Fake it till you make it: Learning(s) from a synthetic ImageNet clone

Mert Bulent Sariyildiz, Karteek Alahari, Diane Larlus, Yannis Kalantidis

Fake it till you make it: Learning(s) from a synthetic ImageNet clone

Mert Bulent Sariyildiz[1,2] Karteek Alahari[2] Diane Larlus[1] Yannis Kalantidis[1]

[1] NAVER LABS Europe [2] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK

Abstract

Recent large-scale image generation models such as Stable Diffusion have exhibited an impressive ability to generate fairly realistic images starting from a very simple text prompt. Could such models render real images obsolete for training image prediction models? In this paper, we answer part of this provocative question by questioning the need for real images when training models for ImageNet classification. More precisely, provided only with the class names that have been used to build the dataset, we explore the ability of Stable Diffusion to generate synthetic clones of ImageNet and measure how useful they are for training classification models from scratch. We show that with minimal and class-agnostic prompt engineering those ImageNet clones we denote as ImageNet-SD are able to close a large part of the gap between models produced by synthetic images and models trained with real images for the several standard classification benchmarks that we consider in this study. More importantly, we show that models trained on synthetic images exhibit strong generalization properties and perform on par with models trained on real data.

1. Introduction

The rise of (shallow) machine learning [16, 63, 101] and later deep learning [29, 32, 51, 96] has entirely changed the landscape of computer vision research over the past few decades, shifting some of the focus from *methods* to the *training data* itself. Datasets, initially of hundreds of images and dozens of classes [24–26], have grown in size and complexity, and started becoming contributions in their own right. They have been fueling the progress of computer vision as much as, if not more than, the methods themselves. ImageNet [19], and mainly its ImageNet-1K [86] subset of about 1 million annotated images, has impacted the field in an unprecedented way. Yet, curating, and annotating such a dataset comes at a very high time, money and labor cost.

The last couple of years has seen the rise of large and generic models, trained on data which is less curated but orders of magnitude larger. Those proved to be easily applicable, either directly, or combined with a tailored model, to a wide range of computer vision transfer tasks [41, 45, 79, 117]. They have also been used beyond prediction tasks, *e.g.*, for text-conditioned image generation. Models such as DALL-E [80] or Stable Diffusion [85] have demonstrated impressive image generation ability. They produce fairly realistic synthetic images and exhibit an impressive degree of compositionality.

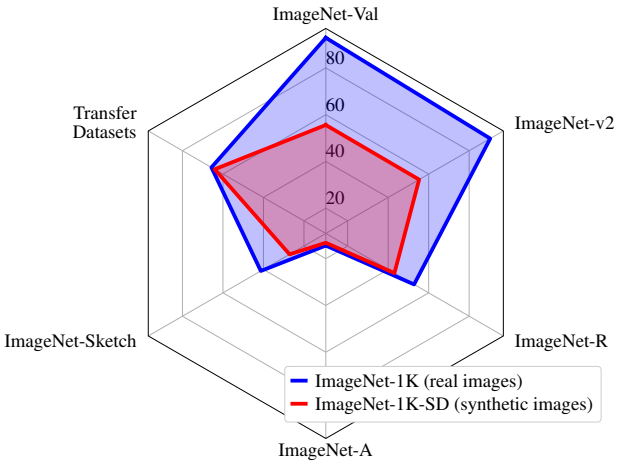


Figure 1. **Performance of ImageNet-SD models.** The blue polygon shows the performance of a model trained on ImageNet-1K. The red polygon depicts the performance of a model trained *only on synthetic data*, generated with Stable Diffusion [85] using the class names of ImageNet-1K. We report top-5 accuracy for all ImageNet test sets, and average top-1 over ten datasets for transfer.

Such generative models are trained on billion-scale datasets [93] composed of noisy image-text pairs scraped from the internet. Although training such models is out of reach for most institutions, a few of them have been made available to the community, and they will most likely fundamentally impact computer vision research. Given the impressive ability of these generative models, it is only natural to ask provocative questions such as: *Is there still a need for real images when training image prediction models?*

In this paper we explore this question through one of the most iconic computer vision dataset, ImageNet [19]. We study to which extent this dataset can be entirely replaced by synthetic images when learning deep models. More precisely, we assume that we are provided with a set of classes, and the Stable Diffusion v1.4 [85] model,[1] a generator that can produce realistic images from a textual prompt.

Our task is to learn an image classification model *from scratch* using a dataset composed only of synthetic images. We then evaluate the performance of this model on several datasets. First and foremost, we measure how well models and classifiers trained only on synthetic images recognize the training classes in real images from the standard ImageNet validation set. Then, we evaluate them on common datasets that test their resilience to domain shifts or adversarial examples, still for the ImageNet training classes. Finally, we consider several transfer learning scenarios where we measure the generalization performance of our models to novel sets of classes. Fig. 1 summarizes the main results by comparing models trained on two equally sized set of images from the same set of classes, one real and one synthetic, on a number of these tasks. The gap is surprisingly narrow, especially for some of these scenarios.

To summarize, our contributions are threefold. First, we leverage Stable Diffusion [85] and generate synthetic ImageNet clones, *i.e.*, datasets with synthetic images for the ImageNet classes, using class names as prompts. We analyse the generated images, highlight important issues, and propose class-agnostic alterations to the basic prompt that reduce semantic issues and increase diversity. Second, we train classification models using different ImageNet clones and show that they can achieve 83% and 51% top-5 accuracy on ImageNet-100 and ImageNet-1K respectively. Finally, we evaluate the generalization of our models. We show that their performance gap to models trained on real images is heavily reduced when testing for resilience to domain shifts or adversarial examples. Moreover, we show that our models perform on par with models trained conventionally when testing on 10 transfer datasets.

2. Related work

2.1. Learning with synthetic Data

Learning with synthetic data has long been a standard way to create large amounts of labeled data for annotation heavy tasks, such as human understanding [31, 61, 78, 100], semantic segmentation [15, 54, 55, 89, 98], optical flow estimation [21, 67, 109] or dense visual alignment [77]. In most cases, this synthetic data requires access to 3D models and renderers [61, 64, 120] or to a simulator [3, 18, 22, 83] with a physically plausible engine. Kataoka *et al*. [46] recently proposed pretraining on a database of synthetic fractal images before fine-tuning the model using real images on a downstream task. In this study we use synthetic data to learn encoders and classifiers that can be used out-of-the-box, without the need for a subsequent fine-tuning step. Closest to our work, [94] generates synthetic OCT images to train a glaucoma detection model to be applied to real images. Here, we target synthetic clones of complex natural image datasets, *i.e.*, ImageNet-1K [86], and we use a *general-purpose* text-to-image generation model.

Synthetic ImageNet clones. Synthetic images for ImageNet classes appear in a number of related works [4,54,81] using class conditional Generative Adversarial Networks (GANs), such as BigGAN [8]. Besnier *et al*. [4] generate images for ten ImageNet classes and propose techniques to reduce the gap between models trained on generated images and real ones. Li *et al*. [54] synthesize five images for each ImageNet-1K class, together with their semantic segmentation annotations to automatically generate pixel-level labels at scale. Our work focuses on image-level classification, and uses a general-purpose text-conditioned generative model instead of ImageNet-1K class-conditioned GANs. It further offers a larger scale study with promising results on the full ImageNet-1K benchmark when training from 1.28 million synthetic images.

Synthetic images as data++. Data sampled from generative models [30, 37, 70, 80, 85, 87] can be seen as data with added functionalities or "data++" [42]. Such data can be manipulated, interpolated or composed [12, 13, 28, 43, 44] with dedicated operators in their latent space, and further used for counterfactual reasoning [56, 65, 72, 92]. In this paper, we do not exploit these any added functionalities. Our prompts consider a class at a time and do not leverage any interpolation or composition properties of synthetic data. Instead, we chose our complete pipeline, including the set of data augmentations, to be identical to the one we use for real images, to allow for a fair comparison.

Zero-shot learning and test-time view synthesis. Generative models have been used to extend models to new classes, or to create novel views at test time. Chai *et al*. [13] synthesize novel views for test-time ensembling by perturbing the latent code of a test image. Aiming at zero-shot recognition [52,84,111], Elhoseiny *et al*. [23] synthesize a classifier for any novel class given its semantic description (*e.g.*, textual or attribute-based), whereas others synthesize image *features* using such descriptions [53, 90, 112]. Here we aim to learn visual encoders from scratch, and do not rely on encoders previously trained on real data.

2.2. Distillation of datasets and models

Knowledge distillation [9, 36] is a mechanism to transfer knowledge from a pretrained "teacher" model into a "student" one (see [105] for a survey), and it usually requires

[1] https://github.com/CompVis/stable-diffusion

images. Our approach can be seen as performing *image-free* distillation from a generic text-to-image generation model into a specific classification model. As we assume not to have access to any images to distill from, instead of distilling the visual encoder of the image generation model, inspired by recent works in NLP [59], we prompt a generation model to produce synthetic images and train a classifier with them.

Dataset distillation [11, 106, 118], on the other hand, is a way of compressing a training set of images into a smaller set of synthetic images such that after training a model on those, it performs as well as if it had been trained on the original set. However, one needs to tailor the generation process to a specific task, whereas in our case, we sample images from a task-agnostic generator.

Reconstructing images from model activations can be considered as another form of distillation. Earlier works reconstruct images from gradient-based features [103, 108] or from CNN activations [60]. Since then many methods have tried to uncover the training data distribution as it is stored in the weights of a model [14, 57, 69, 115]. Instead of trying to recover the training distribution of the teacher image generation model, we use prompting to distill its knowledge into a visual encoder for a specific image classification task.

3. Preliminaries

In this section, we first define the task we want to solve, *i.e.*, learning an image classification model when the training set of real images is replaced by an image generator, and training proceeds using only synthetically generated images. We then briefly describe Stable Diffusion [85], *i.e.*, the text-to-image generation model we use in this paper.

Task formulation. Our goal is to learn an image classification model given a set of class names $\mathcal{C}$ and a text-to-image generator $\mathcal{G}$. This task is a variant of image classification where the fixed-size image training set is replaced by an image generator. The model we want to learn consists of an encoder $\mathbf{z} = f_\theta(\mathbf{x})$ that maps an *image* $\mathbf{x}$ into a vector representation $\mathbf{z} \in \mathbb{R}^d$, and a classifier $\mathbf{y} = q(\mathbf{z})$ that outputs a probability distribution $\mathbf{y}$ over the N classes $c_i \in \mathcal{C}$, where $i = \{1, .., N\}$. We follow the common supervised learning setting [51, 86] and, unless otherwise stated, learn the encoder parameters θ together with the classifier q for the task. This model (encoder and classifier) is evaluated on the initial classification task, by applying it to real images (Sec. 5.1 and Sec. 5.2). We also test the visual encoder in the context of several transfer learning tasks (Sec.5.3).

Text-to-image with Stable Diffusion. We use the recent Stable Diffusion model [85] (SD) as text-to-image generator $\mathcal{G}$. SD is a denoising diffusion model [37] built around the idea of *latent diffusion*: The diffusion process is run on a compressed latent space for efficiency. An image encoder/decoder is used to interface the latent diffusion model with the pixel space. The generation process can be conditioned in many ways, *e.g.*, with text for text-to-image generation, or an image latent vector for image manipulation.

The text-to-image SD model consists of three main components: i) A variational autoencoder [47] whose visual encoder outputs a structured latent representation that is fed as input to the forward diffusion process and whose decoder is then used to convert the latent vectors back to pixels; ii) a denoising U-Net that conducts the diffusion process, and iii) a text encoder, *i.e.*, similar to the one used by CLIP [79].

The text-to-image generation process takes a textual prompt p as input and generates an image $\mathbf{x} \in \mathbb{R}^{W \times H \times 3}$. Let $g(p)$ denote the generation function of model $\mathcal{G}$. Image $\mathbf{x}$ is then given by $\mathbf{x} = g(p)$. In practice, the prompt p is first encoded via the text encoder and the text embedding is used as a conditioning vector for the latent diffusion process that runs for a number of *steps*. The latent representation is then provided to the decoder, which outputs the image $\mathbf{x}$.

There are two important parameters that control the quality and speed of text-conditioned diffusion; the number of diffusion steps and the coefficient that weights the textual conditioning vector. The former is linearly related to extraction time, while the latter can be used to modulate the adherence to the conditional signal. We report the values we used for these parameters in Sec. 5.

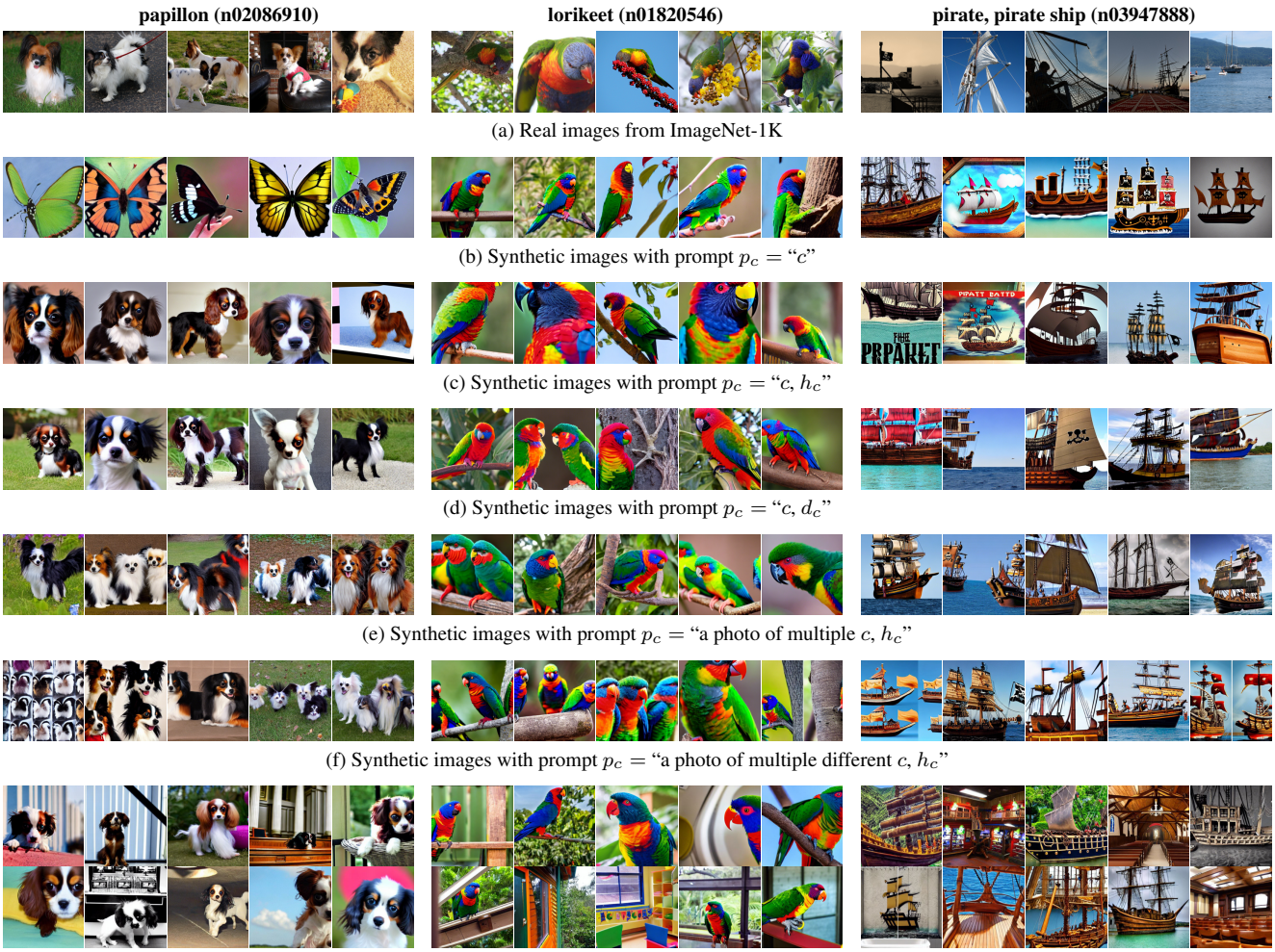
Link to distillation. Since the generator is a model that internally encodes visual information, the image classification model we learn is essentially derived from $\mathcal{G}$. Under this formulation, and as discussed in Sec. 2, one can see this task also as text-guided, image-free knowledge "distillation"[2]. Here we distill knowledge from a model of very different nature, *i.e.*, a text-to-image generation model, to a purely visual encoder, for solving a specific task.

4. Generating synthetic ImageNet clones

For our study, we create clones of the ImageNet [19] dataset by synthesizing images depicting the classes it contains. We refer to all synthetic datasets of ImageNet classes that are created using Stable Diffusion as **ImageNet-SD**. Sec. 4.1 describes different ways of creating ImageNet-SD datasets starting from simply using the class name as the prompt. We then present generic, class-agnostic ways for tackling issues that arise with respect to semantics and diversity in Secs. 4.2 and 4.3, respectively.

Note that we only present a limited set of qualitative results in Fig. 2 because of space constraints, but a far more extensive set can be found in the Appendix.

[2]The community typically uses the term "distillation" to denote knowledge transfer from a large model to a smaller one of the same nature [36].



(a) Real images from ImageNet-1K

(b) Synthetic images with prompt $p_c =$ "c"

(c) Synthetic images with prompt $p_c =$ "c, h_c"

(d) Synthetic images with prompt $p_c =$ "c, d_c"

(e) Synthetic images with prompt $p_c =$ "a photo of multiple c, h_c"

(f) Synthetic images with prompt $p_c =$ "a photo of multiple different c, h_c"

(g) Synthetic images with prompt $p_c =$ "c, h_c inside b"

Figure 2. **Qualitative results.** (a) Real ImageNet images. (b)-(g) Synthetic ImageNet-SD images generated with different prompts. Despite high photo-realistic quality, some issues are noticeable for (b) such as i) semantic errors *e.g.*, for the class "papillon", ii) lack of diversity, and iii) distribution shifts *e.g.*, towards cartoons for the "pirate" class. Such issues are addressed with more expressive prompts in (c)-(g).

4.1. Generating datasets using class names

In the absence of a training set of real images, we use the generator $\mathcal{G}$ presented in the previous section to synthesize images for each class in the class set $\mathcal{C}$. To do so, we need to provide the generator with at least one prompt per class. When used as an input, this class-conditioned prompt p_c triggers the generation of a synthetic image $\mathbf{x}_c = g(p_c)$ from class c. The simplest prompt one could think of is the class name *i.e.*, $p_c =$ "c". Note that although CLIP [79] uses $p_c =$ "a photo of c" for their zero-shot experiments, we found that only using the class name gave better results.

In the case of ImageNet, each class is associated with one or more *synsets*, *i.e.*, entities, in the WordNet [68] graph. We use the synset lemmas corresponding to each class as class-name prompt "c", comma-separated if more than one. Fig. 2b shows random examples of images generated with such prompts. At first glance, one can appreciate the ability of the generator to create photo-realistic images given only a class name (more qualitative experiments can be seen in Appendix). In Sec. 5, we show that one can already obtain surprisingly good image classification results by simply training a model with this synthetic dataset.

Upon close inspection of the generated images, however, some issues become apparent: a) **Semantic errors**: The generated images for some classes may capture the wrong semantics (*e.g.*, see the "papillon" class in Fig. 2b), b) **lack of diversity**: Generated images tend to look alike (an issue more apparent in the figures presented in Appendix C, where we show many more images per class), and c) **visual domain issues**: some classes tend to shift away from natural images towards sketches or art (*e.g.*, the "pirate ship" class in Fig. 2b). We discuss and address these issues next.

4.2. Addressing issues with semantics and domain

As mentioned before, by comparing the (real) images from ImageNet to the synthetic ones generated using only synset names as prompt, we observe that for some classes their semantics do not match. This is due to polysemy, *i.e.*, multiple potential semantic meanings or physical instantiations of the class names we used as prompt. We show one such case on the left-most column of Fig. 2b: The "papillon" images correspond to butterfly images for our generated dataset, while the ImageNet synset contains images of the dog breed of the same name (see Fig. 2a).

To fix this, we employ several simple ways that naturally reduce the semantic ambiguity of the prompt, leveraging the fact that the class names correspond to Wordnet [68] synsets. We augment the prompt for class name c with two additional sources of information that are provided by Wordnet: a) The *hypernyms* h_c of the synset as defined by the Wordnet graph, *i.e.*, the class name(s) of the parent node(s) of this class in the graph; and b) the *definition* d_c of the synset, *i.e.*, a sentence-length description of the semantics of each synset. In both cases, we append this information to the prompt, which becomes $p_c =$ "c, h_c" and $p_c =$ "c, d_c" for hypernyms and definition, respectively.

Qualitatively, we observed that issues regarding the semantic of the most problematic classes is fixed, and so are, up to some point, issues related to visual domain mismatch. These are also visible in Figs. 2c and 2d: Appending the hypernym ($h_c =$ "toy spaniel") or the description ($d_c =$ "small slender toy spaniel with erect ears and a black-spotted brown to white coat") of the class "papillon" in the prompt gives a set of images with the dog breed as the main subject. Appending the hypernym ($h_c =$ "ship") or the description ($d_c =$ "a ship that is manned by pirates") of the class "pirate ship" results in more natural-looking images rather than illustrations, reducing the domain shift.

Quantitatively, as we will observe in Sec. 5, the two variants listed above lead to better classification performance than the initial variant with just the class name.

4.3. Increasing the diversity of generated images

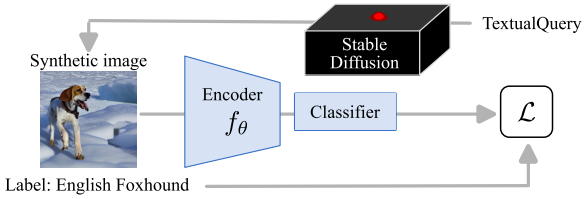
Generating images using more expressive prompts, *e.g.*, by appending class hypernym or definition, not only reduces semantic errors, but also increases the visual diversity of the output images. This is visible, for example, in the "lorikeet" and "pirate ship" classes in Figs. 2c and 2d when compared to Fig. 2b: The pose and viewpoints are slighly more diverse. However, images still tend to display the class instance centered and in a prominent position. The real ImageNet images feature significantly more diversity, several different settings and backgrounds, and, in several cases, multiple instances of the same class (*e.g.*, see Fig. 2a).

Although class-specific prompt engineering is an appealing option, in this study we chose to remain generic, and increase diversity in ways that can be applied regardless of the nature of the classes. To that end, we only make two assumptions about the class: a) There exists the notion of instance for that class; and b) this class can be "inside" a scene or background. These are generic assumptions that in practice can be satisfied by all object-centric classes.

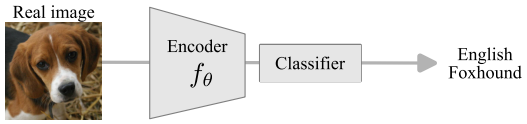
Generating multiple instances. We assume that there exists the notion of instance for class c. We leverage this to improve diversity by considering the following two simple, class-agnostic prompts: 1) $p_c =$ "a photo of multiple c, h_c" and 2) $p_c =$ "a photo of multiple different c, h_c". Although basic, we observed these two to produce the most realistic images over the variants we tried. We show results for these two prompts in Figs. 2e and 2f. Both prompts improve diversity in all three depicted classes, and are even more successful for classes whose instances tend to appear as groups, as for *e.g.*, parrots or dogs. Adding "different" gave a small but noticeable boost in diversity upon visual inspection (see also Appendix C for more qualitative results).

Diversifying the background. We assume that class c can be seen "inside" a scene or background. Once again, we approach prompt engineering in a class-agnostic way. We use *all* scene classes from the Places [121] dataset as background for every class. We generate images for every possible combination of a class c and a scene $b \in \mathcal{B}$ from the set $\mathcal{B}$ of 365 scenes in Places. We found that "c inside b" generally gave the best-looking results among a few prepositions we tried. However, we found that semantic and domain errors that arise from generating by only using class name remained after specifying a background. We therefore build on top of the second simplest, but more semantically correct prompt variant and use $p_c =$ "c, h_c inside b" as the prompt for generating images in diverse scenes and backgrounds. Although we do not consider this in our study, selecting backgrounds tailored for each class, *e.g.*, by matching class names to scenes using features from a text encoder, seems like a promising future direction.

Label noise and visual realism. A large portion of the generated images, especially those with random backgrounds (*e.g.*, see Fig. 2g) contain unlikely but believable combinations of objects and scenes. Some other images miss the foreground object completely (*e.g.*, see the bottom row in the middle column of Fig. 2g) or contain physically impossible combinations. Yet, we see such noisy or unrealistic synthetic images as a way of providing additional stochasticity during training, similar to what strong, non-realistic data augmentation achieves [27, 114]. In fact, it was recently shown [27] that diverse data augmentations, even when inconsistent with the data distribution, can be very valuable (even more than additional training data) for out-of-distribution scenarios. The experimental validation that we perform next corroborates this claim.



(a) Training a model on synthetic images.



(b) Testing the frozen model on real images.

Figure 3. **Overview of our experimental protocol.** During training, the model has access to synthetic images generated by the Stable Diffusion model, provided with a set of prompts per class. During evaluation, real images are classified by the frozen model.

5. Experiments

In this section we analyze the performance of image classification models learned using the different synthetic datasets constructed as described in Sec. 4. Due to the size of ImageNet-1K (roughly 1.3 million images), we perform most of our study on the smaller ImageNet-100 [97] dataset. This allows us to run multiple flavours of each synthetic dataset and to measure the impact of several design choices. Because ImageNet-100 is a random subset of ImageNet-1K, spanning over 100 classes and 126,689 images, it preserves some important characteristics of ImageNet-1K such as its fine-grained nature. We denote synthetic datasets for the two ImageNet subsets as ImageNet-100-SD (IN100-SD) and ImageNet-1K-SD (IN1K-SD), respectively.

Experimental protocol. We follow the protocol illustrated in Figure 3. The generator $\mathcal{G}$ is the Stable Diffusion [85] v1.4 model,[3] trained on the LAION2B-en dataset [93] and fine-tuned on a smaller subset filtered by an aesthetics classifier. During training, the generator is used to synthesize images for each class, which are then used for training the parameters of the encoder and the classifier. Unless otherwise stated, we create datasets of the *exact same size* as their real-image counterparts, *i.e.*, we generate the exact same number of images for every class as in the corresponding real dataset, maintaining any class imbalance.

We evaluate all models on *real images*. When evaluating their performance over the ImageNet classes, we use both the encoder and the classifier learned during training to predict labels of real images for the 5 ImageNet datasets (Secs. 5.1 and 5.2). For transfer learning (Sec. 5.3), we use the pretrained encoder as a feature extractor, and learn a separate linear classifier on each of the 10 transfer datasets.

Implementation details. In all experiments the encoder f_θ is a ResNet50 [32] encoder, trained for 100 epochs (unless otherwise stated) with mixed precision in PyTorch [75] using 4 GPUs where batch norm layers are synchronized. We use an SGD optimizer with 0.9 momentum, a batch size of 256 and a learning rate linearly increased during the first 10% of the iterations and then decayed with a cosine schedule. Unless otherwise stated, we use the data augmentation pipeline from DINO [10] with 1 global and 8 local crops ($M_g = 1$ and $M_l = 8$). For Stable Diffusion we use 50 diffusion steps and a guidance scale factor of 7.5 for all experiments. We generate RGB images of size 512×384. We discuss the effect of these parameters on the quality of generated images in Appendix C.7.

[3] https://huggingface.co/CompVis/stable-diffusion-v1-4

Training Dataset	PyTorch [75]	DINO (+ Multi-crop)
ImageNet-100 (real)	*86.6*	*87.4* (↑ 0.80)
ImageNet-100-SD (synthetic)	28.4	43.1 (↑ **14.6**)

Table 1. **Impact of data-augmentation** for models trained on real and synthetic datasets. Performance is measured on the validation set of ImageNet-100, *i.e.* on real images.

5.1. Results on ImageNet

In this section, we compare the performance of models trained on either real or synthetic images on the validation set of ImageNet-1K [86] (IN-Val) and the test set of ImageNet-v2 [82] (IN-v2).

Impact of data augmentation. We first evaluate the impact of different data augmentation strategies when learning from synthetic datasets. In Tab. 1, we report the performance of models trained on the simplest variant of ImageNet-100-SD, *i.e.*, using the class name as the prompt, utilizing either PyTorch [66, 75] or DINO [10] augmentations. Although the gains for the real images are relatively small (less than one percent), the gains for ImageNet-100-SD are over 14%. We believe this shows two things: i) Synthetic images can benefit from the same augmentations as real images, and ii) these transformations are good for domain generalization. Indeed, strong transformations have been shown to improve domain generalization [102], and consequently can reduce the sim-to-real gap.

ImageNet-SD with different prompts. Tab. 2 compares the performance of models trained using variants of ImageNet-100-SD created by the different prompts presented in Sec. 4. Looking at the results for ImageNet-val and ImageNet-v2 (four left-most columns), we make the following observations:

1. By simply using the class name as a prompt (row 2), one can synthesize images and learn a visual encoder *from scratch* that achieves *more than* 70% *Top-5 accuracy* (43% Top-1 accuracy) on a challenging 100-way classification task like ImageNet-100 that contains many fine-grained classes.
2. Adding the hypernym or the definition as part of the

Training Dataset	Rel. Size	Epoch	Prompt (p_c) or Model	IN-Val Top-1	IN-Val Top-5	IN-v2 Top-1	IN-v2 Top-5	IN-Sketch Top-1	IN-Sketch Top-5	IN-R* Top-1	IN-R* Top-5	IN-A* Top-1	IN-A* Top-5
ImageNet-100	*1×*	*100*	*Baseline model trained on real images*	*87.4*	*96.8*	*82.5*	*95.1*	*39.1*	*58.9*	*58.4*	*79.1*	*25.6*	*68.7*
ImageNet-100-SD	1×	100	p_c = "c"	43.1	70.7	45.4	70.7	29.9	53.5	51.7	75.3	8.8	38.4
			p_c = "c, h_c"	46.9	73.4	47.3	73.7	25.9	50.4	46.3	75.3	11.5	42.2
			p_c = "c, d_c"	47.9	74.2	49.1	74.9	24.7	49.2	41.2	71.5	12.2	38.5
			p_c = "a photo of multiple c, h_c"	46.5	72.7	47.2	73.5	24.4	48.4	45.8	73.6	**14.9**	47.9
			p_c = "a photo of multiple different c, h_c"	47.1	73.5	47.9	74.4	27.4	52.1	45.8	75.3	14.0	46.8
			p_c = "c, h_c inside b"	51.5	76.8	51.2	77.4	27.9	52.5	54.0	81.8	14.1	48.4
	10×	10	p_c = All six prompts above	**58.3**	**82.0**	**57.6**	**81.4**	**36.5**	**61.3**	**59.2**	**84.7**	13.2	**49.0**
ImageNet-1K	*1×*	*100*	*PyTorch [66] trained on real images*	*76.1*	*92.9*	*71.1*	*90.4*	*24.1*	*41.3*	*36.2*	*52.8*	*0.0*	*14.4*
	1×	*100*	*RSB-A1 [110] trained on real images*	*80.1*	*94.5*	*75.6*	*92.0*	*29.2*	*46.5*	*40.6*	*55.1*	*11.1*	*38.6*
ImageNet-1K-SD	1×	100	p_c = "c, d_c"	26.2	51.7	26.0	51.4	9.5	22.1	15.9	32.0	2.2	10.1
	1×	100	p_c = "c, h_c inside b"	30.1	55.6	29.8	55.3	11.9	27.1	23.5	43.1	3.4	13.2

Table 2. **Results on ImageNet datasets.** We report Top-1 and Top-5 accuracy on several ImageNet datasets, namely IN-Val (the official ILSVRC-2012 validation set [86]), IN-v2 [82], IN-Sketch [104], IN-R [34] and IN-A [35]. In all cases, testing is *on real images*. For the prompts, h_c and d_c refer to the hypernym and definition of class c provided by Wordnet [68], while b refers to scene classes from the Places 365 dataset [121]. *ImageNet-R and ImageNet-A only cover a subset of the classes of ImageNet-100; we therefore compute the reported metrics only on the common classes.

prompt (rows 3, 4) addresses some of the semantic and domain issues and this translates to performance gains.

3. Similarly, strong gains can also be achieved by devising prompts that improve diversity by generating multiple instances of the class (rows 5, 6).
4. Generating objects on diverse backgrounds, even in a simple and class-agnostic way, gives the best results so far, reaching over 50% Top-1 and 76% Top-5 accuracy on the challenging ImageNet-100 validation set.

Scaling the number of synthetic images. Unlike real datasets that are capped in the number of images they contain, ImageNet-SD has theoretically no size upper bound as one can generate images on demand. We therefore generated a dataset 10× larger than ImageNet-100, using all six prompt variants presented in Sec. 4 for the classes of ImageNet-100. We see in Tab. 2 (last row of the top section) that this achieves another $6-7\%$ gains in accuracy. Note that, since this model is 10× larger than the rest, we train it for 10× less epochs, *i.e.*, for the same total number of iterations as all other models.

In Fig. 4, we compare the performance of models trained from scratch, either on real or on synthetic datasets, as we vary their size from $(1/100)$-th of ImageNet-100, to 10 times its size. Although synthetic data seems to give a boost when generating an order of magnitude more data, the gain is not large enough to assume that generating even more data could allow the model to reach the performance of real images, at least for the basic case we test here, *i.e.*, without class-specific prompt engineering.

Results on ImageNet-1K. In the bottom section of Tab. 2 we report results on the very challenging 1000-way classification task of ImageNet-1K that contains many fine-grained categories of mushrooms, birds and dog species [39]. We see that the model trained on our synthetic ImageNet-1K-SD dataset using the prompt composed of the class name and hypernym (p_c = "c, d_c") reaches 26.2% Top-1 and 51.7% Top-5 accuracy on the challenging ImageNet-1K validation set. The variant with backgrounds is able to achieve even slightly higher accuracy. Although significantly lower than the results achieved by a model trained on the 1.3 million real images of ImageNet, we see that the synthetic dataset is able to capture at least partially the subtle clues needed to differentiate many fine-grained classes.

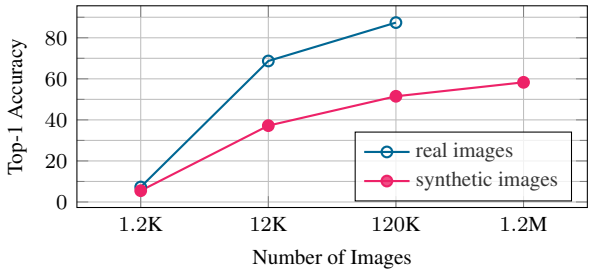


Figure 4. **Scaling the number of images** for training from $(1/100)$-th of the size of ImageNet-100, to 10× (for synthetic).

5.2. Resilience to domain shifts

We further investigate the performance of our models on three challenging evaluations sets for ImageNet-1K classes: ImageNet-Sketch [104] (IN-Sketch), ImageNet-R [34] (IN-R) and ImageNet-A [35] (IN-A). These datasets contain out-of-distribution images and their goal is to test resilience to domain shifts and adversarial images. Results are reported in the right-most columns of Tab. 2. When it comes

Training Dataset	Prompt (p_c) or Model	Aircraft	Cars196	DTD	EuroSAT	Flowers	Pets	Food101	SUN397	iNat18	iNat19	Avg.
ImageNet-100	*Baseline trained on real images*	*43.6*	*41.5*	*67.9*	*96.2*	*85.6*	*78.7*	*63.4*	*51.2*	*22.8*	*33.4*	*58.4*
ImageNet-100-SD	p_c = All six prompts (see Tab. 2)	44.9	37.1	68.3	95.7	84.9	74.5	60.1	51.1	22.4	33.7	57.3
ImageNet-1K	*PyTorch [66] trained on real images*	*48.9*	*49.9*	*72.1*	*96.2*	*89.3*	*92.3*	*71.2*	*60.5*	*35.5*	*41.5*	*65.7*
	RSB-A1 [110] trained on real images	*46.8*	*54.4*	*73.8*	*95.8*	*88.6*	*93.0*	*71.3*	*63.4*	*34.9*	*43.2*	*66.5*
ImageNet-1K-SD	p_c = "c, d_c"	48.7	49.7	71.6	96.5	90.1	81.9	66.4	55.8	28.7	40.6	63.0
	p_c = "c, h_c inside b"	49.6	47.4	72.1	95.9	89.3	87.2	67.7	59.5	30.8	41.4	64.1

Table 3. **Top-1 accuracy on ten transfer learning datasets** for encoders trained on real and synthetic images. We treat encoders as feature extractors and train linear classifiers on top for each dataset. We make the remarkable observation that representations from models trained on synthetic data can match the generalization performance of representations from models trained on millions of real images.

to the class set of ImageNet-100, we can see from the top section of the table that the performance of the best ImageNet-100-SD model, *i.e.*, the model trained using all prompts combined, not only rivals, but in a few cases *outperforms* the model trained on the real images in the training set of ImageNet-100. Results are especially strong on the datasets that aim at assessing robustness to domain shifts like ImageNet-Sketch and ImageNet-R.

When it comes to a much harder classification task like ImageNet-1K, however, we see from the bottom section of Tab. 2 that the same trend does not really hold. The ImageNet-1K-SD model trained on synthetic data lags behind in all cases when compared to the two models [75, 110] that are trained on the ImageNet-1K training set.

5.3. Transfer learning

In all our previous evaluations, we used pretrained models as a whole, *i.e.*, encoders together with classifiers, all trained on synthetic ImageNet datasets, and we directly applied those to predict the label of the (real) test images. Here, we use a slightly different protocol [49]. We evaluate the quality of the representations learned by our encoders alone, by using them as feature extractors and training linear logistic regression classifiers on top.

We consider ten commonly-used transfer datasets: Aircraft [62], Cars196 [50], DTD [17], EuroSAT [33], Flowers [71], Pets [74], Food101 [7], SUN397 [113], iNat2018 [99] and iNat2019 [99]. We report Top-1 accuracy on the (real) test set of these datasets in Tab. 3. We compare the ImageNet-100-SD and ImageNet-1K-SD visual encoders obtained with some of our best prompts with baselines trained on ImageNet-100 and ImageNet-1K, and what we observe is quite striking: On average, representations learned on purely synthetic images exhibit *generalization performance comparable to representations trained on thousands or millions of real images*. This suggests that synthetic images can be used to pretrain strong general-purpose visual encoders.

6. Discussion

This section takes a step back and considers some of the implications from the analysis proposed in this paper.

Applicability beyond ImageNet. The process we followed to create ImageNet-SD requires minimal assumptions and can be applied to a wider set of classes. To disambiguate semantics, we only assume access to a short textual description of the class. This is generally easy to acquire even at a larger scale, *e.g.*, in semi-automatic ways from Wikipedia. To improve diversity, we assumed that the notion of instance exists for every class, and that the class can be found "inside" a scene or a background. This limits the applicability of this process to object-centric classes.

Scaling laws for synthetic data. Conceptually, there is no reason to restrict our approach to training with a finite dataset of synthetic images. We could devise a training process which sees each image only once, following the standard continual or stream learning [73] protocols. However, Fig. 4 suggests that generating more images with basic prompts might not be enough, and that a performance leap will require advanced prompt engineering. We consider a study on scaling synthetic datasets important, but beyond the scope of this paper.

Yet, despite this scaling potential, the quality of the resulting classifier is bounded by the expressivity of the generator and the concepts it can reliably reproduce. No matter how intriguing the promise of an "infinite dataset" via data generation might be, practical applications are bound to the technical cost linked to computations and storage. Also, moderation of the content fueling this generator is to be considered, as it has strong implications discussed next.

Data and model bias. Because of its pioneering role as a source of images to train generic models, and all it has done to advance the computer vision field, ImageNet and some of its bias has been under heavy scrutiny [20, 58, 119]. Its synthetic counterparts have no reason to be immune to bias.

The main advantage of training with synthetic dataset is also its biggest flaw. Instead of manually curating and annotating a dataset, this process is outsourced to a

text-to-image generator, whose training data is not always known. Our study is based on the text-to-image generator of Stable Diffusion (SD). SD is trained on LAION-2B [93], a dataset scraped from the internet and then filtered in an automatic way using CLIP [79]. LAION has been shown to contain problematic content [6], and similarly built datasets are more than likely to share the same issues. Note that algorithmic bias is not only due to bias in the data [38], yet biased datasets lead to biased models and predictions [1, 88, 95]. Frameworks such as [40] could be considered to increase transparency and accountability.

Finally, on top of the bias contained in the data, the architecture itself constraints the generated images, and as such propagates, and potentially amplifies [5] existing bias. A major one that we have already discussed is the lack of diversity. An obvious corollary is the fact that stereotypes are reinforced. The options we have explored mitigate this issue to some limited extent, in that it improves classification results, but this issue is far from being solved.

As a final comment, there are many societal implications of using such models to generate synthetic datasets for training computer vision models, and a more thorough and multi-disciplinary discussion is required before even considering the use of any such models in practice.

7. Conclusions

In this paper, we study to which extent ImageNet, arguably the most popular computer vision dataset, can be replaced by a dataset synthesized by a top performing text-to-image generator. Through an extensive study, we find that one can learn models that exhibit surprisingly good performance on fine-grained classification tasks like ImageNet-100 and ImageNet-1K without any class-specific prompting. However, the most important result of this study is the finding that models trained on synthetic data exhibit exceptional generalization capability that rivals with models learned with real images. We see this study as merely a first glimpse of what is now possible with the latest large models in terms of visual representation learning. We envision similar approaches could be used to fine-tune or adapt models, using those synthetic datasets side-by-side with real ones.

Acknowledgements. This work was supported in part by MIAI@Grenoble Alpes (ANR-19-P3IA-0003), and the ANR grant AVENUE (ANR-18-CE23-0011).

References

[1] Osman Aka, Ken Burke, Alex Bauerle, Christina Greer, and Margaret Mitchell. Measuring model biases in the absence of ground truth. In *AIES*, 2021. 9

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proc. ICKDDM*, 2019. 14

[3] Alexander Amini, Igor Gilitschenski, Jacob Phillips, Julia Moseyko, Rohan Banerjee, Sertac Karaman, and Daniela Rus. Learning robust control policies for end-to-end autonomous driving from data-driven simulation. *RAL*, 2020. 2

[4] Victor Besnier, Himalaya Jain, Andrei Bursuc, Matthieu Cord, and Patrick Pérez. This dataset does not exist: training models from generated images. 2020. 2

[5] Federico Bianchi, Pratyusha Kalluri, Esin Durmus, Faisal Ladhak, Myra Cheng, Debora Nozza, Tatsunori Hashimoto, Dan Jurafsky, James Zou, and Aylin Caliskan. Easily accessible text-to-image generation amplifies demographic stereotypes at large scale, 2022. 9

[6] Abeba Birhane, Vinay Uday Prabhu, and Emmanuel Kahembwe. Multimodal datasets: misogyny, pornography, and malignant stereotypes. *CoRR*, abs/2110.01963, 2021. 9

[7] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – Mining discriminative components with random forests. In *Proc. ECCV*, 2014. 8, 15

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Proc. ICLR*, 2019. 2

[9] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proc. SIGKDD*, 2006. 2

[10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proc. ICCV*, 2021. 6

[11] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proc. CVPR*, 2022. 3

[12] Lucy Chai, Jonas Wulff, and Phillip Isola. Using latent space regression to analyze and leverage compositionality in {gan}s. In *Proc. ICLR*, 2021. 2

[13] Lucy Chai, Jun-Yan Zhu, Eli Shechtman, Phillip Isola, and Richard Zhang. Ensembling with deep generative views. In *Proc. CVPR*, 2021. 2

[14] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proc. ICCV*, 2019. 3

[15] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *Proc. CVPR*, 2019. 2

[16] Yunqiang Chen, Xiang Sean Zhou, and Thomas S Huang. One-class svm for learning in image retrieval. In *Proc. ICIP*, 2001. 1

[17] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proc. CVPR*, 2014. 8, 15

[18] Celso M de Melo, Antonio Torralba, Leonidas Guibas, James DiCarlo, Rama Chellappa, and Jessica Hodgins. Next-generation deep learning based on simulators and synthetic data. *Trends in cognitive sciences*, 2021. 2

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 1, 2, 3

[20] Emily Denton, Alex Hanna, Razvan Amironesei, Andrew Smart, and Hilary Nicole. On the genealogy of machine learning datasets: A critical history of imagenet. *Big Data & Society*, 2021. 8

[21] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. ICCV*, 2015. 2

[22] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *CORL*, 2017. 2

[23] Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *Proc. ICCV*, 2013. 2

[24] Mark Everingham, Luc Van Gool, Christopher Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes Challenge 2007 Results. 1

[25] Mark Everingham, Luc Van Gool, Christopher Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes Challenge 2012 Results. 1

[26] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proc. CVPR-W*, 2004. 1

[27] Jonas Geiping, Micah Goldblum, Gowthami Somepalli, Ravid Shwartz-Ziv, Tom Goldstein, and Andrew Gordon Wilson. How much data are augmentations worth? an investigation into scaling laws, invariance, and implicit regularization. 2022. 5

[28] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *Proc. ICCV*, 2019. 2

[29] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. 2016. 1

[30] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11), 2020. 2

[31] Xi Guo, Wei Wu, Dongliang Wang, Jing Su, Haisheng Su, Weihao Gan, Jian Huang, and Qin Yang. Learning video representations of human motion from synthetic data. In *Proc. CVPR*, 2022. 2

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 1, 6

[33] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *JSTAEORS*, 2019. 8, 15

[34] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proc. ICCV*, 2021. 7, 15

[35] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proc. CVPR*, 2021. 7, 15

[36] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Proc. NeurIPS-W*, 2014. 2, 3

[37] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proc. NeurIPS*, 2020. 2, 3

[38] Sara Hooker. Moving beyond "algorithmic bias is a data problem". *Patterns*, 2021. 9

[39] Minyoung Huh, Pulkit Agrawal, and Alexei Efros. What makes imagenet good for transfer learning? *arXiv:1608.08614*, 2016. 7

[40] Ben Hutchinson, Andrew Smart, Alex Hanna, Emily Denton, Christina Greer, Oddur Kjartansson, Parker Barnes, and Margaret Mitchell. Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. In *ACM FAccT*, 2021. 9

[41] 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. 1

[42] Phillip Isola. When faking your data actually helps – learning vision from gans, nerfs, and noise, 2022. BMVC Keynote talk. 2

[43] Ali Jahanian*, Lucy Chai*, and Phillip Isola. On the "steerability" of generative adversarial networks. In *Proc. ICLR*, 2020. 2

[44] Ali Jahanian, Xavier Puig, Yonglong Tian, and Phillip Isola. Generative models as a data source for multiview representation learning. 2022. 2

[45] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proc. ICML*, 2021. 1

[46] Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without natural images. *IJCV*, 2022. 2

[47] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proc. ICLR*, 2014. 3

[48] Simon Kornblith, Ting Chen, Honglak Lee, and Mohammad Norouzi. Why do better loss functions lead to less transferable features? In *Proc. NeurIPS*, 2021. 14, 16

[49] Simon Kornblith, Jonathon Shlens, and Quoc Le. Do better imagenet models transfer better? In *Proc. CVPR*, 2019. 8, 14

[50] Jonathan Krause, Jia Deng, Michael Stark, and Fei-Fei Li. Collecting a large-scale dataset of fine-grained cars. In *Proc. ICCV-W*, 2013. 8, 15

[51] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. NeurIPS*, 2012. 1, 3

[52] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *PAMI*, 2014. 2

[53] Michalis Lazarou, Tania Stathaki, and Yannis Avrithis. Tensor feature hallucination for few-shot learning. In *Proc. WACV*, 2022. 2

[54] Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Sanja Fidler, and Antonio Torralba. Bigdatasetgan: Synthesizing imagenet with pixel-wise annotations. In *Proc. CVPR*, 2022. 2

[55] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *Proc. CVPR*, 2021. 2

[56] Shusen Liu, Bhavya Kailkhura, Donald Loveland, and Yong Han. Generative counterfactual introspection for explainable deep learning. 2019. 2

[57] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv:1710.07535*, 2017. 3

[58] Alexandra Sasha Luccioni and David Rolnick. Bugs in the data: How imagenet misrepresents biodiversity. *arXiv:2208.11695*, 2022. 8, 18

[59] Xinyin Ma, Xinchao Wang, Gongfan Fang, Yongliang Shen, and Weiming Lu. Prompting to distill: Boosting data-free knowledge distillation via reinforced prompt. In *Proc. IJCAI*, 2022. 3

[60] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proc. CVPR*, 2015. 3

[61] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. In *Proc. ICCV*. 2

[62] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv:1306.5151*, 2013. 8, 15

[63] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *Proc. ICCV*, 2011. 1

[64] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proc. ICCV*, 2019. 2

[65] Chengzhi Mao, Augustine Cha, Amogh Gupta, Hao Wang, Junfeng Yang, and Carl Vondrick. Generative interventions for causal learning. In *Proc. CVPR*, 2021. 2

[66] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proc. ACM-ICM*, 2010. 6, 7, 8

[67] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proc. CVPR*, 2016. 2

[68] George A Miller. Wordnet: A lexical database for english. *Commun. ACM*, 1995. 4, 5, 7, 15

[69] Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *Proc. ICML*, 2019. 3

[70] 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. *arXiv:2112.10741*, 2021. 2

[71] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Proc. ICCVGIP*, 2008. 8, 15

[72] Deniz Oktay, Carl Vondrick, and Antonio Torralba. Counterfactual image networks, 2018. 2

[73] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113, 2019. 8

[74] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *Proc. CVPR*, 2012. 8, 15

[75] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Proc. NeurIPS*, 2019. 6, 8, 14

[76] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *JMLR*, 12, 2011. 14

[77] William Peebles, Jun-Yan Zhu, Richard Zhang, Antonio Torralba, Alexei A Efros, and Eli Shechtman. Gan-supervised dense visual alignment. In *Proc. CVPR*, 2022. 2

[78] Albert Pumarola, Jordi Sanchez-Riera, Gary P. T. Choi, Alberto Sanfeliu, and Francesc Moreno-Noguer. 3dpeople: Modeling the geometry of dressed humans. In *Proc. ICCV*, 2019. 2

[79] 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 *Proc. ICML*, 2021. 1, 3, 4, 9

[80] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proc. ICML*, 2021. 1, 2

[81] Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. In *Proc. NeurIPS*, 2019. 2

[82] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *Proc. ICML*, 2019. 6, 7, 15

[83] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *Proc. ECCV*, 2016. 2

[84] Marcus Rohrbach, Michael Stark, and Bernt Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Proc. CVPR*, 2011. 2

[85] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. CVPR*, 2022. 1, 2, 3, 6, 17

[86] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 1, 2, 3, 6, 7, 15

[87] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv:2205.11487*, 2022. 2

[88] Hadi Salman, Saachi Jain, Andrew Ilyas, Logan Engstrom, Eric Wong, and Aleksander Madry. When does bias transfer in transfer learning? *arXiv:2207.02842*, 2022. 9

[89] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *Proc. CVPR*, 2018. 2

[90] Mert Bulent Sariyildiz and Ramazan Gokberk Cinbis. Gradient matching generative networks for zero-shot learning. In *Proc. CVPR*, 2019. 2

[91] Mert Bulent Sariyildiz, Yannis Kalantidis, Diane Larlus, and Karteek Alahari. Concept generalization in visual representation learning. In *Proc. ICCV*, 2021. 14

[92] Axel Sauer and Andreas Geiger. Counterfactual generative networks. In *Proc. ICLR*, 2021. 2

[93] 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:2210.08402*, 2022. 1, 6, 9

[94] Ashish Jith Sreejith Kumar, Rachel S. Chong, Jonathan G. Crowston, Jacqueline Chua, Inna Bujor, Rahat Husain, Eranga N. Vithana, Michaël J. A. Girard, Daniel S. W. Ting, Ching-Yu Cheng, Tin Aung, Alina Popa-Cherecheanu, Leopold Schmetterer, and Damon Wong. Evaluation of Generative Adversarial Networks for High-Resolution Synthetic Image Generation of Circumpapillary Optical Coherence Tomography Images for Glaucoma. *JAMA Ophthalmology*, 140(10), 2022. 2

[95] Ryan Steed and Aylin Caliskan. Image representations learned with unsupervised pre-training contain human-like biases. In *FAccT*, 2021. 9

[96] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2015. 1

[97] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv:1906.05849*, 2019. 6

[98] Nontawat Tritrong, Pitchaporn Rewatbowornwong, and Supasorn Suwajanakorn. Repurposing gans for one-shot semantic part segmentation. In *Proc. CVPR*, 2021. 2

[99] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proc. CVPR*, 2018. 8, 14, 15

[100] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proc. CVPR*, 2017. 2

[101] Andrea Vedaldi, Varun Gulshan, Manik Varma, and Andrew Zisserman. Multiple kernels for object detection. In *Proc. ICCV*, 2009. 1

[102] Riccardo Volpi, Diane Larlus, and Gregory Rogez. Continual adaptation of visual representations via domain randomization and meta-learning. In *Proc. CVPR*, 2021. 6

[103] Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, and Antonio Torralba. Hoggles: Visualizing object detection features. In *Proc. ICCV*, 2013. 3

[104] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Proc. NeurIPS*, 2019. 7, 15

[105] Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *PAMI*, 2021. 2

[106] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv:1811.10959*, 2018. 3

[107] Yizhou Wang, Shixiang Tang, Feng Zhu, Lei Bai, Rui Zhao, Donglian Qi, and Wanli Ouyang. Revisiting the transferability of supervised pretraining: an mlp perspective. In *Proc. CVPR*, 2022. 14, 16

[108] Philippe Weinzaepfel, Hervé Jégou, and Patrick Pérez. Reconstructing an image from its local descriptors. In *Proc. CVPR*, 2011. 3

[109] Yo whan Kim, Samarth Mishra, SouYoung Jin, Rameswar Panda, Hilde Kuehne, Leonid Karlinsky, Venkatesh Saligrama, Kate Saenko, Aude Oliva, and Rogerio Feris. How transferable are video representations based on synthetic data? In *NeurIPS Datasets and Benchmarks Track*, 2022. 2

[110] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. In *Proc. NeurIPS-W*, 2021. 7, 8

[111] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *PAMI*, 41(9), 2018. 2

[112] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *Proc. CVPR*, 2018. 2

[113] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proc. CVPR*, 2010. 8

[114] Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. In *Proc. ICLR*, 2021. 5

[115] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proc. CVPR*, 2020. 3

[116] Yaodong Yu, Kwan Ho Ryan Chan, Chong You, Chaobing Song, and Yi Ma. Learning diverse and discriminative representations via the principle of maximal coding rate reduction. In *Proc. NeurIPS*, 2020. 14, 16

[117] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv:2111.11432*, 2021. 1

[118] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *Proc. ICLR*, 2021. 3

[119] Shengjia Zhao, Hongyu Ren, Arianna Yuan, Jiaming Song, Noah Goodman, and Stefano Ermon. Bias and generalization in deep generative models: An empirical study. In *Proc. NeurIPS*, 2018. 8

[120] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. In *Proc. ECCV*, 2020. 2

[121] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *PAMI*, 2017. 5, 7, 17

Contents

A. Evaluation protocol

We evaluate our models in two ways. For the different ImageNet test sets, *i.e*. datasets with images from the training classes (ImageNet-Val/v2/R/A/Sketch), we use the pretrained models as well as the classifiers we learn during pretraining using only synthetic images. For classification tasks on novel classes, *i.e*. on 10 the transfer datasets we consider, we freeze the pretrained encoder and train from scratch a new set of linear classifiers for each transfer task. The list of all datasets we use is given in Tab. 4.

For transfer learning evaluations, we follow the linear classification protocols from [49, 91]. More precisely, for each of the 10 transfer datasets (2nd part of Tab. 4), we first extract image representations (features) from the pretrained encoders and then train linear logistic regression classifiers using these features. As most transfer datasets are small, we follow [49] and train classifiers using LBFG-s implemented in Scikit-learn [76]. For the larger iNaturalist 2018 [99] and iNaturalist 2019 [99] datasets we train linear classifiers in PyTorch [75] using SGD. In all cases we resize the images with bicubic interpolation so that their shortest side is 224 pixels, and then taking a central crop of 224×224 pixels. We tune hyper-parameters (learning rate and weight decay for the SGD optimizer, and regularization coefficient for the LBFG-s optimizer) using Optuna [2] over at least 25 trials.

B. Extended experimental results

B.1. Analysis of the learned features

In the main paper, we evaluate our models on a list of image classification tasks, *i.e*. 5 test sets for the ImageNet classes, and 10 datasets for transfer learning. In this section, we analyze and contrast the *representations* obtained with models we trained using synthetic images to representations from models trained on real images. We perform our analysis for ImageNet-100 and using **four metrics**: a) Sparsity, b) intra-class distance, c) feature redundancy and d) coding length. Note that we use the terms "representations" and "features" interchangeably.

We compare four different models trained on either real or synthetic data for the 100 classes of ImageNet-100: One model trained on real images, ImageNet-100-Real, two models trained on synthetic image sets of the same size obtained by using two different prompts: p_c = "c" and p_c = "c, h_c inside b", and the ImageNet-100-SD-10x model, trained using ten times more images.

We perform these analyses on all the datasets we consider in the main paper and listed in Tab. 4. For the sake of this study, we split them into three groups: a) ImageNet-100-Val/v2, b) ImageNet-100-Sketch/A/R and c) the 10 transfer datasets. For each pretrained model and dataset, we extract features for either only the images in the test set (for the ImageNet test sets), or for all images (for the small transfer datasets). We then compute each of the four metrics separately on each dataset, and average them over all datasets in the same group. Before computing metrics, we ℓ_2-normalize features.

Result analysis for each of the four metrics follows.

Sparsity. Inspired by [48], we compute feature *sparsity ratio*, *i.e*., the percentage of feature dimensions close to zero with a threshold of 10^{-5}. We report sparsity ratios in Fig. 5a. We see that the sparsity ratio for the models trained on synthetic images increases as the "diversity" of a synthetic dataset increases, *i.e*., we see gradual increase in sparsity scores from p_c = "c" and p_c = "c, h_c inside b" to ImageNet-100-SD-10x. This observation aligns with their performance as well, *i.e*., in the main paper we show that ImageNet-100-SD-10x performs best in general while p_c = "c" performs worst. More interestingly, we see that ImageNet-100-Real, the model trained on real images, learns the most sparse representations.

Intra-class distance. In the main paper, we present simple ways to increase the diversity of synthetic images. Now we check if these efforts increase the variance of samples in the representation space. To do that, we compute the average ℓ_2-distance between samples from the same class (*i.e*., intra-class distance). We see in Fig. 5b that models trained with more diverse images indeed learn representations with higher intra-class variance.

Feature redundancy. Following [107], we compute feature redundancy, *i.e*., average pairwise Pearson correlation among dimensions. From Fig. 5c we see that the redundancy of features learned on real images increase more rapidly than the ones learned on synthetic images, as we move from ImageNet-100-Val/v2 towards out-of-domain or transfer datasets.

Coding length. To further investigate our observation on feature redundancy, we follow [116] and compute the

Dataset	# Classes	# Train samples	# Val samples	# Test samples	Val provided	Test provided
ImageNet test sets (training classes)						
ImageNet-Val [86] (IN-Val)	1000	–	–	50000	–	✓
ImageNet-v2 [82] (IN-v2)	1000	–	–	3×10000	–	✓
ImageNet-Sketch [104] (IN-Sketch)	1000	–	–	50889	–	✓
ImageNet-R [34] (IN-R)	200	–	–	30000	–	✓
ImageNet-A [35] (IN-A)	200	–	–	7500	–	✓
Transfer tasks (novel classes)						
Aircraft [62]	100	3334	3333	3333	✓	✓
Cars196 [50]	196	5700	2444	8041	–	✓
DTD [17]	47	1880	1880	1880	✓	✓
EuroSAT [33]	10	13500	5400	8100	–	–
Flowers [71]	102	1020	1020	6149	✓	✓
Pets [74]	37	2570	1110	3669	–	✓
Food101 [7]	101	68175	7575	25250	–	✓
Pets [74]	397	15880	3970	19850	–	✓
iNaturalist 2018 [99]	8142	437513	–	24426	–	✓
iNaturalist 2019 [99]	1010	265213	–	3030	–	✓

Table 4. **Datasets** we use for evaluating our models.

average coding length per sample on each dataset (see Fig. 5d). We see that models trained on ImageNet-100-Real and ImageNet-100-SD-10x are comparable.

B.2. Additional spider plots

In Fig. 6 we show spider plots for the models trained on either real or synthetic data for ImageNet-100 and ImageNet-1K. In both cases, we show two plots which respectively report top-1 and top-5 accuracy for the ImageNet datasets, *i.e.*, ImageNet-Val/v2/R/A/Sketch. For transfer datasets, we always report average top-1 accuracy.

C. Extended qualitative results

In this section, we provide additional qualitative results. First we show random images for *all* ImageNet-100 classes from three datasets: ImageNet-100-Val (real images) and two ImageNet-100-SD datasets generated by the prompts $p_c =$ "c" and $p_c =$ "c, h_c inside b". Then we discuss in more detail several types of issues that we observed in these synthetic images.

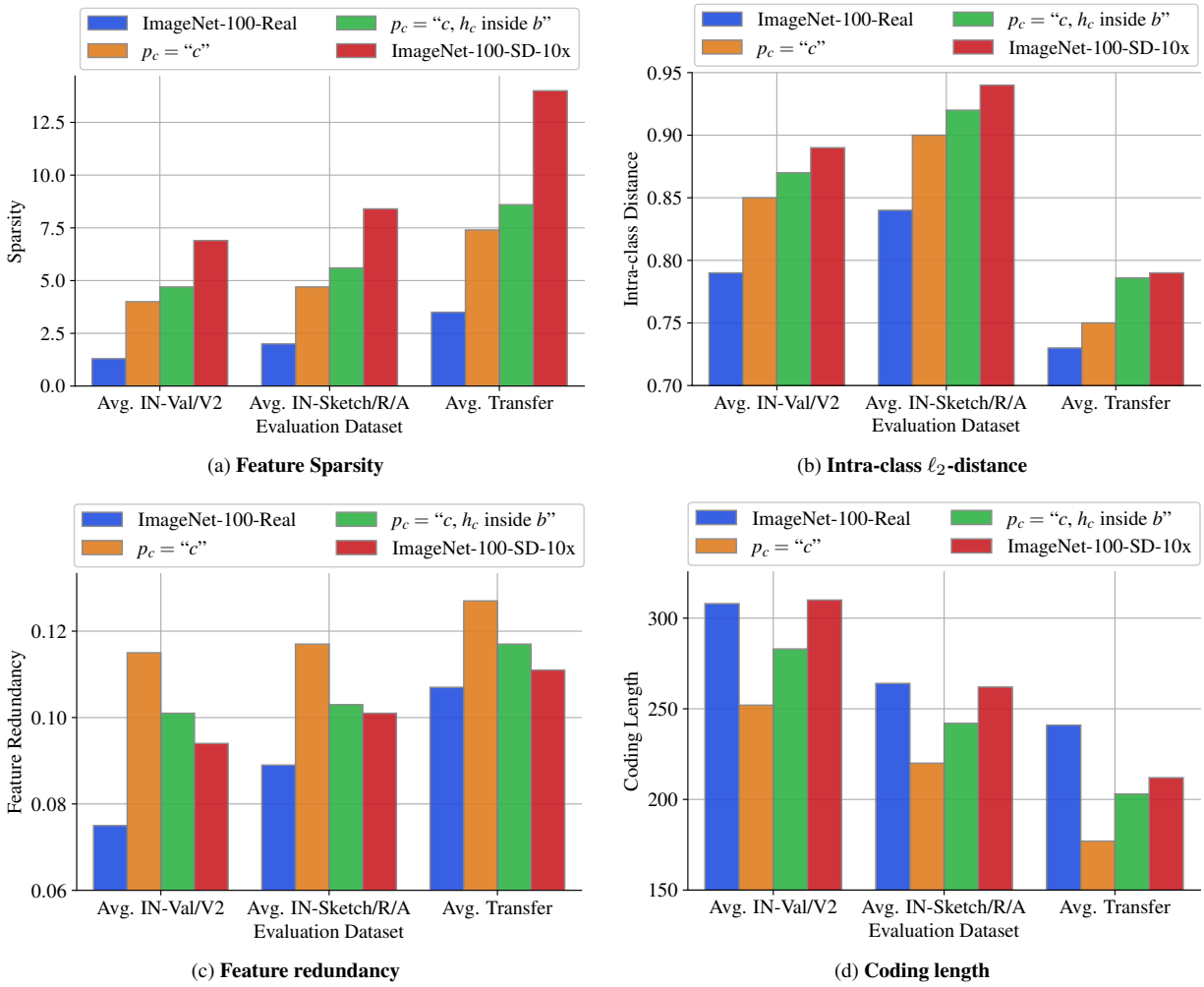
Qualitative results for *all* ImageNet-100 classes. In Fig. 10, we show a few random images from each of the 100 classes in ImageNet-100, for three datasets: a) The real images from ImageNet-100, b) synthetic images generated by a simple prompt, which is only composed of the name of the class, and c) synthetic images generated with a prompt that enforces thoses classes to appear in diverse backgrounds to improve the diversity of generated images. From this exhaustive list, even with a few images per class, one can observe a number of issues around the semantics, diversity and domain of those images.

Showcasing domain and diversity issues. We also show extended results for three classes in order to illustrate issues related to the domain and diversity. Fig. 9 compares generated images between two fine-grained classes of crabs, while Fig. 8 shows many images from multiple different generated datasets for a single dog class. We discuss both figures in the next subsections.

C.1. Semantic errors

From closely inspecting the generated images we can see that there exists two classes for which the prompt $p_c =$ "c" produces images of the wrong semantics: For the classes "papillon" and "wing", we see the generated images in the middle column of Fig. 10 to be wrong due to *polysemy* associated with the class names. What is more, although not fully visible from the small set of images we show here, we saw that semantics are partially wrong for at least the classes "green mamba", "walking stick" and "iron". For "green mamba", although the synset refers to the snake species, there is a car model of the same name appearing in some of the generated images instead. For "walking stick", the synset refers to the insect, while a subset of the generated images also contained walking sticks that are not insects.

As we discuss in the paper, appending the hypernym or definition of each synset seems to fix polysemy issues in many cases, including the ones mentioned above. However, we can see at least two cases where adding the hypernym in the prompt leads to worse results. According to WordNet [68], the hypernym for "shih-tzu" is "toy dog" something that results in dog-shaped toys in many of the generated images. Another example is the class "boathouse",



(a) **Feature Sparsity**

(b) **Intra-class ℓ_2-distance**

(c) **Feature redundancy**

(d) **Coding length**

Figure 5. **Feature analyses** for models. We perform these analyses on top of features extracted from pretrained encoders f trained on either real or synthetic data for ImageNet-100 (training data is specified in the legends of the subfigures). *Sparsity* is measured by the percentage of dimensions close to zero [48]. *Intra-class ℓ_2-distance* is the average pairwise ℓ_2-distance between samples from the same class. These two metrics are computed on ℓ_2-normalized features. *Feature redundancy* [107] is obtained by $\mathcal{R} = \frac{1}{d^2}\sum_i \sum_j |\rho(\mathbf{X}_{:,i}, \mathbf{X}_{:,j})|$, where $\mathbf{X} \in N \times d$ is a feature matrix containing N samples, each encoded into a d-dimensional representation (2048 in our case) and $\rho(\mathbf{X}_{:,i}, \mathbf{X}_{:,j})$ is the Pearson correlation between a pair of feature dimensions i and j. *Coding length* [116] is measured by $R(\mathbf{X}, \epsilon) = \frac{1}{2}\log\det(\mathbf{I}_d + \frac{d}{N\epsilon^2}\mathbf{X}^\top\mathbf{X})$, where $\mathbf{I}_d$ is a d-by-d identity matrix, ϵ^2 is the precision parameter set to 0.5.

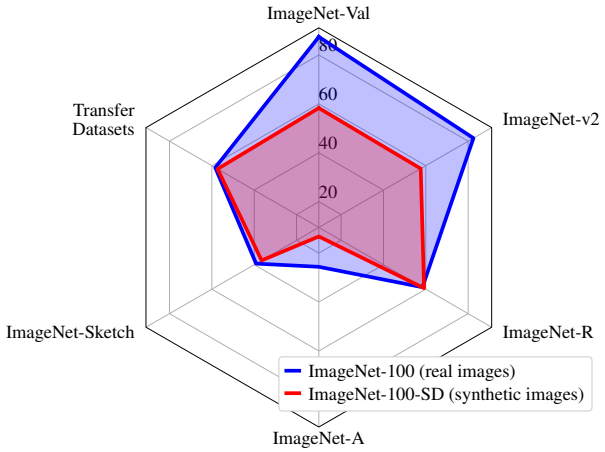
where appending the parent class "shed" leads to sheds that are not inside a body of water.

C.2. NSFW content

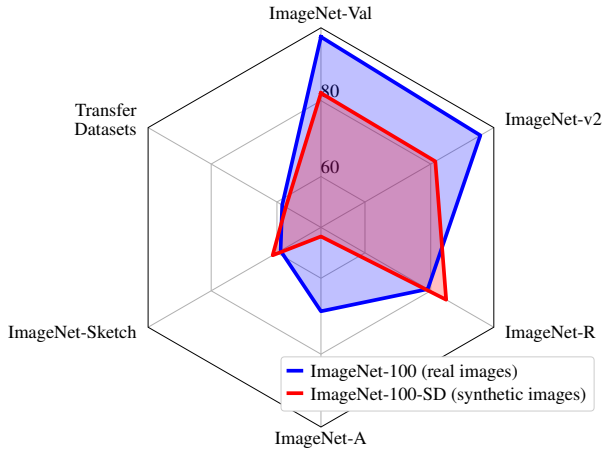
Another issue that was not very prominent, but still visible, even in the case of generic animal and object categories present in ImageNet-100, was the fact that some of the generated images contained NSFW (Not Suitable For Work) content in the form of nudity. The open-source code for Stable Diffusion comes with a highly selective safety module, that discards generated images that might contain NSFW content.[4] We disabled this module when generating images for the ImageNet synsets as we wanted to study the model as-is first, and to understand the problem.

We thoroughly inspected all classes of ImageNet-100 and observed minor NSFW issues with only two of the classes: 1) The basic prompt for the class "sarong" led to a few images that had partial nudity. This effect was exaggerated when adding the description of the concept that reads "a loose skirt consisting of brightly colored fabric wrapped around the body; worn by both women and men in

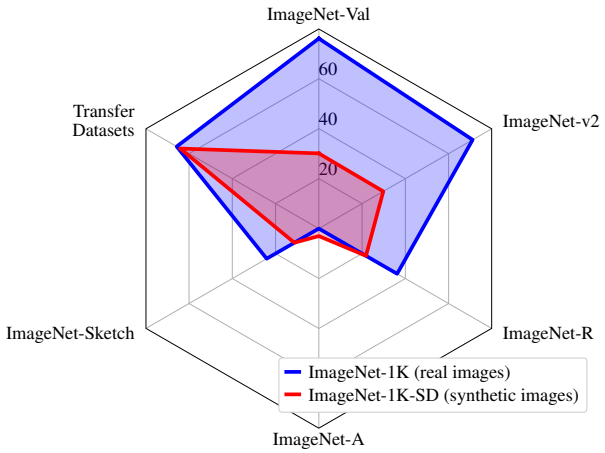
[4] https://huggingface.co/CompVis/stable-diffusion-v1-4?text=Safety



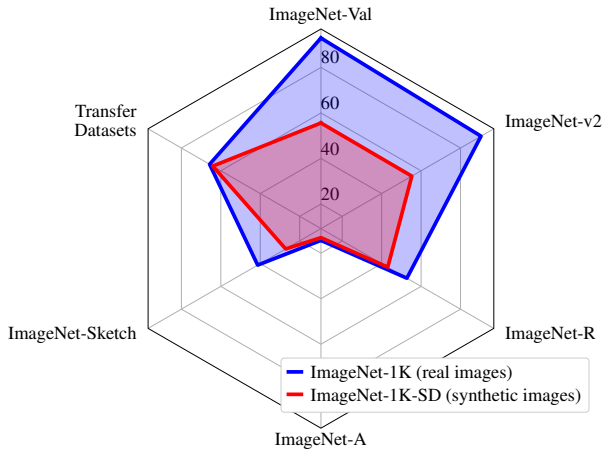
(a) Top-1 accuracy, training on **ImageNet-100**.



(b) Top-5 accuracy, training on **ImageNet-100** (top-1 for transfer tasks).



(c) Top-1 accuracy, training on **ImageNet-1K**.



(d) Top-5 accuracy, training on **ImageNet-1K** (top-1 for transfer tasks).

Figure 6. **Performance card of models** trained on either real or synthetic data for 100 classes of ImageNet-100 (Figs. 6a and 6b) and for all the 1000 classes of ImageNet-1K (Figs. 6c and 6d). In all figures, the blue polygon shows the performance of a model trained on the real images from ImageNet, and the red polygon depicts the performance of a model trained *only on synthetic data*, generated with Stable Diffusion [85] using p_c = "c, h_c inside b" as the prompt. In Figs. 6a and 6c and in Figs. 6b and 6d we report top-1 and top-5 accuracy over the ImageNet datasets (*i.e.*, ImageNet-Val/v2/R/A/Sketch), whereas, in all figures we report top-1 accuracy averaged over 8 transfer datasets. Note that Fig. 6d corresponds to Fig 1 of the main paper.

the South Pacific". It seems that words like "body" biases the image generation process towards more NSFW content. 2) Prompts for the class "ski mask" in combination with certain backgrounds from the Places dataset [121] also resulted in nudity. Overall, we want to emphasize that the Stable Diffusion models we tested were all highly susceptible to generate such content, something that shows the biases existing in the LAION training set.

C.3. Misrepresentation of biodiversity

The degree of misrepresentation of biodiversity in the images generated from Stable Diffusion is very high. We partially showcase the issue in Fig. 9 where we show many generated images for two fine-grained classes, *i.e.*, "rock crab" and "fiddler crab".

"Rock crab" is defined in WordNet as "crab of eastern coast of North America", while the "fiddler crab" as a "burrowing crab of American coastal regions having one claw much enlarged in the male". The fact that the male fiddler crab has one claw much larger is a prominent theme when it comes to the real ImageNet-100 images shown on the right side of Fig. 9a.

It does not take an expert ecologist to see that, although most of the generated images capture the coarser class "crab", the visual differences between the two sets of images, *e.g.*, in Fig. 9b, are not focusing on the single enlarged

claw for the fiddler crab case. What is more, the exhibited intra-class visual diversity, *i.e.*, crabs of different shapes and colors, seems to exceed a single species of crab.

This is just a single example, but from our inspection of many other fine-grained animal and fungi classes, we could see that this is not an isolated issue. On the contrary, it seems prominent across many fine-grained domains. One exception for the subset of ImageNet classes we delved into is dog breeds, possibly due to the sheer volume of dog images on the internet. It is however fair to say that the generated images highly misrepresent biodiversity.

It is worth noting that, as Luccioni and Rolnick discuss in their recent paper [58], the ImageNet dataset itself contains a number of issues when it comes to the annotations of fine-grained classes of wild animals. They found that "many of the classes are ill-defined or overlapping, and that 12% of the images are incorrectly labeled, with some classes having $> 90\%$ of images incorrect". Although we did not conduct a similar experiment using experts, we expect similar statistics to be much higher for the images generated by Stable Diffusion.

C.4. Semantic issues arising with backgrounds

A common issue we observe when adding diverse backgrounds to class images is that a subset of the generated images do not really contain the object, and merely reflect the background scene. See for example the images in the first and last row, on the last column of Fig. 9c, and a few more spread in that figure, or the background samples for class "reel" in Fig. 10. This is to be expected given how a prompt like this is relying on the compositionality of the Stable Diffusion model.

What is really interesting is that in some cases the resulting images, although not containing an instance from the class, retains some of the object's shape or texture in the background. See for example a pedestal-looking table in Fig. 9c for class "pedestal", a pirate themed bedroom for class "pirate", green shirts for "green mamba", or the reddish produce stand for "red fox".

C.5. Issues with diversity

We observe issues with diversity for most of the classes when only the class name is used as the prompt, *e.g.*, in the middle set of results in Fig. 10. This is also visible for the crab classes in Fig. 9b, or the Shih-tzu class in Fig. 7b, Fig. 8a and Fig. 8c. We see that such issues are partially solved when using the multiple instance prompts (*e.g.*, Fig. 8b) or backgrounds (*e.g.*, the right-most set of images in Fig. 10). We expect more advanced prompt engineering to further increase diversity.

As expected, increasing diversity correlates with more semantic errors. We see that such issues appear far more frequently in the most diverse synthetic dataset, *i.e.* as shown in the right-most set of images of Fig. 10.

C.6. Non-natural images

Even from the very small random sample of generated images shown in the figures of this paper, we see that there is a non-negligible percentage of the generated images that are non-natural. They can be illustrations, graphics images or even paintings. This is not necessarily undesirable and it can lead to models with higher robustness to related domain changes.

C.7. Varying the stable diffusion parameters

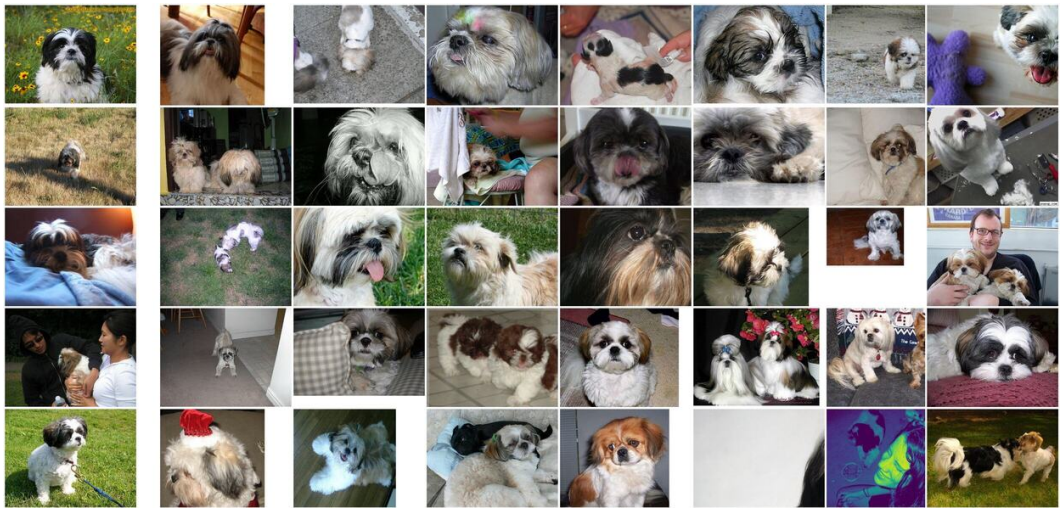
We identify two important parameters for Stable Diffusion, which affect the visual quality of generated images: The guidance scale and the number of diffusion steps. In Fig. 15 we show several examples where we vary one of these two parameters. More specifically, we generate images for the ImageNet synset `n01558993` with class name "robin, American robin, Turdus migratorius", for the simplest case where the prompt is just the class name. We fix the seed to 1947262 and vary either the guidance scale or the number of diffusion steps.

Guidance Scale. From Fig. 15a, we see that increasing the guidance scale coefficient over 10 starts giving hyper-realistic results. When the scale is under 2, we see that many details of the class are not really prominent.

Diffusion Steps. From Fig. 15b, we see that, although with 5 steps the generated images still contain a lot of noise, running 25-50 steps is enough for fully-formed, sharp images to emerge. Since this is a parameter that linearly impacts generation time, increasing the number of steps further than 50 seems excessive.

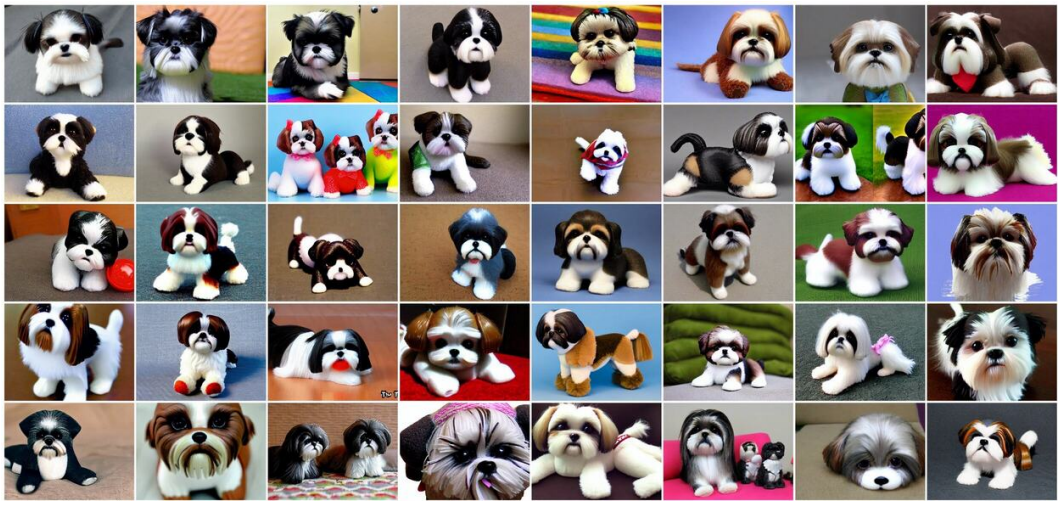
Output Resolution. The resolution that was used during training of the Stable Diffusion models was (512×512).[5] We notice that if one deviates from this training resolution, generated results get worse. We chose to simply switch the aspect ratio to the one for the average ImageNet image and keep the long dimension to 512.

[5] https://github.com/CompVis/stable-diffusion



(a) Real images from ImageNet-1K for class "Shih-Tzu"

(b) Synthetic images with prompt $p_c =$ "c" for class "Shih-Tzu"

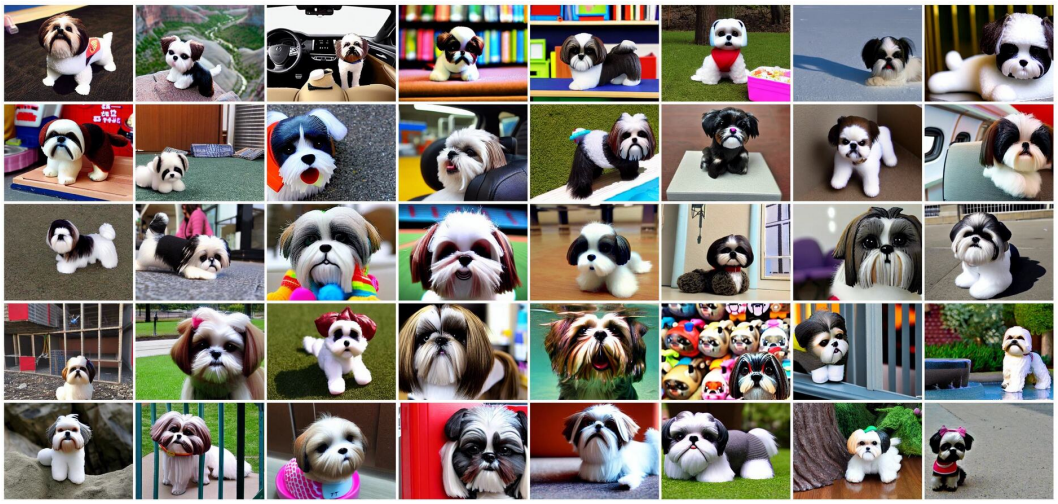


(c) Synthetic images with prompt $p_c =$ "c, h_c" for class "Shih-Tzu"

Figure 7. **Qualitative results for class "Shih-Tzu" to illustrate domain and diversity issues.**

(a) (cont.) Synthetic images with prompt p_c = "c, d_c" for class "Shih-Tzu"

(b) Synthetic images with prompt p_c = "a photo of multiple different c, h_c"

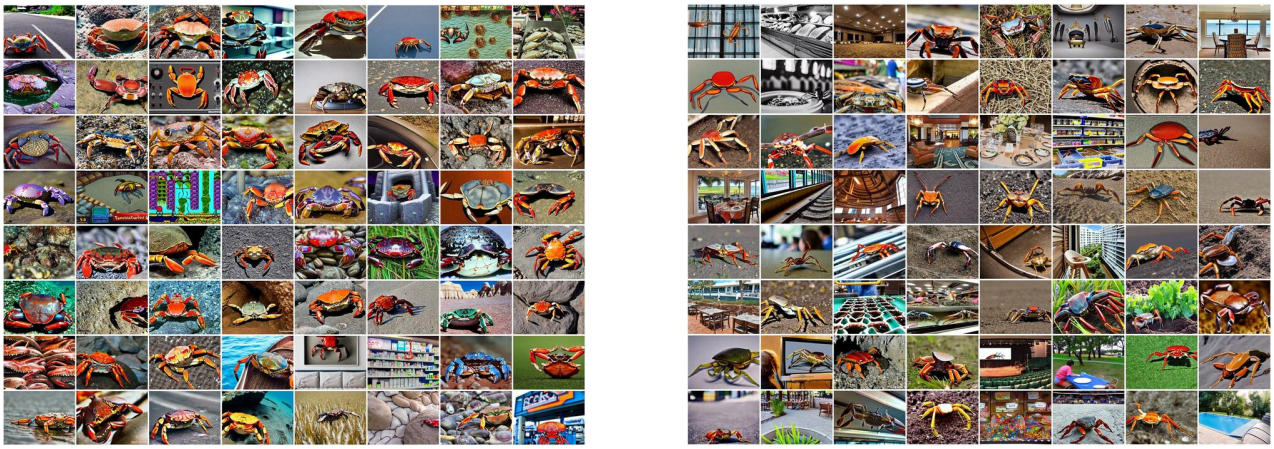


(c) Synthetic images with prompt p_c = "c, h_c inside b"

Figure 8. **(cont.) Qualitative results for class "Shih-Tzu" to illustrate domain and diversity issues.**

(a) Real images from ImageNet-1K for classes “Rock crab” (left) and “Fiddler crab” (right)

(b) Synthetic images with prompt $p_c =$ “c” for classes “Rock crab” (left) and “Fiddler crab” (right)



(c) Synthetic images with prompt $p_c =$ “c, h_c inside b” for classes “Rock crab” (left) and “Fiddler crab” (right)

Figure 9. **Qualitative results for classes “Rock crab” (left) and “Fiddler crab” (right), to illustrate issues around fine-grained and domain specific semantics.**

Figure 10. **Visualization of the 100 ImageNet-100 classes** for the three different datasets: ImageNet-100-Val (real) and two ImageNet-100-SD datasets created with prompts p_c = "c" and p_c = "c, h_c inside b".

Figure 11. **(cont.) Visualization of the 100 ImageNet-100 classes** for the three different datasets: ImageNet-100-Val (real) and two ImageNet-100-SD datasets created with prompts $p_c =$ "c" and $p_c =$ "c, h_c inside b".

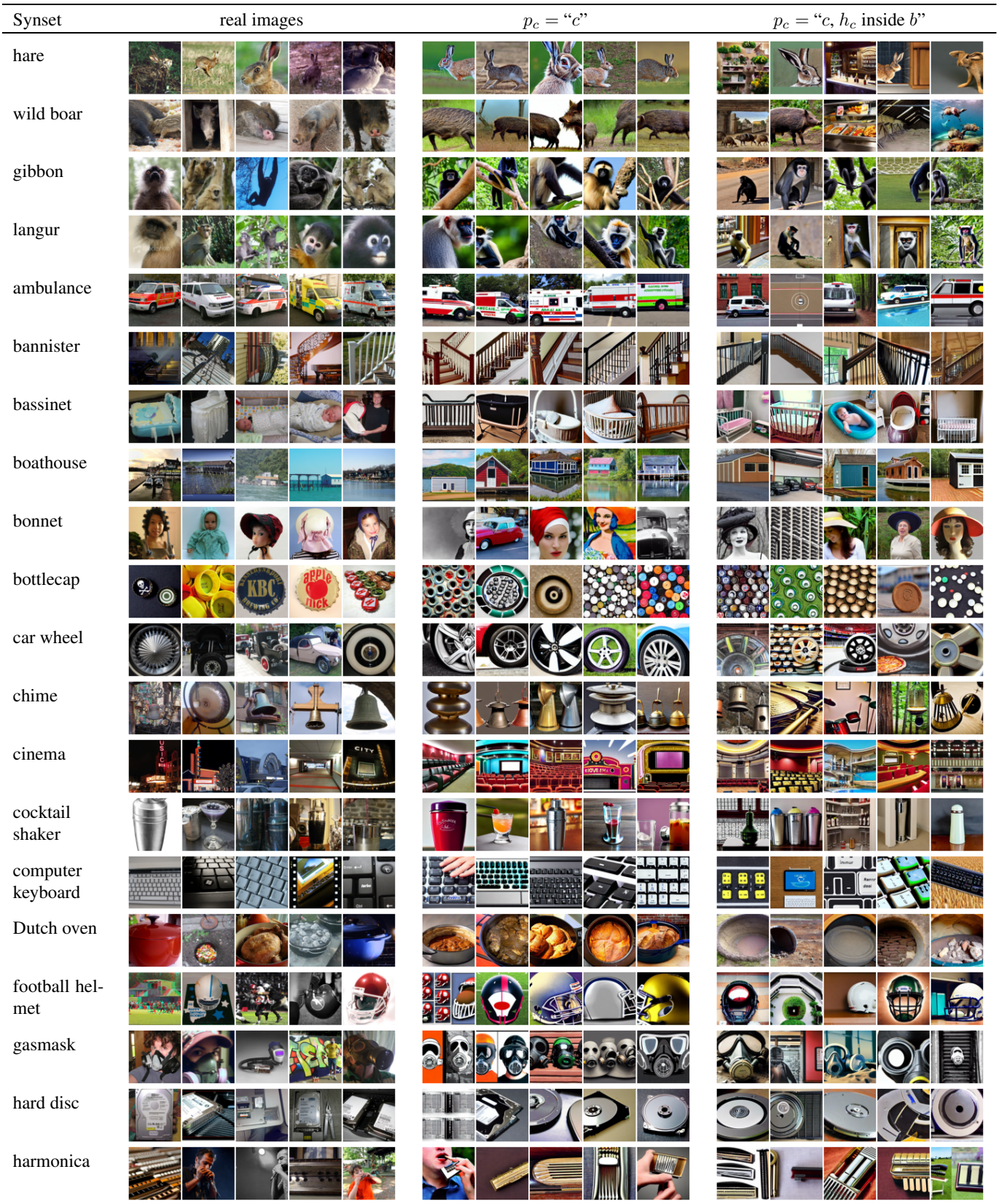
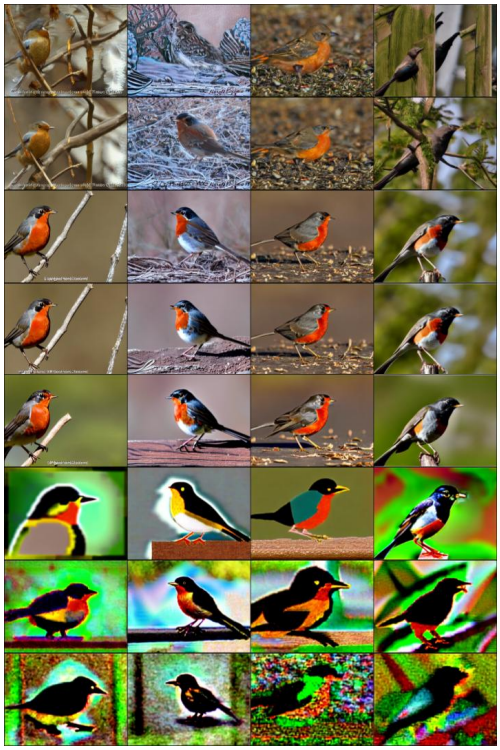


Figure 12. **(cont.) Visualization of the images for the 100 ImageNet-100 classes** in the three different datasets: ImageNet-100-Val (real) and two ImageNet-100-SD datasets created with prompts $p_c =$ "c" and $p_c =$ "c, h_c inside b".

Figure 13. **(cont.) Visualization of the 100 ImageNet-100 classes** for the three different datasets: ImageNet-100-Val (real) and two ImageNet-100-SD datasets created with prompts $p_c =$ "c" and $p_c =$ "c, h_c inside b".

Figure 14. **(cont.) Visualization of the 100 ImageNet-100 classes** for the three different datasets: ImageNet-100-Val (real) and two ImageNet-100-SD datasets created with prompts $p_c =$ "c" and $p_c =$ "c, h_c inside b".



(a) Varying the guidance scale parameter (steps = 50)

(b) Varying the number of diffusion steps (scale = 7.5)

Figure 15. **Qualitative results as we change the guidance scale parameter and the number of diffusion steps during Stable Diffusion generation.** The seed is fixed to `1947262` and the prompt is "robin, American robin, Turdus migratorius". Unless otherwise stated the scale (resp. steps) parameters are set to 7.5 (resp. 50).