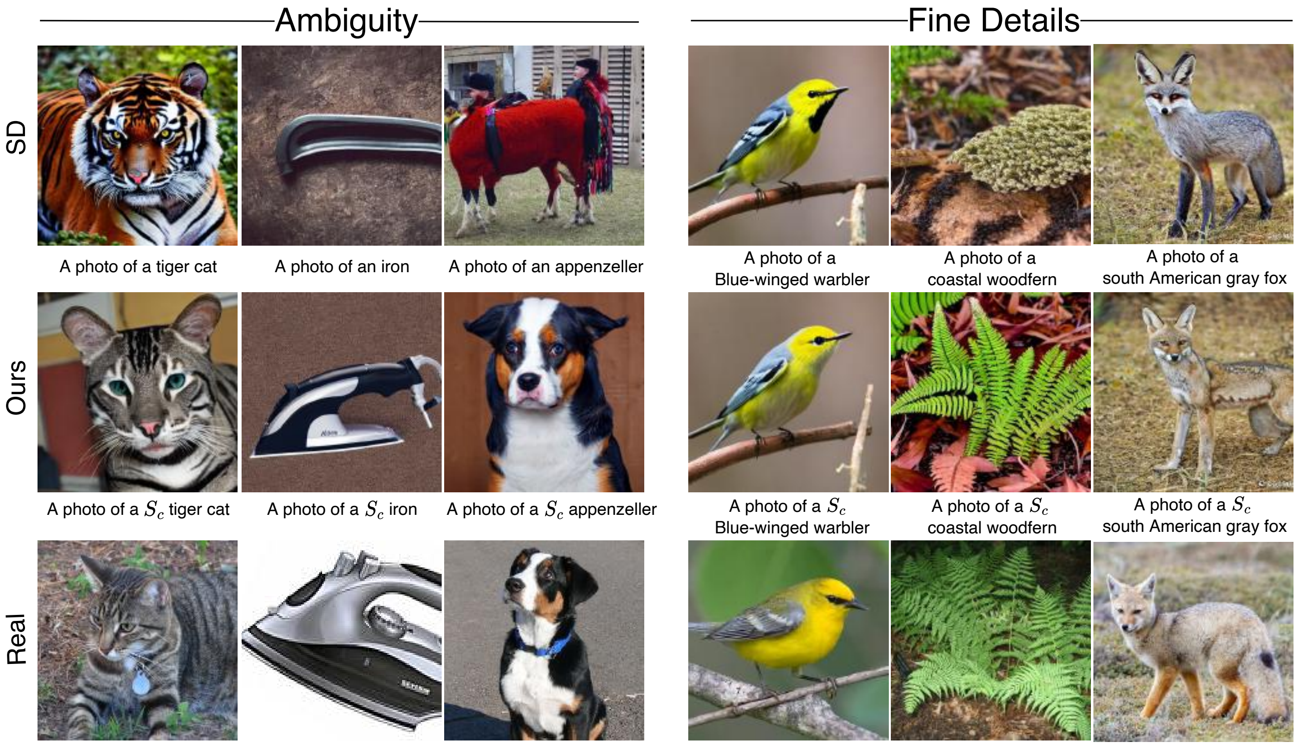
Ambiguity
Fine Details
SD
Ours
Real
A photo of a tiger cat
A photo of an iron
A photo of an appenzeller
A photo of a Blue-winged warbler
A photo of a coastal woodfern
A photo of a south American gray fox
A photo of a S_c tiger cat
A photo of a S_c iron
A photo of a S_c appenzeller
A photo of a S_c Blue-winged warbler
A photo of a S_c coastal woodfern
A photo of a S_c south American gray fox

Figure 1: We propose a technique that introduces a token (S_c) corresponding to an external classifier label class c. By including the token in the input text we can improve both text-to-image alignment when there is lexical ambiguity (left) and enhance the depiction of intricate details (right).

Our method has several advantages. First, unlike other class conditional methods such as [10], it only requires an off-the-shelf classifier and does not require a classifier trained on noisy data. Second, our method is fast and allows for "plug-and-play" improvements of generated images after a class-token has been trained. This is in comparison to other methods, such as Textual Inversion [15], which can take a few hours to converge.

Third, our method employs a classifier trained on an extensive collection of images without needing access to those images. This is beneficial as (i) the token is trained using the full set of class-discriminative features, as opposed to features from a small set of images, and (ii) it may be desirable to share only the classifier and not the data on which it is trained, such as when privacy concerns are involved.

We evaluate our method both in the fine-grained and coarse-grained settings. In the fine-grained setting, we investigate the ability of our method to generate details of species in the CUB [42] and iNat21 [40] datasets. In the coarse setting, we consider the ImageNet [9] dataset. Our primary metric is the accuracy of the generated samples as measured in two ways: (i) we show that our generated images are more often correctly classified using pre-trained classifiers, in comparison to baselines, and (ii) we show that classification models trained on generated samples, either on their own or in combination with a limited amount of real training data, result in improved accuracy compared to the baseline. We also measure the quality and diversity of the generated images compared to SD and a different class-conditioning technique, showing that our method is superior in terms of the commonly used Fréchet inception distance (FID) [21]. Finally, we include many qualitative examples demonstrating the effectiveness of our approach. In Fig. 1, we show how our method can resolve ambiguity in the input text and add discriminative features for a given class. In the ambiguous category, the image of a *tiger cat* becomes the cat species instead of a tiger, *iron* is made to refer to the tool as opposed to the metal. *Appenzeller* moves from depicting a group of people, from the Appenzeller area, to the dog species. In the fine-grained category, the *Blue winged warbler's* throat color is corrected, the shape features of the *Coastal woodfern* are corrected, and the *American gray fox* more closely resembles the true species.

2. Related work

The field of text-based image generation has been studied extensively, both for GANs [18] and, more recently, for diffusion models [11, 22, 41, 27, 28, 32, 33, 36, 45, 8, 14, 37, 25]. The use of diffusion models has, in particular, enabled an unprecedented capability in generating high-quality diverse images from natural language input with models such as DALL·E 2 [35], Imagen [39], Parti [44], Make-A-Scene [14], Stable Diffusion (SD) [37], and CogView2 [12].

A recent line of work extends models of this kind by tuning the input embeddings to personalize image generation. In particular, some contributions generate images based on a small group of images: Textual inversion (TI) [15] optimizes the embedding of a new textual token that represents a concept found in a handful of images. DreamBooth [38] proposes fine-tuning of the *full* image generation model where a unique identifier represents the concept. Both works require 3-5 training images to learn the identity of the target concept. A related line of work enables editing of a given image based on input text or another image [31, 16, 7, 3, 2, 4].

More recently, some have suggested methods to leverage large text-based image generators for image editing. Prompt-to-prompt [20] edits the input prompt directly via manipulation of cross-attention maps, and Imagic [26] optimizes the corresponding textual prompt and fine-tunes the model such that the image is accurately reconstructed. When only a few images are used for training, though, there is always the inherent risk that a concept can become too similar to the original images.

In contrast, we aim to steer existing diffusion models toward a more general understanding of classes via the characteristics needed to discriminate between them, while still taking full advantage of the diversity of the underlying generative model. Our method is also faster than image-based training methods such as TI and can effectively make use of an off-the-shelf classifier to train the token needed to refine an image within minutes.

Manipulating an image using classifier conditioning can also provide a counterfactual explanation for classifiers [6, 17, 1]. In that sense, our method may also be used to reveal salient visual features of the classifer. Since semantic differences are relatively small during each iteration of the image generation, they can be detected during the process.

3. Method

We now describe how the discriminative token embeddings are learned. We first introduce conditional diffusion models in general, including *classifier guidance* (not to be confused with our method), and then describe our conditioning approach and the gradient skipping. An overview of our method is provided in Fig. 2.

Conditional diffusion models Diffusion models [23, 10] estimate a process that generates data $x \sim p(x)$ from randomly sampled noise. During training, an iterative denoising process predicts step-wise added noise $x_T \sim \mathcal{N}(0, \mathrm{I})$. More specifically, given an input image (or a latent encoding) $x_0 \sim p(x)$, one first produces samples $x_t = \sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t} \cdot \epsilon_t$, with $0 < \alpha_T < \alpha_{T-1} < \cdots < \alpha_0 = 1$ being hyperparameters and $\epsilon_t \sim \mathcal{N}(0, \mathrm{I})$. A neural network is then trained to predict the added noise ϵ_t via the objective function:

$$\mathbb{E}_{x,\epsilon_t,t}[||\hat{\epsilon}_\theta(x_t, t) - \epsilon_t||_2^2], \tag{1}$$

A conditional denoising process, where each denoising step depends on a conditioning input (e.g., a class identifier or a text prompt y), can be defined similarly:

$$\mathbb{E}_{x,\epsilon_t,t}[||\hat{\epsilon}_\theta(x_t, t, y) - \epsilon_t||_2^2], \tag{2}$$

To condition the diffusion process on a class, gradients obtained with trained classifiers can be used in the denoising process [10]. In particular, gradients of contrastive image-text models, like CLIP [34], have been used for text conditioning. Utilizing classifier guidance improves the sample quality and enables a trade-off between sample quality and diversity.

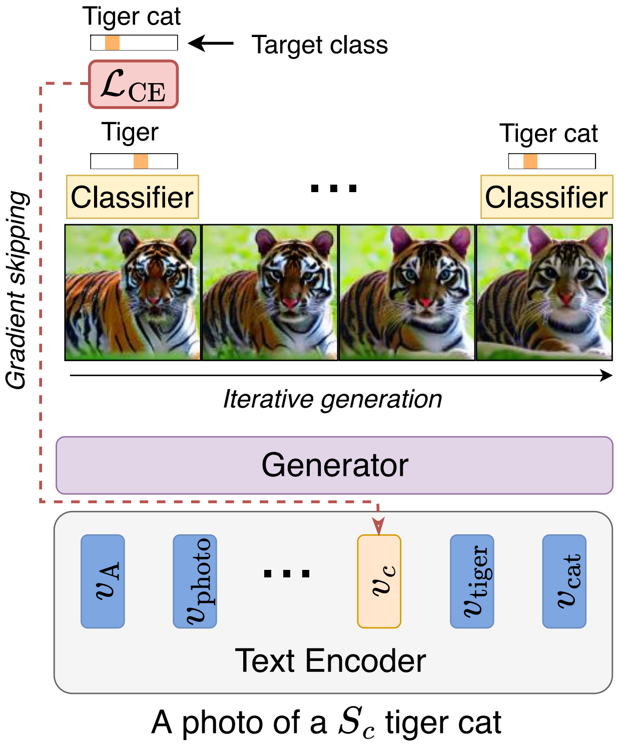

Figure 2: An overview of our method for optimizing a new discriminative token representation (v_c) using a pre-trained classifier. For the prompt 'A photo of a S_c tiger cat,' we expect the output generated with the class c to be 'tiger cat'. The classifier, however, indicates that the class of the generated image is 'tiger'. We generate images iteratively and optimize the token representation using cross-entropy. Once v_c has been trained, more images of the target class can be generated by including it in the context of the input text.

There are two main drawbacks to using classifier guidance within the diffusion process: (i) the classifier must be retrained to deal with noised images as every noisy sample generated along the iterative denoising process must be passed through the classifier, and (ii) the classifier needs to be present throughout the generative process. To mitigate these issues, a *classifier-free* approach has been proposed [10]. Instead of relying on gradients from an image classifier, this approach approximates the gradient of an implicit classifier by modeling the difference between conditional, $p_\theta(x|y)$, and unconditional, $p_\theta(x)$, denoising modules. The conditional and unconditional modules are parameterized using the same noise predicting model $\epsilon_\theta(x_t, y)$. The conditional network then becomes unconditional by using an empty input, i.e., $\epsilon_\theta(x_t) = \epsilon_\theta(x_t.\text{``''})$. The final denoising network is formally expressed as follows:

$$\bar{\epsilon}_\theta(x, x_t, y) = (1+w)\epsilon_\theta(y_t, y) - w(\epsilon_\theta(x_t)), \qquad (3)$$

where w is a hyperparameter determining the strength of the conditioning guidance.

Our method is complementary to both the *classifier-based* and *classifier-free guidance*, and can be used in conjunction with both. While our method can be deployed using any diffusion model, here we consider Stable Diffusion (SD) [37]. In SD the denoising process is applied not directly to the pixel values of the images, but in the lower dimensional latent dimensions of a neural network.

Discriminative Token Embeddings Classifiers capture discriminative signals needed to discern between classes. In that sense, pretrained classifiers can be seen as experts models in different domains. For example, a bird classifier can provide a compact source of discriminative details that separate one species from another.

To avoid relying on a classifier at inference time, and reduce the need to fine-tune the classifier on noised images (as in earlier work), our method fine-tunes a token that is added to the input vocabulary. Our technique iteratively generates images and refines this added token (as opposed to a word or subword found in the original embedding matrix) to associate the generated images with a target class of the pre-trained classifier. The only weights being updated are those of the new class token.

The process starts with a discriminative class token S_c and a generic prompt p = "A photo of a S_c *label*", where *label* is the English name of the class. We include the class name as part of the prompt to take advantage of existing knowledge in the pre-trained diffusion model (but only update the embedding for S_c during training). By looking at the images, e.g., in Fig. 1 it is evident that the configuration has gained knowledge across various domains, including expertise in generating specific bird species, though with some limitations. Our approach aims to enhance the precision of the generated image by introducing minor semantic modifications that leverage the model's existing knowledge.

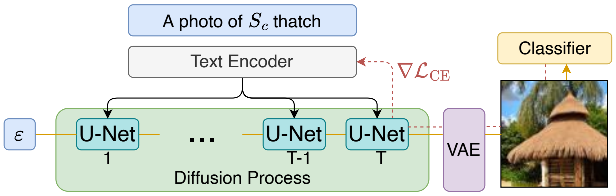

Figure 3: An illustration of the gradient skipping technique (indicated by the red line). During backpropagation, the gradient is propagated only through the final denoising step of the diffusion procedure.

We now describe how we associate the characteristics of the target class with the added token's representation. We denote the embedding of the class token S_c as v_c, and learn it by utilizing an image classifier C. To speed up training, v_c is initialized as the embedding of a related token. For instance, in the case of a bird classifier, we initialize the embedding to that of the 'bird' token in the input encoder. For more general classifiers, such as those trained on ImageNet, we use the indefinite article token 'a' as a base for more generalized concepts. Training starts by generating an image $x(p)$ conditioned on p, which includes the v_c representation. We feed the resulting image into the classifier and use cross-entropy loss over the classifier labels, i.e.,

$$\min_{v_c} \text{CE}\left(C\left(\psi_C(x(p))\right), \mathbb{1}_c\right), \qquad (4)$$

where ψ_C transforms the image to align it with the classifier's expected input (e.g., resizing), and $\mathbb{1}_c$ is one hot vector of the target class. To summarize, our method does not rely on a given set of images. Instead, it generates images iteratively, starting with the output from SD, where each optimization step shifts the generated images closer to the target class distribution by updating the class token. A single image can be optimized directly by training a token that generally converges in a relatively small number of steps.

Gradient skipping Fig. 3 illustrates the generation of a single image using a diffusion process and the flow of the learning signal. Propagating gradients through all diffusion steps requires a significant amount of memory. In our experiments, propagating gradients solely through the final denoising step (i.e., step T) produces high-quality images representing the intended class while requiring fewer resources. While deeper backpropagation could lead to further enhancements, we do not explore this direction further due to memory constraints.

Dataset	Classifier	Guidance	Top-1	Top-5
ImageNet	ImageNet	-	70.5	89.7
ImageNet	ImageNet	ImageNet	**74.5**	**92.6**
CUB	CUB	-	39.7	73.3
CUB	CUB	CUB	**57.9**	**88.6**
iNat179	iNat	-	28.5	56.5
iNat179	iNat	CUB	**32.8**	**63.5**
iNat50	iNat	-	14.1	28.7
iNat50	iNat	iNat	**25.8**	**50.3**

Table 1: Classification results on four datasets: ImageNet, CUB, and two subsets from iNaturalist21: iNat50, and iNat179. For each dataset, we calculate the accuracy using a classifier (Classifier) on images generated with or without the guidance of another classifier (Guidance). For all datasets, accuracy is higher when guidance is used, including in the case (iNat179) where the classifier used (iNat) differs from that used for guidance (CUB).

Design Choices Our approach involves several design choices. (i) *Batch size*: Our goal is not to refine a single image but to find a broad token representation that can generate new images without incurring extra costs. By generating images with different seeds, we get diverse images. A larger batch size picks up more generic discriminative features, but training takes slightly longer to converge. We set the batch size to 5 after experimenting with values of 1-6. More details and examples of generations are shown in the supplementary. (ii) *Number of prompts*: Increasing the number of prompts can introduce additional variability. However, we find too much variability during training harmful to convergence. Thus, we limited the number of prompts used in the training phase to two: p_1 ="A high-resolution realistic image of a S_c *label*", and p_2 ="A photo of S_c *label*". It is worth noting that one can still utilize the discriminative token across various prompts, as shown in Fig. 7. (iii) *Updated tokens*: Our experiments focus on optimizing only the embedding of S_c. While it is possible to update other pre-existing tokens, doing so would modify the model and prevent it e.g. from being used in case of lexical overlap between classes, such as in the case of mouse as depicted in Fig. 11. (iv) *Early stopping strategy*: Please refer to the supplementary for additional details.

4. Results

Quantitatively, we evaluate the ability of our method to conform to the input class and to generate high-fidelity images. For the former, we consider the following: (i) taking pre-trained classifiers and assessing the classification accuracy of generated images, and (ii) evaluating the accuracy of classifiers trained using generated and real images. This assesses the generated images in two complementary ways. If the pre-trained classifier correctly classifies generated images, then they capture features from the correct class. If the generated images can be used to improve a classifier's performance (over the same evaluation set), then they capture a broad range of additional discriminative features that can improve classification accuracy. To evaluate the quality of generated images, we consider the commonly used FID score [21] and KID score [5]. Finally, we consider the memory footprint and the speed of our method. Qualitatively, we demonstrate the effectiveness of our approach in adding fine-grained details and resolving problematic cases of ambiguity. Please refer to the supplementary for a full description of the experimental setup, including implementation and training details. We note that all reported SD baselines are with classifier-free guidance using default parameters.

	# Real img.	Baseline		SD		Ours (CUB)	
	per class	T-1	T-5	T-1	T-5	T-1	T-5
CUB	0	0.0	0.0	37.1	67.8	**48.6**	**78.5**
	3	1.8	5.1	52.6	84.9	**62.5**	**87.9**
	9	35.1	72.3	68.3	92.7	**76.3**	**95.1**
	15	71.5	94.1	77.3	95.8	**81.4**	**96.4**
iNat179	0	0.0	0.0	22.1	47.7	**28.1**	**55.2**
	3	1.3	5.7	31.6	61.1	**37.1**	**66.9**
	9	8.0	27.3	43.0	74.3	**49.5**	**78.3**
	15	28.7	62.9	49.8	81.4	**58.3**	**84.1**

Table 2: Results for classifiers trained with 100 generated and 0-15 real images (# Real img.) from each CUB class, and in the overlap of 179 species in iNat. For the baseline column, we use only real images. Results with our method use **CUB guidance** and improve performance even for the different iNat179 dataset.

Method	FID ↓	KID ↓
Text-conditioned (SD v1.4)	23.0	0.00398
Text-conditioned (SD v2.1)	15.7	0.00858
Class Conditioned Methods		
Class-conditioned [37]	47.6	0.0115
Ours (SD v1.4)	22.4	**0.00364**
Ours (SD v2.1)	**14.7**	0.00775

Table 3: FID and KID scores for generated ImageNet classes. For SD and our method, we consider two versions of the underlying SD model (1.4/2.1).

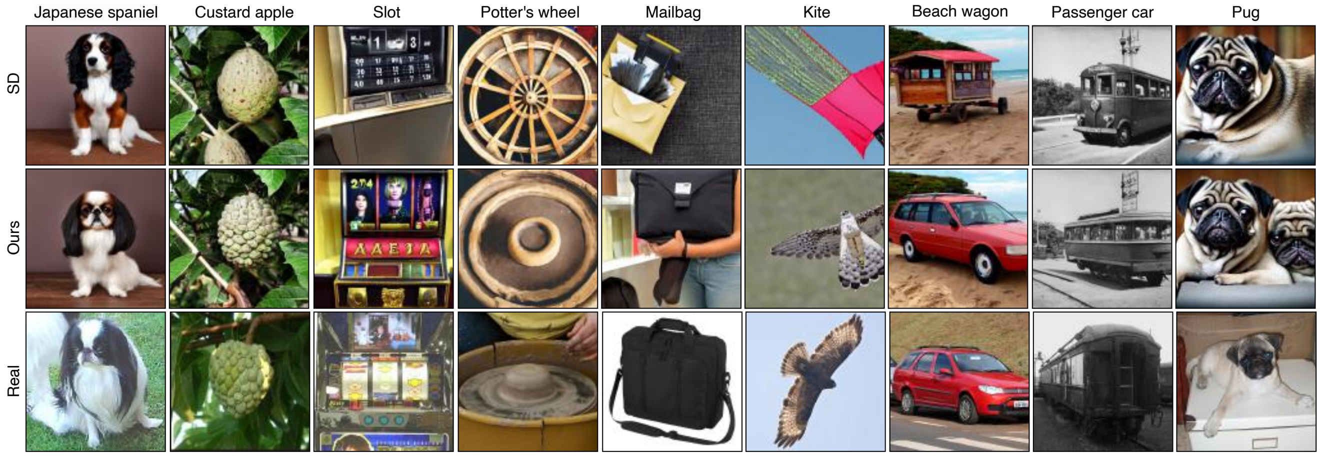

Figure 4: Images generated based on ImageNet classes, using SD or our method. Real images are shown for comparison.

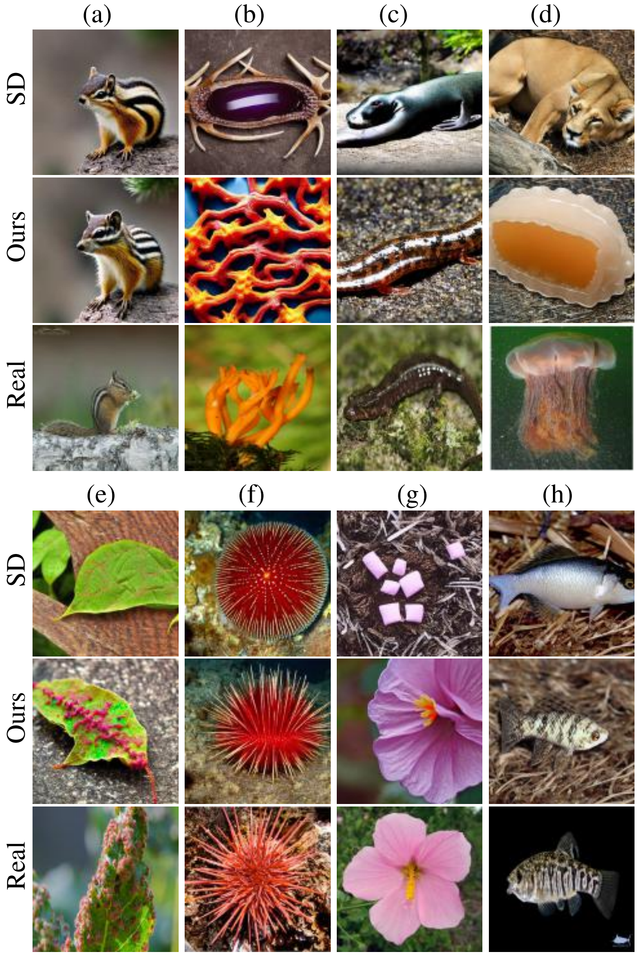

Figure 5: A selection of images based on iNat classes generated with Stable Diffusion (SD) and our method. A real image is shown for comparison. (a) Yellow pine chipmunk, (b) Jelly antler, (c) Salamander, (d) Pacific lions mane jelly, (e) Leaf mite, (f) Red sea urchin, (g) Seashore mallow, (h) Sheepshead minnow.

Datasets For *fine-grained* categories, such as those of birds and animal species, we use the CUB dataset [42] and the iNaturalist 2021-mini (iNat from here on) dataset [40]. We also consider two subsets of iNat: (a) iNat179, a subset that overlaps with CUB in 179 out of the 200 labels and allows us to classify images generated when utilizing guidance from a different classifier (in particular, one trained on a different dataset), (b) iNat50, a sample of 50 species randomly selected from each of the 11 supercategories. In the *course-grained* setting, we consider the ImageNet dataset [9].

4.1. Quantitative evaluation

Evaluation using pre-trained classifiers We first evaluate the generated images with classifiers trained on real data. We generate 100 images for each class and calculate the accuracy of each method: one employing our class token and another with only SD. In Tab. 1, we show that for ImageNet, which mainly consists of classes at a coarse level of granularity, the vanilla SD can generate most classes accurately (70.5%). By utilizing class token guidance, we get better results in complex cases, such as those with ambiguity, resulting in an improved accuracy of 74.5%. For our method and SD, the seed and textual context were held constant, so the images correspond to each other, with the differences being due to the use of the token. As a further comparison, we consider SD model v2.1 (as opposed to the default v1.4 but with the same hyperparameter configuration). Here our method achieves 57.1% accuracy in comparison to SD's 54.6%. We also evaluate our method using different classifier architectures. We test ViT-L [13], Resnet-50 [19], Swin-v2 [29], ConvNext-L [43], reaching major improvements for ImageNet1k (**72.34%/70.93%/76.45%/73.80%** vs 39.74%/39.42%/42.19%/40.83% for SD).

We next assess fine-grained classes. Testing the model on the images generated with labels found in the CUB dataset, the accuracy of SD-generated images is only

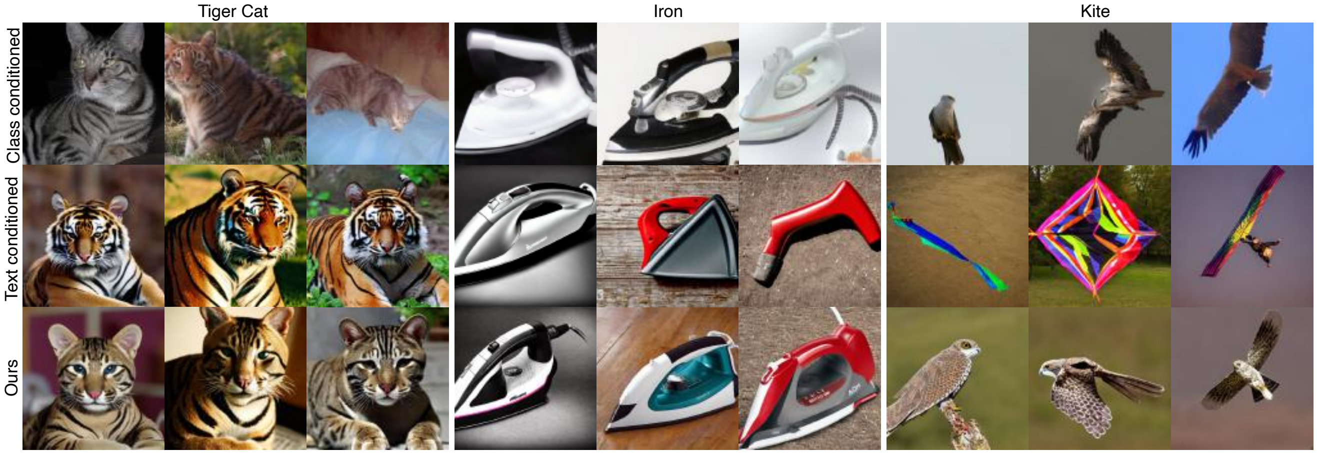

Figure 6: Images generated using class-conditioned LDM, text-conditioning (SD), and our method with ImageNet classifier guidance.

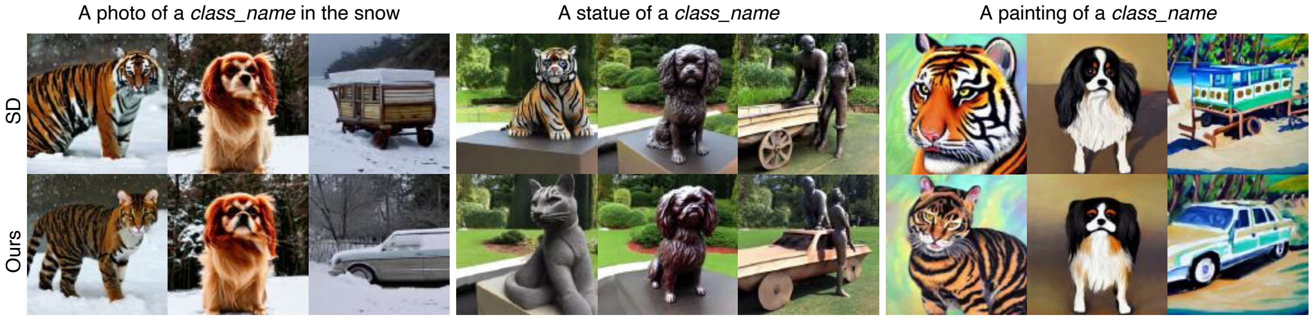
A photo of a *class_name* in the snow
A statue of a *class_name*
A painting of a *class_name*
SD
Ours

Figure 7: Results with different prompts for three classes: (i) tiger cat, (ii) Japanese spaniel, and (iii) beach wagon.

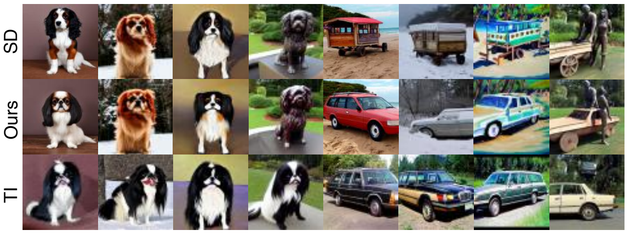

Figure 8: Comparison of our method to Textual Inversion (TI). TI-generated images often lack diversity and are prone to incorporating background features from a limited set of images. For example, in the Japanese spaniel class, TI generates a black-and-white dog due to limited color variations in the given images. In the beach wagon class, TI includes the road as part of the object, changing the background to a road from a beach. For reference, we also display SD's generated samples.

39.7%. This emphasizes the inherent limitations of the SD model in generating highly detailed and specific categories, such as bird species. Our approach adds the fine details necessary to improve accuracy to 57.9%. As a further comparison, we consider SD model v2.1 (as opposed to the default v1.4). Here our method achieves 47.7% accuracy in comparison to SD's 44.0%. We also assess the accuracy of the generated images using the iNat classifier (trained on 10k species) on the same image generated with the guidance of the CUB classifier. Our approach yields a noteworthy improvement in performance from 28.5% to 32.8%, indicating its potential to enhance fine-grained classes beyond the specific classifier-selected characteristics. Finally, we look at a diverse set of 50 classes from the iNat dataset. SD only generates images that are classified accurately 14.1% of the time. Our method significantly improves accuracy to 25.8%, while still leaving ample room for further improvement.

Evaluation by training classifiers In Tab. 2, we show results for training classifiers on 100 generated images and 0-15 real images. When we incorporate the classifier-guided generated images, the real evaluation accuracy is better than when augmenting only with SD-generated images. With only nine real images per class, we already reach 76.3% accuracy compared to 35.1% with no generated images and

68.3% with SD-only augmentations. A high accuracy indicates that the generated images capture a large part of the distribution necessary to classify images correctly. Our findings also show that utilizing the CUB classifier for generating images can enhance performance when evaluated in the iNat179 setting. Specifically, incorporating our proposed image augmentation method improves accuracy, reaching 58.3% with only 15 real images, compared to the significantly lower accuracy of 49.8% and 28% with SD or when no augmentation is applied. These results indicate that our approach shows potential for augmenting data in low-resource settings by transferring knowledge from diverse classifiers.

FID and KID evaluation Class-conditioned image generation models may enjoy the benefit of eliminating ambiguity in generated images. However, datasets used to train these models are limited only to specific classes, and do not capture the wide variety of images depicting free-form text. In Tab. 3, we make use of FID [21] and KID [5] scores to assess the quality of the generated images with respect to the real datasets. Our evaluation shows that the text-conditioned method generates higher-quality images compared to a prior class-conditioned method. The text-conditioned method of SD, on the other hand, is limited by ambiguity issues and has difficulty in depicting fine details. Our proposed method provides a balance between generating high-quality images accurately and avoiding ambiguity issues.

Memory Requirements and Speed Our memory footprint is low, due to our gradient skipping method. For training and inference, a single commercial GPU with 10GB of VRAM can be used. This is comparable to Textual Inversion (TI). At inference, the same memory requirements are needed as for classifier-free guidance, TI, and SD.

In terms of speed, our method takes ∼10 minutes to train on a single GPU. For TI, ∼1 hour is required. To capture additional class diversity with TI, one must train on a full class-representative set of images (∼1000 for ImageNet), with ∼5 images (1 hour) at a time, requiring days at best. Further, our method doesn't require retraining an expensive diffusion model (e.g. 256 A100 for two weeks as SD). Inference time is the same as for SD/TI and takes about 0.88s per sample using our configuration.

4.2. Qualitative assessment

In Fig. 4, we show various generated samples based on ImageNet classes. From left to right, our method reinforces distinctive features of the dog species, in particular, the face. The distinct characteristics of the custard apple and slot images are highlighted with our method. For the mailbag, the 'mail' term appears to confuse SD to generate a mail-related image rather than a mailbag. More ambiguities arise from the term wheel in potter's wheel and the kite class. In some cases, we only partially resolve ambiguity. For instance, beach wagon cars still appear on the beach, the bird kite resembles a toy kite. Another interesting case is when the method adds another instance of the target class object, as is the case in the pug image.

In Fig. 5, we show images generated using labels from the iNat dataset. We sample 50 species from each and compare them using SD and our method. Our method corrects for attributes such as patterns (e.g., (a), and (h)), anatomical issues (e.g., (b), (c), (f)), and resolves lexical ambiguity (e.g., (b), (d), (g)).

In Fig. 6, we show that only class-conditioned images appear less natural despite being highly relevant to the class. Our approach allows us to benefit from the advantages of both worlds, producing high-quality images that are both precise and devoid of ambiguity.

Another advantage of our approach over simple class conditioning [37] is the flexibility to use trained tokens with various prompts. In Fig. 7, we demonstrate that our discriminative tokens can be employed in different prompts, resulting in minimal semantic changes that primarily affect the object of interest that is relevant to the class.

Textual Inversion Textual Inversion (TI) employs an optimization technique whereby embeddings are found to represent a given concept captured by a small set of 5 images. As can be seen in Figure 8, TI's generated images can lack diversity and are prone to incorporating background features from a limited set of images. For example, in the Japanese spaniel class, TI generates a black-and-white dog due to limited color variations in the given images. In the beach wagon class, TI includes the road as part of the object, preventing it from appearing in the snow.

Face attributes Our method is not only capable of enhancing objects and animals. We show that a classifier based on CelebFaces attributes [30] can be used to learn a token representing a facial attribute. We generate a facial image using the prompt "An image of a S_c person's face.". We optimize S_c using a classifier consisting of six convolutional layers followed by two fully connected layers. In Fig. 9, we present our results obtained by training with the guidance of baldness and gender attributes. During training, we observed that the hair feature is more dominant for the 'not bald' class. For the 'bald' class, the generated image depicts old men and their identity is lost. This suggests that age may be a hidden factor in the training data.

Classifier inversion Our method has the ability to inverse the action of a classifier without access to its trained data. For example, we often observe changes in the background

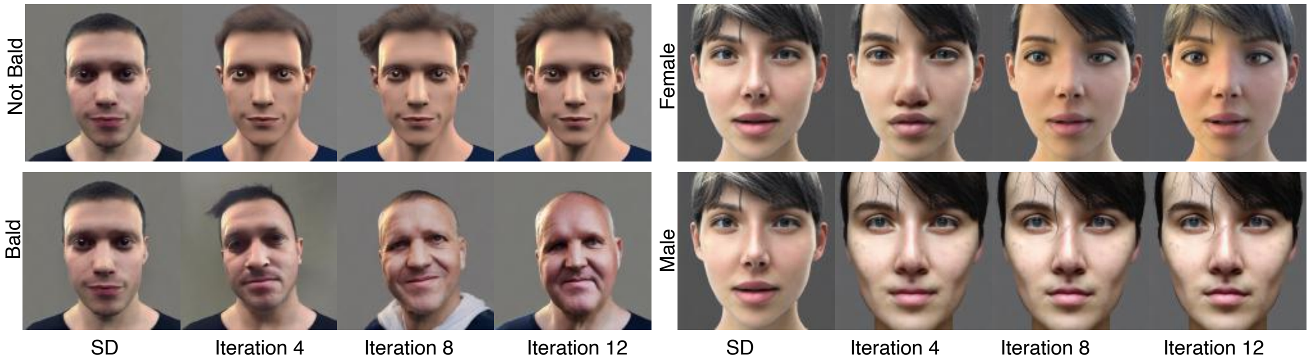

Figure 9: Demonstration of using discriminative tokens for gender and bald attributes, ‘SD’ shows the initial generation.

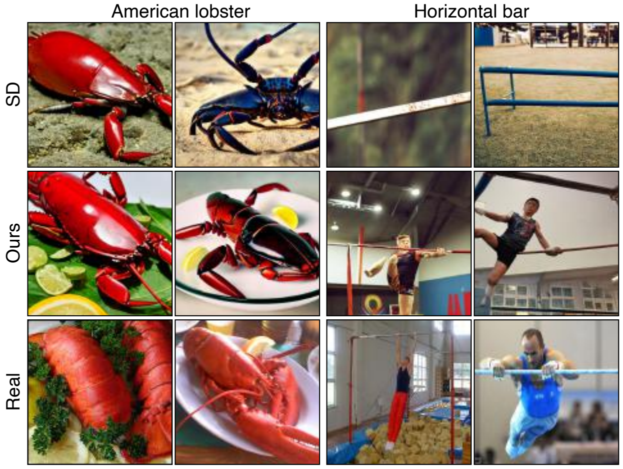

Figure 10: Examples of revealed features from the training data when using an ImageNet classifier for guidance.

when optimizing for an object’s class. As Fig. 10 shows, applying our method with an ImageNet-trained classifier results in an image of a lobster on a plate given the phrase ‘American lobster’. Another example is the ‘horizontal bar’ class, for which our method predominantly generates images containing athletes and a gym environment. We manually assessed ImageNet’s training data by classifying 100 images from the ‘American lobster’ and ‘horizontal bar’ classes and determining whether they exhibit these features in the training data. For the ‘American lobster’ class, 55% of the images featured a plate and the lobster in an edible form, and for the ‘horizontal bar’ class, 95% of the images included an athlete performer. In Fig. 10, we present some instances from the training data that illustrate these characteristics. Nevertheless, interpreting the results needs to be done with caution. It is possible that this bias toward a certain type of image reflects one local minimum in our optimization process.

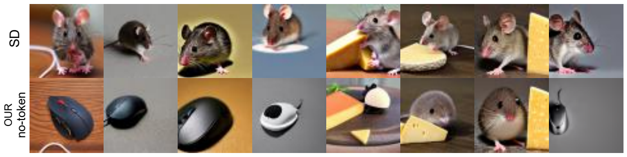

Figure 11: An advantage of employing a special token over directly modifying original tokens. Updating the ‘mouse’ token directly would mistakenly associate attributes with all instances of ‘mouse,’ including the animal with ‘computer mouse.’

Ablation To illustrate the benefit of using a special token, instead of directly updating the original tokens, we ran a simple experiment where the ‘mouse’ token is updated using the class when used in the sense of ‘computer mouse’. This can prevent the model from generating the animal ‘mouse’ when using that token, as shown in Figure 11.

5. Conclusion

In this paper, we introduced a “plug-and-play” approach for rapidly fine-tuning text-to-image diffusion models by using a discriminative signal. Our approach trains a new token without additional images, enhancing fine-grained details for classifiers pre-trained on datasets such as CUB and iNat and resolving lexical ambiguity. We have also demonstrated how our method can be used to distill generative image models to supplement datasets lacking imagery, edit faces based on attributes classifier, and analyze hidden factors in the training data. Going forward, we aim to extend our approach to other model types beyond classification.

6. Acknowledgements

This project was supported by a grant from the Tel Aviv University Center for AI and Data Science (TAD). VS, SB, and SB are supported by the Pioneer Centre for AI, DNRF grant number P1.

References

[1] Maximilian Augustin, Valentyn Boreiko, Francesco Croce, and Matthias Hein. Diffusion Visual Counterfactual Explanations. *NeurIPS*, 2022. 3

[2] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Trans. Graph.*, 42(4), jul 2023. 3

[3] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022. 3

[4] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2LIVE: Text-Driven Layered Image and Video Editing. In *Computer Vision – ECCV 2022*, pages 707–723, Cham, 2022. Springer Nature Switzerland. 3

[5] Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. 5, 8

[6] Valentyn Boreiko, Maximilian Augustin, Francesco Croce, Philipp Berens, and Matthias Hein. Sparse visual counterfactual explanations in image space. In *DAGM German Conference on Pattern Recognition*, pages 133–148. Springer, 2022. 3

[7] Hila Chefer, Sagie Benaim, Roni Paiss, and Lior Wolf. Image-Based CLIP-Guided Essence Transfer. In *Computer Vision – ECCV 2022*, pages 695–711, Cham, 2022. Springer Nature Switzerland. 3

[8] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. VQGAN-CLIP: Open Domain Image Generation and Editing with Natural Language Guidance. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 88–105, Cham, 2022. Springer Nature Switzerland. 2

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 6

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 1, 2, 3, 4

[11] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021. 2

[12] Ming Ding, Wendi Zheng, Wenyi Hong, and Jie Tang. Cogview2: Faster and better text-to-image generation via hierarchical transformers. *arXiv preprint arXiv:2204.14217*, 2022. 3

[13] 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. *ICLR*, 2021. 6

[14] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-A-Scene: Scene-Based Text-to-Image Generation with Human Priors. In *Computer Vision – ECCV 2022*, pages 89–106, Cham, 2022. Springer Nature Switzerland. 2, 3

[15] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2, 3

[16] Rinon Gal, Or Patashnik, Haggai Maron, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators. *ACM Trans. Graph.*, 41(4), jul 2022. 3

[17] Itai Gat, Guy Lorberbom, Idan Schwartz, and Tamir Hazan. Latent space explanation by intervention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 679–687, 2022. 3

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

[19] 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

[20] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-Prompt Image Editing with Cross-Attention Control. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[21] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2, 5, 8

[22] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Semantic object accuracy for generative text-to-image synthesis. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 3

[24] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 1

[25] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 867–876, 2022. 2

[26] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-Based Real Image Editing with Diffusion Mod-

els. In *Conference on Computer Vision and Pattern Recognition 2023*, 2023. 1, 3

[27] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. Controllable text-to-image generation. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[28] Wenbo Li, Pengchuan Zhang, Lei Zhang, Qiuyuan Huang, Xiaodong He, Siwei Lyu, and Jianfeng Gao. Object-driven text-to-image synthesis via adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12174–12182, 2019. 2

[29] 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*, pages 10012–10022, 2021. 6

[30] Z Liu, P Luo, X Wang, and X Tang. Deep learning face attributes in the wild. arxiv. *ICCV*, 2015. 8

[31] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2085–2094, October 2021. 3

[32] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Learn, imagine and create: Text-to-image generation from prior knowledge. *Advances in neural information processing systems*, 32, 2019. 2

[33] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Mirrorgan: Learning text-to-image generation by redescription. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1505–1514, 2019. 2

[34] 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 Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. 3

[35] 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. 3

[36] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 2

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022. 1, 2, 3, 4, 5, 8

[38] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 3

[39] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 3

[40] Jong-Chyi Su and Subhransu Maji. The semi-supervised inaturalist challenge at the FGVC8 workshop, 2021. 2, 6

[41] Ming Tao, Hao Tang, Fei Wu, Xiao-Yuan Jing, Bing-Kun Bao, and Changsheng Xu. DF-GAN: A Simple and Effective Baseline for Text-to-Image Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16515–16525, 2022. 2

[42] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 6

[43] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16133–16142, 2023. 6

[44] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling Autoregressive Models for Content-Rich Text-to-Image Generation. *Transactions on Machine Learning Research*, 2022. Featured Certification. 3

[45] Zizhao Zhang, Yuanpu Xie, and Lin Yang. Photographic text-to-image synthesis with a hierarchically-nested adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6199–6208, 2018. 2