Supplementary Material for
Fake it till you make it:
Learning transferable representations from synthetic ImageNet clones

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

Contents

1. Implementation details

In all experiments the encoder f_θ is a ResNet50 [5] encoder, trained for 100 epochs (unless otherwise stated) with mixed precision in PyTorch [19] 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 [3] 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.

2. 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 with synthetic images. For the classification tasks on novel classes, *i.e.*, on the 10 small transfer datasets considered in Tab. 2 of the main paper plus the ImageNet-CoG benchmark in Sec. 3.2, 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. 1.

For transfer learning evaluations, we follow the linear classification protocols from [11, 25]. More precisely, for each of the transfer datasets, we first extract image representations (features) from the pretrained encoders and then train linear logistic regression classifiers using these features. For the larger transfer datasets, *i.e.*, iNaturalist 2018 [27] and iNaturalist 2019 [27] datasets and the CoG levels, we train linear classifiers in PyTorch [19] using SGD, following [25]. For the remaining 8 smaller transfer datasets, we follow [11] and train classifiers using L-BFGS implemented in Scikit-learn [20]. In all cases, we resize the images with bicubic interpolation so that their shortest side is 224 pixels, and then take 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 L-BFGS optimizer) using Optuna [1] over at least 25 trials. Code for evaluations can be found here[1].

3. Extended experimental results

3.1. Impact of data augmentation

We conducted some basic experiments to evaluate the impact of different data augmentation strategies when learning from synthetic datasets. In Tab. 2, we report the performance of models trained on the simplest variant of ImageNet-100-

[1] https://github.com/naver/trex/tree/master/transfer

Dataset	# Classes	# Train samples	# Val samples	# Test samples	Val provided	Test provided
ImageNet test sets (training classes)						
ImageNet-Val [24] (IN-Val)	1000	–	–	50000	–	✓
ImageNet-v2 [22] (IN-v2)	1000	–	–	3×10000	–	✓
ImageNet-Sketch [29] (IN-Sketch)	1000	–	–	50889	–	✓
ImageNet-R [7] (IN-R)	200	–	–	30000	–	✓
ImageNet-A [8] (IN-A)	200	–	–	7500	–	✓
Transfer tasks (novel classes)						
Aircraft [14]	100	3334	3333	3333	✓	✓
Cars196 [12]	196	5700	2444	8041	–	✓
DTD [4]	47	1880	1880	1880	✓	✓
EuroSAT [6]	10	13500	5400	8100	–	–
Flowers [17]	102	1020	1020	6149	✓	✓
Pets [18]	37	2570	1110	3669	–	✓
Food101 [2]	101	68175	7575	25250	–	✓
Pets [18]	397	15880	3970	19850	–	✓
iNaturalist 2018 [27]	8142	437513	–	24426	–	✓
iNaturalist 2019 [27]	1010	265213	–	3030	–	✓
CoG L_1 [25]	1000	895359	223445	50000	–	✓
CoG L_2 [25]	1000	892974	222814	50000	–	✓
CoG L_3 [25]	1000	876495	218708	50000	–	✓
CoG L_4 [25]	1000	886013	221115	50000	–	✓
CoG L_5 [25]	1000	873630	218024	50000	–	✓

Table 1. **Datasets** we use for evaluating our models.

SD, *i.e.*, using the class name as the prompt, utilizing either PyTorch [15, 19] or DINO [3] 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 [28], and consequently can reduce the sim-to-real gap.

Training Dataset	PyTorch [19]	DINO (+ Multi-crop)
ImageNet-100 (real)	*86.6*	*87.4* (↑ 0.80)
ImageNet-100-SD (synthetic)	28.4	43.1 (↑ **14.6**)

Table 2. **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.

3.2. Results on the ImageNet-CoG [25] benchmark

We also evaluated our best ImageNet-SD model on the ImageNet-CoG benchmark introduced in [25] to measure concept generalization. This benchmark consists of evaluations on the set of training classes of ImageNet-1K (IN1K) and five "concept generalization levels", *i.e.*, five IN1K-size datasets of 1000 concepts each. These 5 concept generalization levels contain concepts from the full ImageNet-19K dataset which do not appear in IN1K. Moreover, they are ordered, *i.e.*, each containing concepts that are semantically further and further from the IN1K ones.

Training Dataset	Prompt (p_c) / Model	IN1K	L_1	L_2	L_3	L_4	L_5
ImageNet-1K	*PyTorch [15]*	*75.8*	*67.8*	*63.1*	*58.9*	*58.2*	*52.0*
	RSB-A1 [31]	***79.8***	*69.9*	*65.0*	*60.9*	*59.3*	*52.8*
	DINO [3]	*74.8*	***71.1***	***67.2***	***63.2***	***62.6***	***57.6***
ImageNet-1K-SD	$p_c =$ "c, d_c"	70.4	65.7	61.8	58.5	58.0	52.4

Table 3. **Top-1 accuracy on the ImageNet-CoG benchmark [25]** We report performance for the best ImageNet-1K-SD model from Tab 2. of the main paper (with guidance scale equal to 2).

We follow the evaluation protocol presented in Sec. 2 and report Top-1 accuracy obtained on the test sets of these datasets in Tab. 3. We compare the performance of the best ImageNet-1K-SD model (from Tab. 2 of the main paper) to strong baselines trained on ImageNet-1K like the supervised RSB-A1 [31] and self-supervised DINO [3] models. We observe that on L_5, which is the most challenging level, the performance of the representations learned on synthetic images is comparable to that of learned on real images. As we move towards L_1, we see that the gap between these two models increases in favor of RSB-A1. Finally, after training classifiers (only) using the real images of IN1K, our model reaches 70.4% accuracy, significantly closing the gap to even the most optimized models trained on real data like RSB-A1.

3.3. Analysis of the learned features

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. For this analysis, we used ImageNet-SD models for images that were generated using the default prompt guidance scale of Stable Diffusion, *i.e.*, 7.5. We perform our analysis for ImageNet-100 and using **four metrics**: i) Sparsity, ii) intra-class distance, iii) feature redundancy and iv) 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 listed in Tab. 1, except for the 5 ImageNet-CoG levels. For the sake of this study, we split them into three groups: i) ImageNet-100-Val/v2, ii) ImageNet-100-Sketch/A/R and iii) the 10 transfer datasets (long-tail and small-scale). For each pre-trained 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 [10], 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. 1a. 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. 1b that models trained with more diverse images indeed learn representations with higher intra-class variance.

Feature redundancy. Following [30], we compute feature redundancy, *i.e.*, average pairwise Pearson correlation among dimensions. From Fig. 1c 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 [33] and compute the average coding length per sample on each dataset (see Fig. 1d). We see that models trained on ImageNet-100-Real and ImageNet-100-SD-10x are comparable.

3.4. Impact of guidance scale and diffusion steps

In Fig. 2 we analyse the impact of the guidance scale and diffusion step hyper-parameters of Stable Diffusion [23]. As we discuss in the main paper, a lower guidance scale leads to more visual diversity and that is reflected of performance. Values of 1 to 3 all seem like a good choice. When it comes to the number of diffusion steps, values like 25 and (the default) 50 seem like a safe choice, with 25 being slightly worse, but requiring half the time to extract. Interestingly, using more steps seems to slightly hurt performance on the training classes. It is worth noting that transfer learning performance is surprisingly and consistently high for even 5 diffusion steps. This corroborates recent finding that training on complex but possibly semantically meaningless images like fractals [9] or sinusoidal waves [26] can provide a strong starting point for visual representations that generalize well.

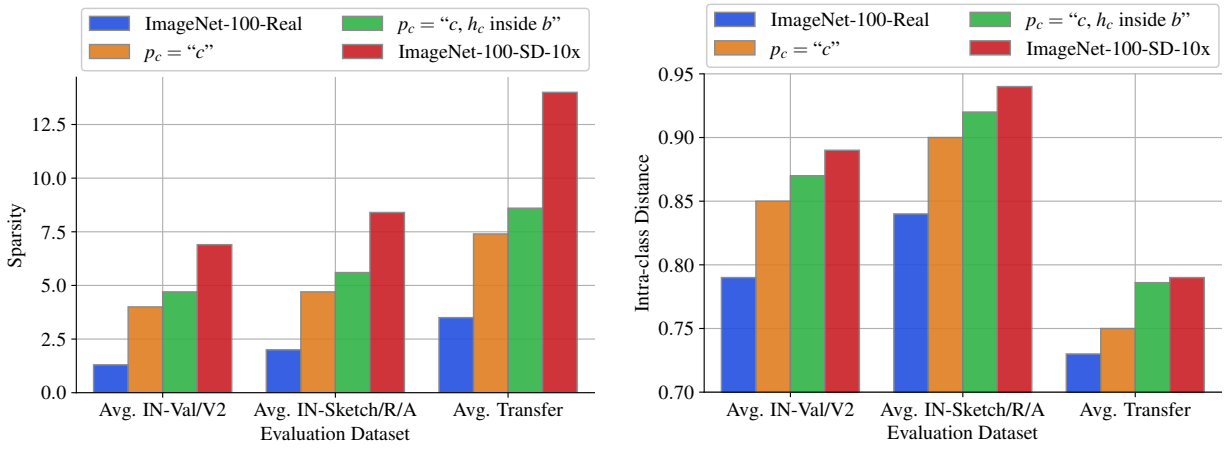
3.5. Prefixing the prompt with domain identifiers

Handcrafted, dataset-level prompt engineering was used for the zero-shot experiments in the CLIP [21] paper. For example they use the prompt template "A photo of a c" as default for classification tasks. For other fine-grained image classification datasets they go one step further and append "a type of {domain}" where {domain}={pet,food,aircraft} for datasets containing pet, food or aircraft classes.

In the main paper, instead presented automatic ways of clarifying the domain, *i.e.*, using extra information from WordNet for each class. In Tab. 4 we present some preliminary results when using generic prompt templates like "a photo of c" and "an image of c" as input to the Stable Diffusion v1.4 model. We found them to decrease performance for ImageNet-100.

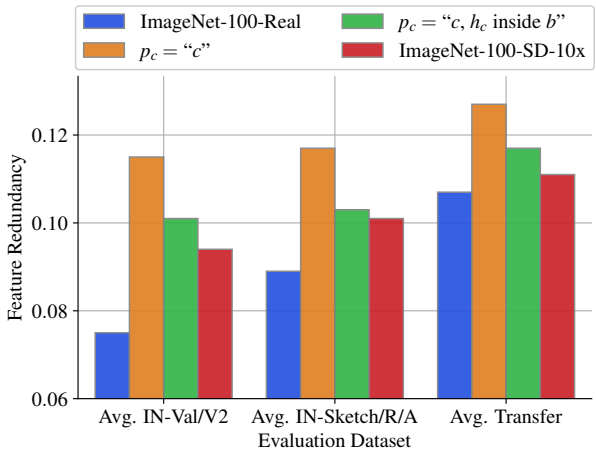
p_c	"c"	"a photo of c"	"an image of c"
Top-1 Acc.	**64.8**	59.5	58.3

Table 4. Top-1 Accuracy on ImageNet-100 when prepending the prompt with domain identifiers. Guidance scale is equal to 2.0.

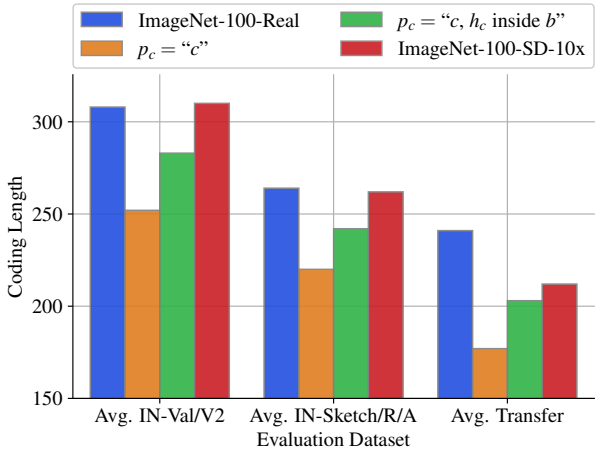


(a) **Feature Sparsity**

(b) **Intra-class ℓ_2-distance**



(c) **Feature redundancy**



(d) **Coding length**

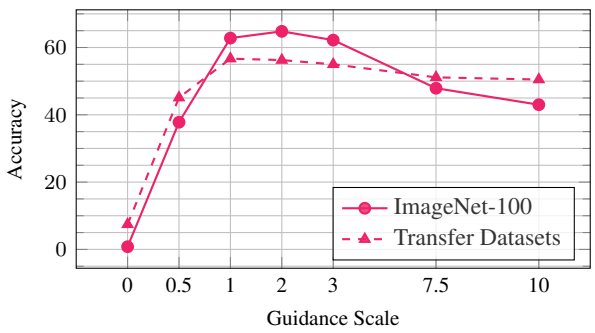
Figure 1. **Feature analyses** for models. We perform these analyses on top of features extracted from pretrained encoders f trained on either real or synthetic data for ImageNet-100 (training data is specified in the legends of the subfigures). For the purpose of this study, we use synthetic data generated with guidance scale equal to 7.5. *Sparsity* is measured by the percentage of dimensions close to zero [10]. *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* [30] 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* [33] 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.

3.6. Additional scaling plots for synthetic data

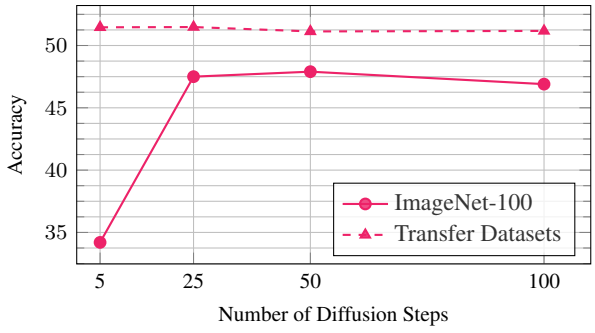
In Fig. 3 we report accuracy when training on ImageNet-100 using $(1/10)$-th to $50\times$ images, relative to the real dataset size. Fig. 3a suggests that generating more images with basic prompts might not be enough, and that a performance leap will require advanced prompt engineering. We consider a study on scaling synthetic datasets is important, but beyond the scope of this paper. Note that Fig. 3b is also shown in the main paper and repeated here for completeness.

3.7. Additional spider plots

In Fig. 4 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 and similar to the teaser figure in the main paper, we report top-1 accuracy averaged over the transfer datasets in each of the following three groups: **(a)** eight common small-scale datasets (Aircraft [14], Cars196 [12], DTD [4], EuroSAT [6], Flowers [17], Pets [18], Food101 [2], SUN397 [32]), **(b)** two

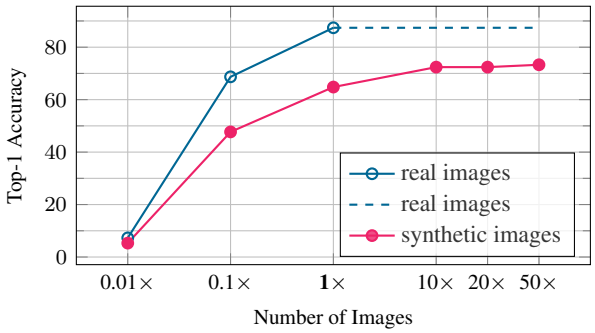


(a) Impact of the guidance scale parameter

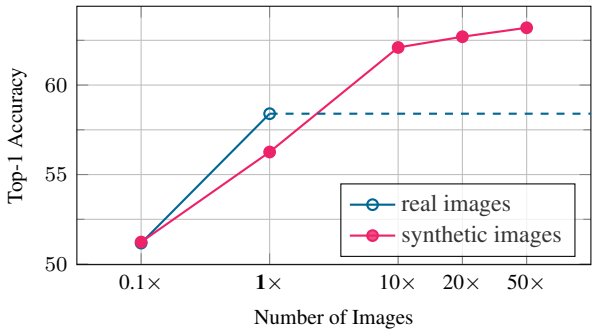


(b) Impact of the nubmer of diffusion steps

Figure 2. **Impact of the guidance scale parameter and number of diffusion steps.** Top-1 Accuracy on ImageNet-100 and averaged over 10 transfer datasets for $p_c =$ "c, d_c". In the left plot, steps are set to 50, in the right plot guidance scale is 7.5.



(a) Top-1 accuracy on ImageNet-100.



(b) Top-1 accuracy on the 10 transfer datasets from **??** of the main paper.

Figure 3. **Scaling the number of training images.** Accuracy when training on ImageNet-100 using $(1/10)$-th to $50\times$ images (relative to the real dataset size). Fig. 3b is also shown in the main paper.

long-tail datasets (iNat2018 [27] and iNat2019 [27]), and **(c)** the five datasets ("levels") of the CoG benchmark [25].

4. Extended qualitative results

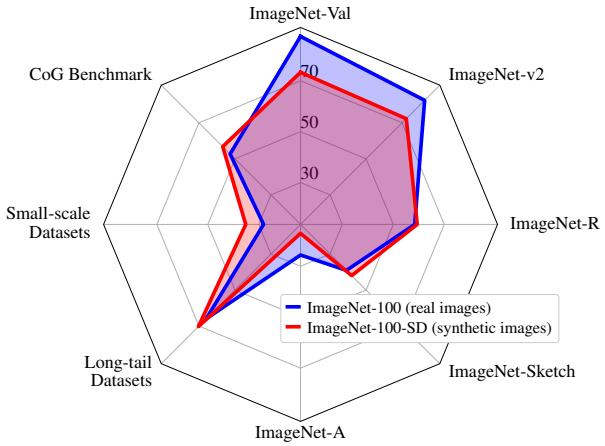
In this section, we provide additional qualitative results. First we show random images for *all* ImageNet-100 classes from three datasets: ImageNet-100-Val (real images) and two ImageNet-100-SD datasets generated by the prompts $p_c =$ "c" and $p_c =$ "c, h_c inside b". Then we discuss in more detail several types of issues that we observed in these synthetic images. Unless otherwise stated, the guidance scale used is 7.5.

Qualitative results for *all* ImageNet-100 classes. In Fig. 9, we show a few random images from each of the 100 classes in ImageNet-100, for three datasets: i) The real images from ImageNet-100, ii) synthetic images generated by a simple prompt, which is only composed of the name of the class, and iii) synthetic images generated with guidance scale equal to 2.0 and 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.

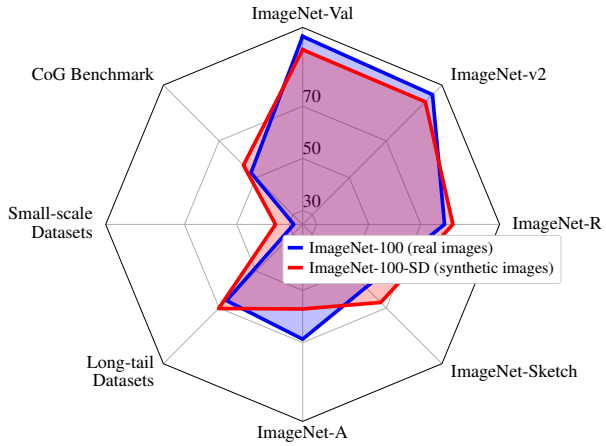
Showcasing domain and diversity issues. We also show extended results for three classes in order to illustrate issues related to the domain and diversity. Fig. 8 compares generated images between two fine-grained classes of crabs, while Fig. 7 shows many images from multiple different generated datasets for a single dog class. We discuss both figures in the next subsections.

4.1. Semantic errors

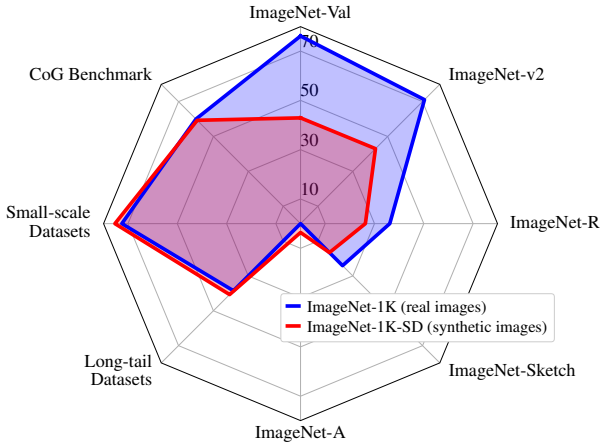
From closely inspecting the generated images we can see that there exists two classes for which the prompt $p_c =$ "c" produces images of the wrong semantics: For the classes "papillon" and "wing", we see the generated images in the middle column of Fig. 9 to be wrong due to *polysemy* associated with 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



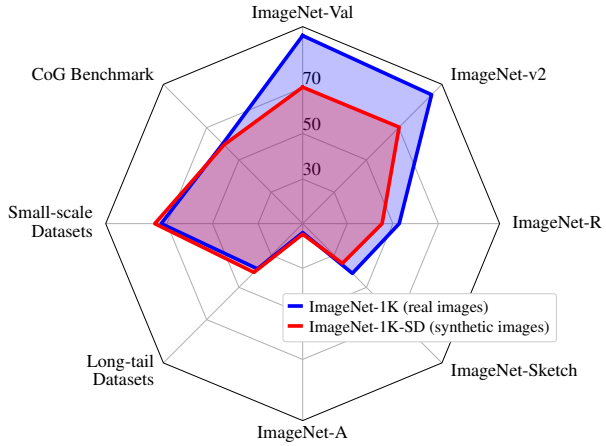
(a) Top-1 accuracy, training on **ImageNet-100**.



(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 4. **Performance card of models** trained on either real or synthetic data for 100 classes of ImageNet-100 (Figs. 4a and 4b) and for all the 1000 classes of ImageNet-1K (Figs. 4c and 4d). 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 [23] using $p_c =$ "c, h_c inside b" as the prompt. In Figs. 4a and 4c and in Figs. 4b and 4d 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. 4d corresponds to Fig 1 of the main paper.

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 [16], the hypernym for "shih-tzu" is "toy dog" something that results in dog-shaped toys in many of the generated images (see also Fig. 7). Another example is the class "boathouse", where appending the parent class "shed" leads to sheds that are not inside a body of water.

4.2. 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.[2] We disabled this module when generating images for the ImageNet synsets as we wanted to study the model

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

as-is first, and to understand the problem.

We thoroughly inspected all classes of ImageNet-100 and observed minor NSFW issues with 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 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 [34] also resulted in nudity. Overall, we want to emphasize that the Stable Diffusion models we tested were all highly susceptible to generate such content.

4.3. 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. 8 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. 8a.

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. 8b, 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 [13], 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.

4.4. 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. 8c, and a few more spread in that figure, or the background samples for class "reel" in Fig. 9. 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. 8c for class "pedestal", a pirate themed bedroom for class "pirate", green shirts for "green mamba", or the red-ish produce stand for "red fox".

4.5. 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. 9. This is also visible for the crab classes in Fig. 8b, or the Shih-tzu class in Fig. 6b, Fig. 7a and Fig. 7b. We see that such issues are partially solved when lowering the guidance scale and relying less to the prompt, or using backgrounds (*e.g.*, the right-most set of images in Fig. 9). 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. 9.

4.6. 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.

4.7. 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. 5 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. 5a, 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. 5b, 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).[3] 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.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proc. ICKDDM*, 2019. 1

[2] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – Mining discriminative components with random forests. In *Proc. ECCV*, 2014. 2, 4

[3] 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. 1, 2

[4] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proc. CVPR*, 2014. 2, 4

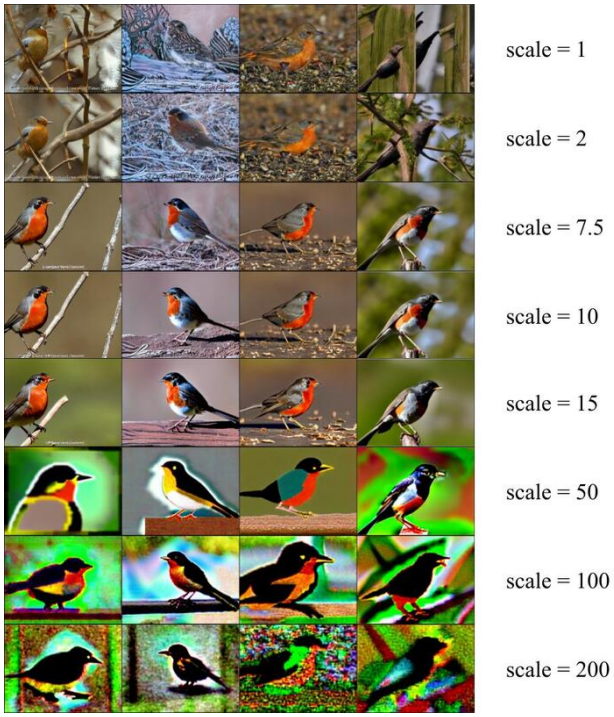
[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 1

[6] 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. 2, 4

[7] 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. 2

[8] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proc. CVPR*, 2021. 2

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



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

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

Figure 5. **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).

[9] Hirokatsu Kataoka, Kazushige Okayasu, Asato Matsumoto, Eisuke Yamagata, Ryosuke Yamada, Nakamasa Inoue, Akio Nakamura, and Yutaka Satoh. Pre-training without natural images. *IJCV*, 2022. 3

[10] Simon Kornblith, Ting Chen, Honglak Lee, and Mohammad Norouzi. Why do better loss functions lead to less transferable features? In *Proc. NeurIPS*, 2021. 3, 4

[11] Simon Kornblith, Jonathon Shlens, and Quoc Le. Do

better ImageNet models transfer better? In *Proc. CVPR*, 2019. 1

[12] Jonathan Krause, Jia Deng, Michael Stark, and Fei-Fei Li. Collecting a large-scale dataset of fine-grained cars. In *Proc. ICCV-W*, 2013. 2, 4

[13] Alexandra Sasha Luccioni and David Rolnick. Bugs in the data: How ImageNet misrepresents biodiversity. *arXiv:2208.11695*, 2022. 7

[14] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv:1306.5151*, 2013. 2, 4

[15] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proc. ACM-ICM*, 2010. 2

[16] George A Miller. Wordnet: A lexical database for English. *Commun. ACM*, 1995. 6

[17] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Proc. ICCVGIP*, 2008. 2, 4

[18] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *Proc. CVPR*, 2012. 2, 4

[19] 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. 1, 2

[20] 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. 1

[21] 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. 3

[22] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *Proc. ICML*, 2019. 2

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

[24] 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. 2

[25] Mert Bulent Sariyildiz, Yannis Kalantidis, Diane Larlus, and Karteek Alahari. Concept generalization in visual representation learning. In *Proc. ICCV*, 2021. 1, 2, 5

[26] Sora Takashima, Ryo Hayamizu, Nakamasa Inoue, Hirokatsu Kataoka, and Rio Yokota. Visual atoms: Pre-training vision transformers with sinusoidal waves. *arXiv:2303.01112*, 2023. 3

[27] 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. 1, 2, 5

[28] Riccardo Volpi, Diane Larlus, and Gregory Rogez. Continual adaptation of visual representations via domain randomization and meta-learning. In *Proc. CVPR*, 2021. 2

[29] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Proc. NeurIPS*, 2019. 2

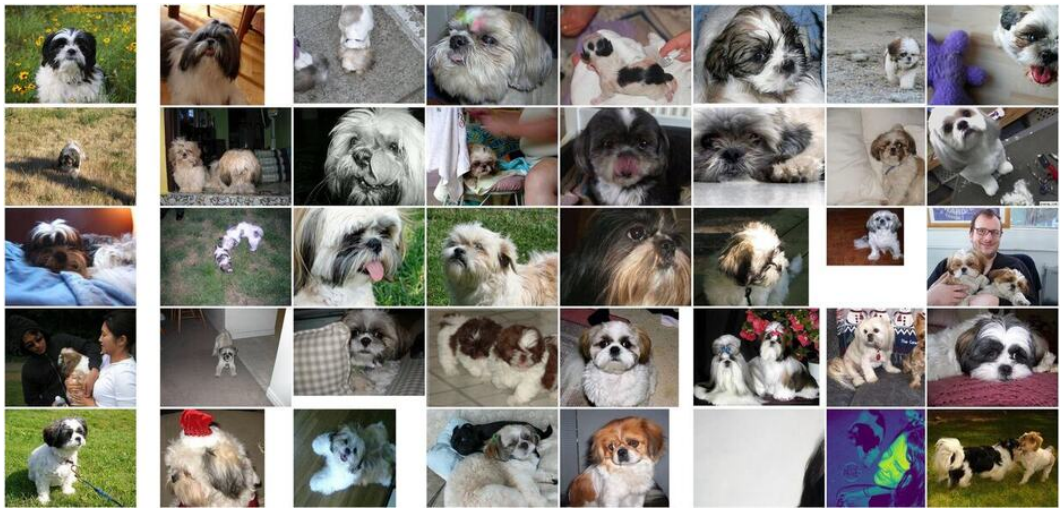
[30] Yizhou Wang, Shixiang Tang, Feng Zhu, Lei Bai, Rui Zhao, Donglian Qi, and Wanli Ouyang. Revisiting the transferability of supervised pretraining: an MLP perspective. In *Proc. CVPR*, 2022. 3, 4

[31] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. In *Proc. NeurIPS-W*, 2021. 2

[32] 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. 4

[33] 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. 3, 4

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



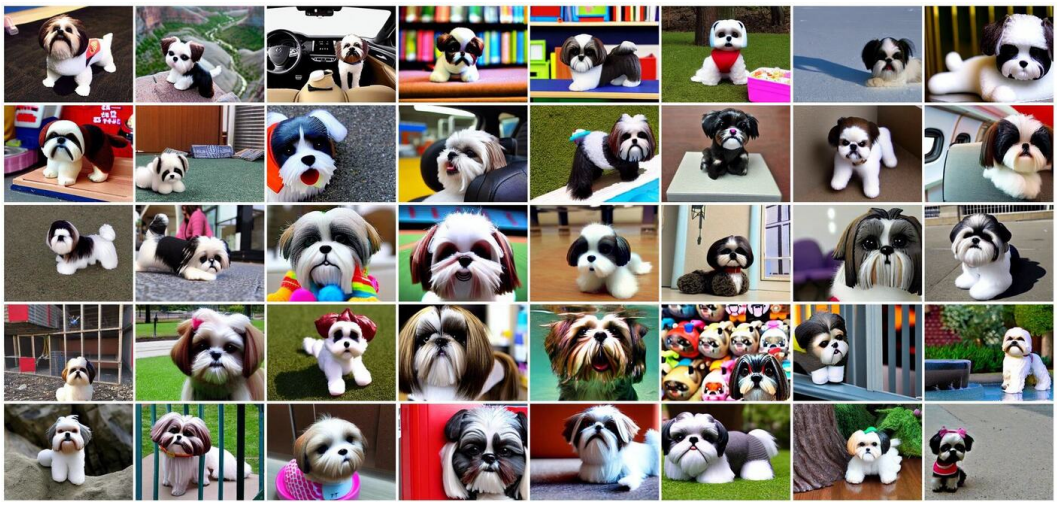
(a) Real images from ImageNet-1K for class "Shih-Tzu"

(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 6. **Qualitative results for class "Shih-Tzu"** to illustrate domain and diversity issues. Guidance scale is equal to 7.5.

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

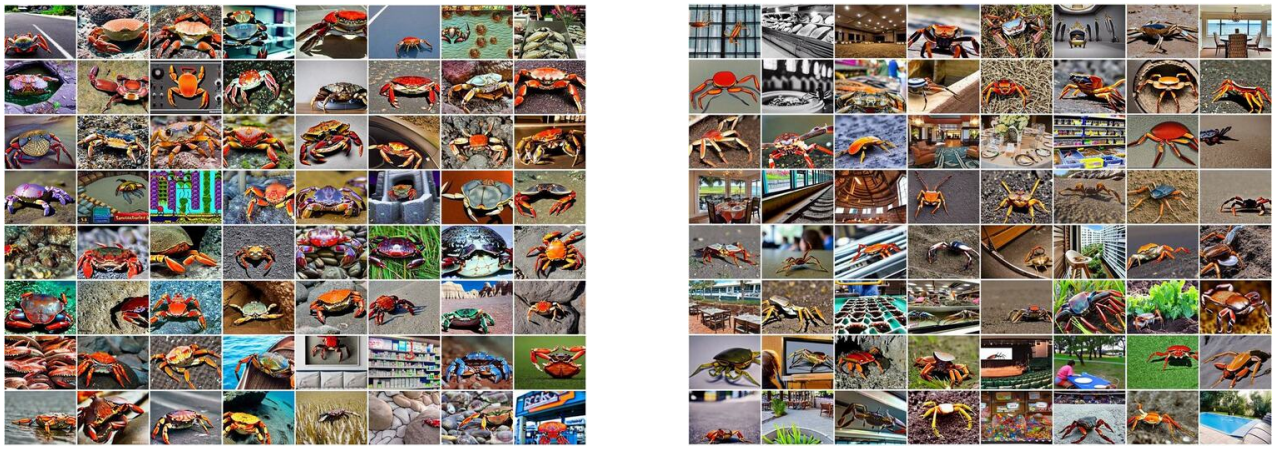


(b) Synthetic images with prompt p_c = "c, h_c inside b"

Figure 7. **(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 8. **Qualitative results for classes “Rock crab” (left) and “Fiddler crab” (right)**, to illustrate issues around fine-grained and domain specific semantics. Guidance scale is equal to 7.5.

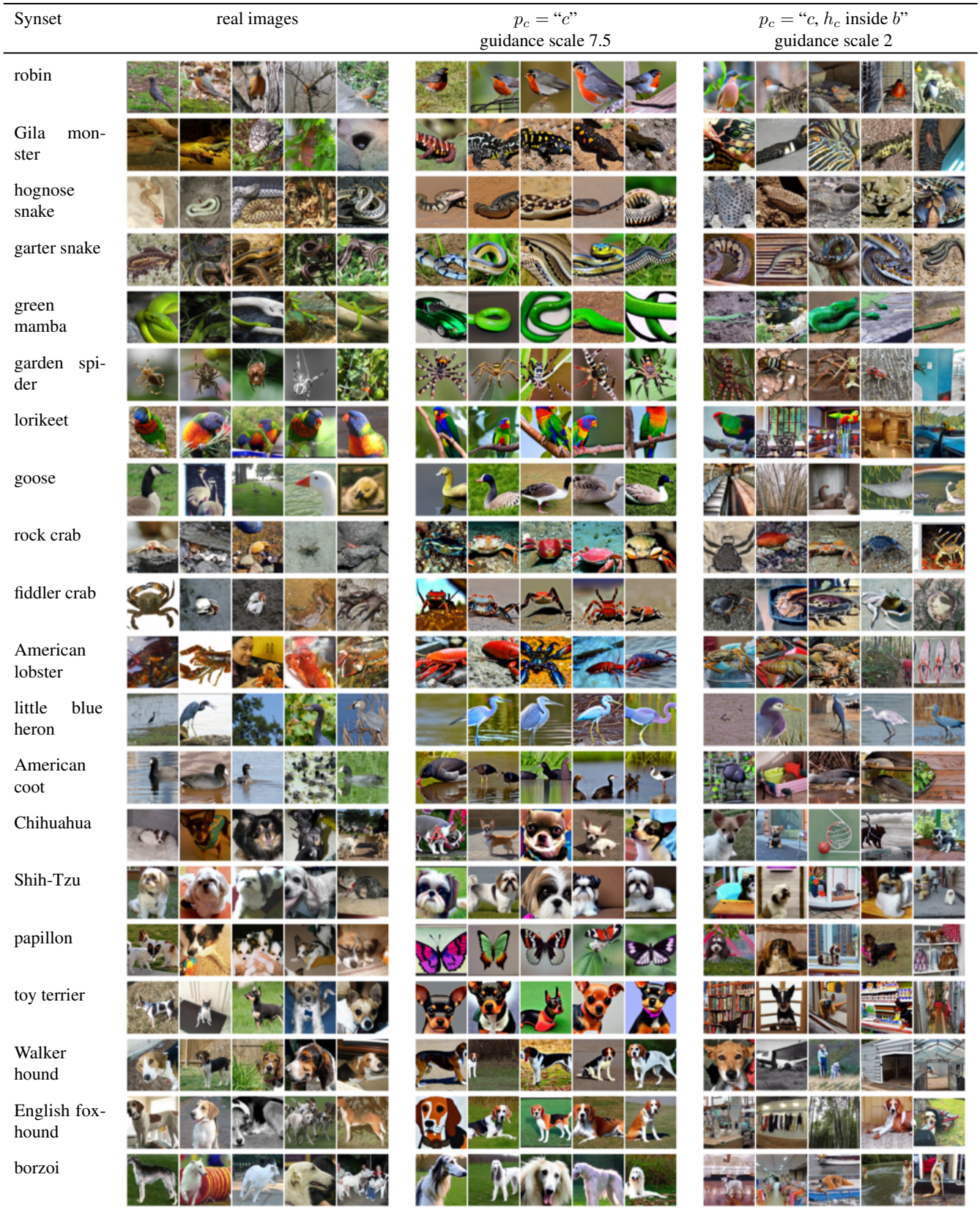


Figure 9. **Visualization of the 100 ImageNet-100 classes** for the three different datasets: ImageNet-100-Val (real) and two ImageNet-100-SD datasets created with prompts $p_c =$ "c" and $p_c =$ "c, h_c inside b".

Figure 10. **(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 11. **(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".

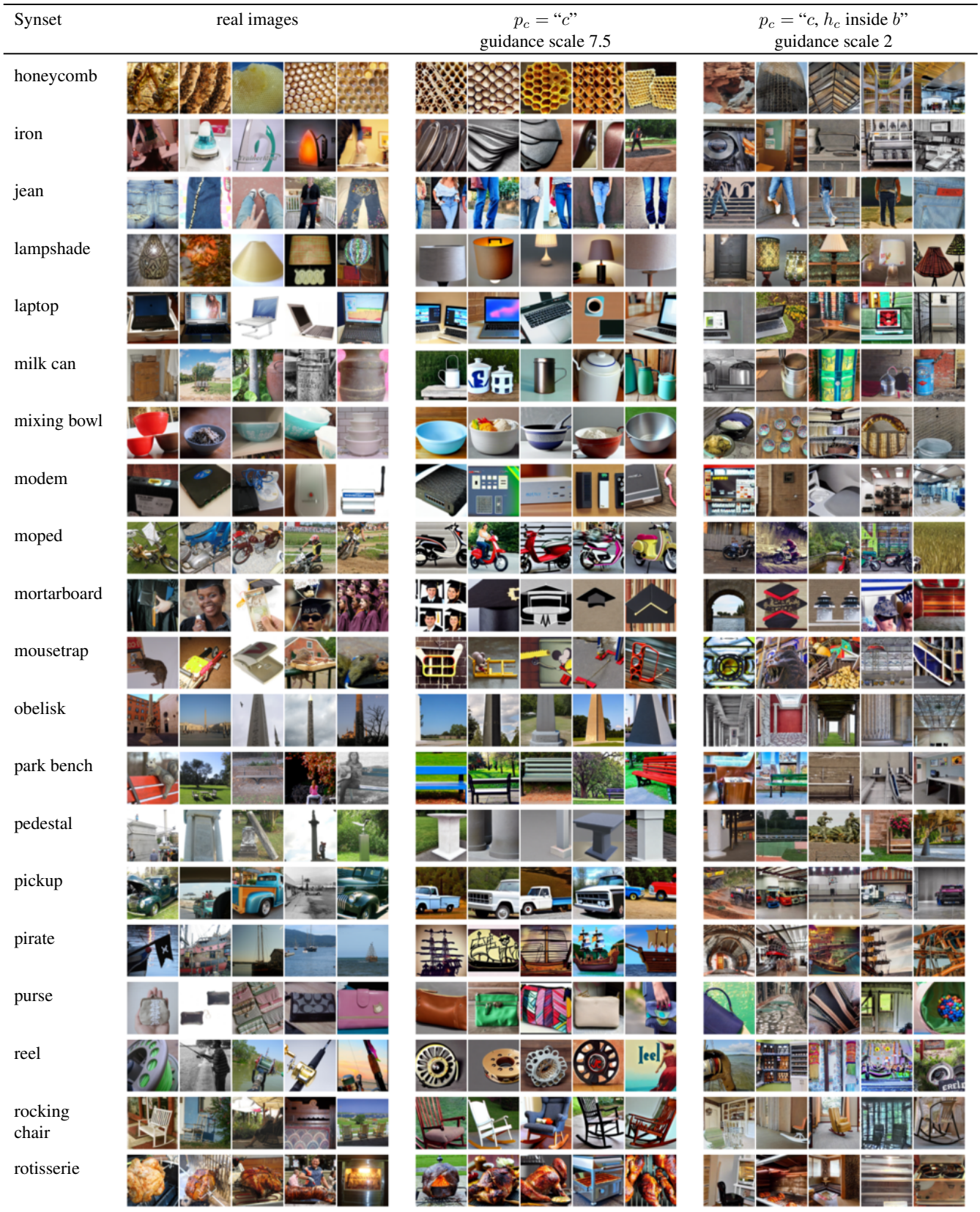


Figure 12. **(cont.) Visualization of the 100 ImageNet-100 classes** for the three different datasets: ImageNet-100-Val (real) and two ImageNet-100-SD datasets created with prompts $p_c =$ "c" and $p_c =$ "c, h_c inside b".

Figure 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".