IS SYNTHETIC DATA FROM GENERATIVE MODELS READY FOR IMAGE RECOGNITION?

Ruifei He[1*] **Shuyang Sun**[2] **Xin Yu**[1] **Chuhui Xue**[3] **Wenqing Zhang**[3] **Philip Torr**[2]
Song Bai[3†] **Xiaojuan Qi**[1†]
[1]The University of Hong Kong [2]University of Oxford [3]ByteDance

ABSTRACT

Recent text-to-image generation models have shown promising results in generating high-fidelity photo-realistic images. Though the results are astonishing to human eyes, how applicable these generated images are for recognition tasks remains under-explored. In this work, we extensively study whether and how synthetic images generated from state-of-the-art text-to-image generation models can be used for image recognition tasks, and focus on two perspectives: synthetic data for improving classification models in data-scarce settings (*i.e.* zero-shot and few-shot), and synthetic data for large-scale model pre-training for transfer learning. We showcase the powerfulness and shortcomings of synthetic data from existing generative models, and propose strategies for better applying synthetic data for recognition tasks. Code: https://github.com/CVMI-Lab/SyntheticData.

1 INTRODUCTION

Over the past decade, deep learning powered by large-scale annotated data has revolutionized the field of image recognition. However, it is costly and time-consuming to manually collect a large-scale labeled dataset, and recent concerns about data privacy and usage rights further hinder this process. In parallel, generative models that aim to model real-data distributions can now produce high-fidelity photo-realistic images. In particular, recent text-to-image generation models (Nichol et al., 2021; Ramesh et al., 2022; Saharia et al., 2022b) have made major breakthroughs in synthesizing high-quality images from text descriptions. This promotes us to ask: is synthetic data from generative models ready for image recognition tasks?

There are a few early attempts at exploring synthetic data from generative models for image recognition tasks. Besnier et al. (2020) use a class-conditional GAN (BigGAN (Brock et al., 2018) trained for ImageNet-1000 classes) to generate images for training image classifiers. Zhang et al. (2021) leverage StyleGAN (Karras et al., 2019) to produce synthetic labeled data for object-part segmentation. Jahanian et al. (2021) manipulate the latent space of a GAN model to produce multi-view images for contrastive learning. Albeit promising, early works either address tasks on a small scale or only for a specific setting. Besides, they all focus on GAN-based models and none have explored the recent revolutionary text-to-image generation models, which hold more promises to benefit recognition tasks.

In this paper, we present the first study on the state-of-the-art text-to-image generation models for image recognition. With the power of text-to-image generation, we could hopefully not only generate massive high-quality labeled data, but also achieve domain customization by generating synthetic data targeted for a specific label space, *i.e.* the label space of a downstream task. Our study is carried out on one open-sourced text-to-image generation model, GLIDE (Nichol et al., 2021) [1]. We attempt to uncover the benefits and pitfalls of synthetic data for image recognition through the lens of investigating the following two questions: 1) is synthetic data from generative models ready for improving classification models? 2) whether synthetic data can be a feasible source for transfer

[*] Part of the work is done during an internship at ByteDance. Email: ruifeihe@eee.hku.hk

[†] Corresponding authors: songbai.site@gmail.com, xjqi@eee.hku.hk

[1] At the beginning of this project, GLIDE is the only open-sourced text-to-image synthesis model that also delivers high-quality synthesis results.

learning (*i.e.* model pre-training)? It is worth noting that for 1), we only studied the zero-shot and few-shot settings because the positive impact of synthetic data diminishes as more shots are present. And, we build most of our investigations on the state-of-the-art method CLIP (Radford et al., 2021) with the feature extractor initialized with large-scale pre-trained weights frozen.

Our Findings. First, in the zero-shot setting, *i.e.* no real-world data are available, we demonstrate that synthetic data can significantly improve classification results on 17 diverse datasets: the performance is increased by 4.31% in top-1 accuracy on average, and even improved by as much as 17.86% on the EuroSAT dataset. To better leverage synthetic data in this setting, we also investigate useful strategies to increase data diversity, reduce data noise, and enhance data reliability. This is achieved by designing diversified text prompts and measuring the correlation of text and synthesized data with CLIP features.

Second, in the few-shot setting, *i.e.* a few real images are available, albeit not as significant as in the zero-shot task, synthetic data are also shown to be beneficial and help us achieve a new state of the art. Our observation shows that the domain gap between synthetic data and downstream task data is one challenge on further improving the effectiveness of synthetic data on classifier learning. Fortunately, in this setting, the accessibility of real data samples can provide useful information about the data distribution of the downstream task. We thus propose to use real images as guidance in the generation process to reduce domain gaps and improve effectiveness.

Third, in large-scale model pre-training for transfer learning, our study shows that synthetic data are suitable and effective for model pre-training, delivering superior transfer learning performance and even outperforming ImageNet pre-training. Especially, synthetic data work surprisingly well in unsupervised model pre-training. We also demonstrate that by increasing the label space (*i.e.* text prompts) for data generation, the enlarged data amount and diversity could further bring performance boosts. Besides, synthetic data can work collaboratively with real data (*i.e.* ImageNet) where we obtain improved performance when the model is initialized with ImageNet pre-trained weights.

2 Related works

Synthetic Data for Image Recognition. There are mainly two forms of synthetic data for image recognition, *i.e.* 1) synthetic datasets generated from a traditional simulation pipeline; 2) synthetic images output from generative models.

The first type, synthetic datasets (Dosovitskiy et al., 2015; Peng et al., 2017; Richter et al., 2016), are usually generated from a traditional pipeline with a specific data source, *e.g.*synthetic 2D renderings of 3D models or scenes from graphics engines. However, this traditional way of generating synthetic datasets has several drawbacks: 1) manually defined pipeline generated synthetic data may have a certain gap with real-world data; 2) taking up huge physical space to store and huge cost to share and transfer; 3) data amount and diversity bounded by the specific data source.

Compared with synthetic datasets, generative models are a more efficient means of synthetic data representation, exhibiting favorable advantages: 1) could produce high-fidelity photorealistic images closer to real data since they are trained on real-world data; 2) highly condensed compared to synthetic data itself, and take up much reduced storage space; 3) potentially unlimited synthetic data size. Only recently, few works attempt to explore synthetic data generated from generative models for image recognition. Besnier et al. (2020) use a class-conditional GAN to train classifiers of the same classes. Zhang et al. (2021) leverage the latent code of StyleGAN (Karras et al., 2019) to produce labels for object part segmentation. While they achieve promising results, both works are task-wise and only employed on a small scale. Jahanian et al. (2021) use a GAN-based generator to generate multiple views to conduct unsupervised contrastive representation learning. These works, however, explore upon the traditional GAN-based models; in contrast, our work investigates with the best released text-to-image generation model, which demonstrates new customization ability for different downstream label space.

Text-to-Image Diffusion Models. Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Nichol & Dhariwal, 2021) have recently emerged as a class of promising and powerful generative models. As a likelihood-based model, the diffusion model matches the underlying data distribution $q(x_0)$ by learning to reverse a noising process, and thus novel images can be sampled from a prior Gaussian distribution via the learned reverse path. Because of the high sample quality, good mode coverage and promising training stability, diffusion models are quickly becoming a new trend in both unconditional (Ho et al., 2020; Nichol & Dhariwal, 2021; Ho et al., 2022) and conditional (Dhariwal & Nichol, 2021; Rombach et al., 2022; Lugmayr et al., 2022; Saharia et al., 2022a; Meng et al., 2021; Saharia et al., 2022c) image synthesis fields.

In particular, text-to-image generation can be treated as a conditional image generation task that requires the sampled image to match the given natural language description. Based upon the formulation of the diffusion model, several text-to-image models such as Stable diffusion (Rombach et al., 2022), DALL-E2 (Ramesh et al., 2022), Imagen (Saharia et al., 2022b) and GLIDE (Nichol et al., 2021) deliver unprecedented synthesis quality, largely facilitating the development of the AI-for-Art community. Despite achieving astonishing perceptual results, their potential utilization for high-level tasks is yet under-explored. In this paper, we utilize the state-of-the-art model GLIDE and showcase its powerfulness and shortcomings for synthesizing data for recognition tasks.

3 Is synthetic data ready for image recognition?

In the following sections, we answer the question by studying whether synthetic data can benefit recognition tasks and how to better leverage synthetic data to address different tasks. We carry out our exploration through the lens of two basic settings with three tasks: synthetic data for improving classification models in the data-scarce setting (*i.e.* zero-shot and few-shot) (see Sec. 3.1 and Sec. 3.2) and synthetic data for model pre-training for transfer learning (see Sec. 3.3).

Model Setup for Data-scarce (*i.e.* Zero-shot and Few-shot) Image Classification. As CLIP (Radford et al., 2021) is the state-of-the-art approach for zero-shot learning, we conduct our study for zero-shot and few-shot settings upon pre-trained CLIP models, aiming to better understand synthetic data upon strong baselines. There have been a few attempts on better tuning pre-trained CLIP for data-scarce image classification, such as CoOp (Zhou et al., 2022b), CLIP Adapter (Gao et al., 2021), and Tip Adapter (Zhang et al., 2022), where the image encoder is frozen for better preserving the pre-trained feature space. We argue that different tuning methods could all be regarded as different ways of learning classifier weights, *e.g.* CoOp optimizes learnable prompts for better learning classifiers. Here, we adopt a simple tuning method, Classifier Tuning (**CT**) (Wortsman et al., 2022), which directly tunes a classifier attached to the CLIP image encoder and initializes the classifier weights with pre-trained text embedding. We empirically show that **CT** performs comparably with other CLIP tuning methods. Compared with complex designed tuning methods, we hope to use a simpler tuning method for better investigate the effectiveness of synthetic data.

3.1 Is Synthetic data ready for Zero-shot image recognition?

Our aim is to investigate to what degree synthetic data are beneficial to zero-shot tasks and how to better leverage synthetic data for zero-shot learning.

Zero-shot Image Recognition. We study the inductive zero-shot learning setting where no real training images of the target categories are available. CLIP models are pre-trained with large-scale image-caption pairs, and the similarities between paired image features (from an image-encoder g) and text features (from a text-encoder h) are maximized during pre-training. The pre-trained feature extractor can then be used to solve zero-shot tasks where given an image, its features from g are compared with text features of different classes from h and the image is further assigned to the class that has the largest similarity in the CLIP text-image feature space.

Synthetic Data for Zero-shot Image Recognition. Though CLIP models exhibit strong zero-shot performance thanks to the large-scale vision-language dataset for pre-training, there are still several shortcomings when the model is deployed for a downstream zero-shot classification task, which may be attributed to unavoidable data noise in CLIP's pre-training data or the label space mismatch between pre-training and the zero-shot task. Hence, with a given label space for a zero-shot task, we study whether synthetic data can be used to better adapt CLIP models for zero-shot learning.

How to generate the data? Given a pre-trained text-to-image generation model, to synthesize novel samples, the basic (**B**) strategy is to use the label names of the target categories to build the language input and generate a corresponding image. Then, the paired label names and synthesized data can be employed to train the classifier with the feature extractor frozen.

Dataset	Task	CLIP	CLIP+SYN
CIFAR-10	object-level	70.31	80.06 (+9.75)
CIFAR-100	object-level	35.35	45.69 (+10.34)
Caltech101	object-level	86.09	87.74 (+1.65)
Caltech256	object-level	73.36	75.74 (+2.38)
ImageNet	object-level	60.33	60.78 (+0.45)
SUN397	scene-level	58.51	60.07 (+1.56)
Aircraft	fine-grained	17.34	21.94 (+4.60)
Birdsnap	fine-grained	34.33	38.05 (+3.72)
Cars	fine-grained	55.63	56.93 (+1.30)
CUB	fine-grained	46.69	56.94 (+10.25)
Flower	fine-grained	66.08	67.05 (+0.97)
Food	fine-grained	80.34	80.35 (+0.01)
Pets	fine-grained	85.80	86.81 (+1.01)
DTD	textures	42.23	43.19 (+0.96)
EuroSAT	satellite images	37.51	55.37 (+17.86)
ImageNet-Sketch	robustness	33.29	36.55 (+3.26)
ImageNet-R	robustness	56.16	59.37 (+3.21)
Average	/	55.13	59.47 (+4.31)

Table 1: **Main Results on Zero-shot Image Recognition.** All results are top-1 accuracy on test set.

How to enrich diversity? Only using the label names as inputs might limit the diversity of synthesized images and cause bottlenecks for validating the effectiveness of synthetic data. Hence, we leverage an off-the-shelf word-to-sentence T5 model (pre-trained on "Colossal Clean Crawled Corpus" dataset (Raffel et al., 2020) and finetuned on CommonGen dataset (Lin et al., 2019)) to increase the diversity of language prompts and the generated images, namely language enhancement (**LE**), hoping to better unleash the potential of synthesized data. Concretely, we input the label name of each class to the word-to-sentence model which generates diversified sentences containing the class names as language prompts for the text-to-image generation process. For example, if the class label is "airplane", then the enhanced language prompt from the model could be "a white airplane hovering over a beach and a city". The enhanced text descriptions introduce rich context descriptions.

How to reduce noise and enhance robustness? It's unavoidable that the synthesized data may contain low-quality samples. This is even more severe in the setting with language enhancement as it may introduce undesired items into language prompts (see Appendix for more examples). Therefore, we introduce a CLIP Filter (**CF**) strategy to rule out these samples. Specifically, CLIP zero-shot classification confidence is used to assess the quality of synthesized data, and the low-confidence ones are removed which have a high potential to be unreliable. Besides, as soft-target is more robust than hard-target in countering sample noise, we study whether soft cross-entropy loss (**SCE**, details in Appendix) which uses the normalized clip scores as a target could be used to enhance robustness against data noise.

Experiment Setup. We select 17 diverse datasets covering object-level (CIFAR-10 and CIFAR-100 ((Krizhevsky et al., 2009), Caltech101 (Fei-Fei et al., 2006), Caltech256 (Griffin et al., 2007), ImageNet (Deng et al., 2009)), scene-level (SUN397 (Xiao et al., 2010)), fine-grained (Aircraft (Maji et al., 2013), Birdsnap (Berg et al., 2014), Cars (Krause et al., 2013), CUB (Wah et al., 2011), Flower (Nilsback & Zisserman, 2008), Food (Bossard et al., 2014), Pets (Parkhi et al., 2012)), textures (DTD (Cimpoi et al., 2014)), satelite images (EuroSAT (Helber et al., 2019)) and robustness (ImageNet-Sketch (Wang et al., 2019), ImageNet-R (Hendrycks et al., 2021)) for zero-shot image classification. For synthetic data amount, we generate 2000 synthetic images for each class in **B** and **LE**. For **LE**, we generate 200 sentences for each class name.

Main Results: 1) zero-shot classification results on 17 datasets; 2) study of synthetic data diversity; 3) study of synthetic data reliability; 4) study of model/classifier tuning; 5) study of the behavior of synthetic data for zero-shot classification in the training from scratch settings.

Synthetic data can significantly improve the performance of zero-shot learning. Our main studies in zero-shot settings are conducted with CLIP-RN50 (ResNet-50 (He et al. (2016)) as CLIP backbone), and we report results with our best strategy of **LE+CF+SCE**. As shown in Table 1, on 17 diverse downstream zero-shot image classification datasets, we achieve a remarkable average gain of 4.31% in terms of top-1 accuracy. Significantly, on the EuroSAT dataset, we achieve the largest perfor-

Dataset	CLIP	B		LE		LE+CF	
		CE	SCE	CE	SCE	CE	SCE
CIFAR-10	70.31	77.39 (+7.08)	78.23 (+7.92)	77.20 (+6.89)	77.55 (+7.24)	80.01 (+9.70)	**80.06 (+9.75)**
CIFAR-100	35.35	43.99 (+8.64)	44.25 (+8.90)	44.08 (+8.73)	44.91 (+9.56)	44.55 (+9.20)	**45.69 (+10.34)**
EuroSAT	37.51	45.64 (+8.13)	48.23 (+10.72)	53.26 (+15.75)	54.94 (+17.43)	54.75 (+17.24)	**55.37 (+17.86)**

Table 2: Ablation study on **Language Enhancement (LE), CLIP-based Filtering (CF), and Soft-target Cross-Entropy (SCE).**

mance boost of 17.86% in top-1 accuracy. We notice that the performance gain brought by synthetic data varies differently across datasets, which may be caused by the difference in the generation ability of the generative model for different label spaces.

Language diversity matters. By introducing more linguistic context into the text input, **LE** helps increase the diversity of synthetic data. As shown in Table 2, **LE** can achieve additional performance gains upon **B** in most cases (0.66↑ on CIFAR-100, 6.71↑ on EuroSAT), which demonstrates the efficacy of **LE** and the importance of synthetic data diversity for zero-shot classification.

Reliability matters. While **LE** could help increase the diversity of synthetic data, it also introduces the risks of noisy samples. Observed on CIFAR-10 in Table 2, **LE** sometimes even brings performance drops compared with **B** (0.68% ↓ on CIFAR-10), which may attribute to the noise introduced by enhanced language prompts, *e.g.* the sentence extended from the class name word may contain other class names or confusing objects. Fortunately, with **CF** to filter out unreliable samples, **LE+CF** yields consistent improvement upon **B**. Moreover, **SCE** generally achieves better performance than **CE**, showing its better adaptation to label noise.

Classifier tuning is enough for CLIP. Here, we investigate if only tuning the final classifier is the optimal solution in our setting with synthetic data. As shown in Table 3, we tune different proportions of the full model parameters on synthetic data for EuroSAT (0.02% corresponds to our default case where only the classifier is tuned), and report the zero-shot performance on the test set of EuroSAT. The best results are obtained by only tuning the classifier, and the performance gradually decreases as we gradually incorporate more parameters in the feature extractor for optimization, which agrees with the traditional strategy. The potential reason that tuning the feature extractor with synthetic data will not bring additional performance gains or might degrade model performance for zero-shot learning, is that pre-trained CLIP models already present good feature space which could be potentially corrupted due to the quality of synthetic data.

Param Tuned (%)	0	0.02	0.04	62.50	64.06	69.53	82.81	92.19
Acc	37.51	**55.37**	55.11	55.28	54.56	54.34	53.63	52.09

Table 3: **Parameters tuned v.s. Accuracy.** Dataset: EuroSAT.

Real shot	1	16	32	64	80	90	95	100
Acc	2.48	10.4	14.95	21.96	24.4	25.52	27.99	29.95

Table 4: **Setting when training from scratch.** Dataset: CIFAR-100.

Synthetic data deliver inferior performance in the training from scratch setting and are much less data-efficient than real data. To exclude the influence of powerful CLIP initialization in our study of synthetic data, we also conduct a from-scratch setting on the CIFAR-100 dataset, where we optimize a ResNet-50 model from random initialization. Given the label space of the CIFAR-100 dataset, we generate a synthetic dataset of 50k (500 images per class) to train a ResNet-50 model from scratch for image classification. We achieve a performance of 28.74% top-1 accuracy on CIFAR-100 test set, which is much lower than the performance of the pre-trained CLIP model (see Table 1). Further, we hope to investigate how many real in-domain training data can match the performance of our 50k synthetic data. As shown in Table 4, training with 95 images per category ($95 \times 100 = 9.5$k) will achieve comparable performance as that of 50k synthetic data. This manifests that synthetic data are not as efficient and effective as real data when solving downstream tasks. It requires around 5 times more data in order to achieve a comparable performance as that of real data.

Summary. Current synthetic data from text-to-image generation models could indeed bring significant performance boosts for a wide range of zero-shot image classification tasks, and is readily applicable with carefully designed strategies such as large-scale pre-trained models. Diversity and reliability matter for synthetic data when employed for zero-shot tasks. When the model is trained from scratch with synthetic data, synthetic data cannot deliver satisfactory performance and are much less data-efficient and effective for solving the classification task in comparison with real data.

3.2 Is Synthetic data ready for Few-shot Image Recognition?

In this section, we explore the effectiveness of synthetic data for few-shot tasks and how synthetic data impact the performance as more and more shots are included. Also, we design effective strategies to better leverage synthetic data.

Few-shot Image Recognition. We adopt the CLIP-based method as the model for few-shot image recognition due to its state-of-the-art performance (Radford et al., 2021). As discussed previously, various prompt learning based methods can be treated as tuning the classifier weights. We thus study how to tune the classifier weights with synthetic data. In an N-way M-shot case, we are given M real images of each test class, where $\mathrm{M} \in \{1, 2, 4, 8, 16\}$ in our experiments. With a total of $\mathrm{N} \times \mathrm{M}$ training samples, we hope to achieve favorable performance on a hold-out test set of the N classes.

Synthetic Data for Few-shot Image Recognition. While there have been a few attempts to study how to better adapt CLIP models for few-shot tasks (Zhou et al., 2022b;a; Zhang et al., 2022), they all focus on the model optimization level, and none have explored from the data level. Here, we systematically study whether and how synthetic data can be employed for solving few-shot image recognition tasks.

With the experience from synthetic data for zero-shot tasks, we adopt the best strategy (*i.e.* **LE+CF**) in the zero-shot setting as the basic strategy (**B**). Further, as the few-shot real samples can provide useful information on the data distribution of the classification task, we develop two new strategies leveraging the in-domain few-shot real data for better using synthetic data: 1) Real Filtering (**RF**): given synthetic data of one class c, we use the features of few-shot real samples to filter out synthetic images whose features are very close to the features of real samples that belong to other categories different from class c; 2) Real guidance (**RG**): we use the few-shot real samples as guidance to generate synthetic images where the few-shot real samples (added noise) replace the random noise at the beginning of the generation to guide the diffusion process (details in Appendix).

Experiment Setup. For datasets, we carefully select 8 image classification datasets from recent works (Zhou et al., 2022b;a; Zhang et al., 2022) that conduct few-shot learning upon CLIP: ImageNet (Deng et al., 2009), Caltech101 (Fei-Fei et al., 2006), Pets (Parkhi et al., 2012), Cars (Krause et al., 2013), Aircraft (Maji et al., 2013), SUN397 (Xiao et al., 2010), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019). For synthetic image number, we generate 800 images per class for **RG** method to approximately match the number of images in **B** and **RF**.

Main Results: 1) few-shot classification results on 8 datasets; 2) ablation study of training strategy; 3) ablation study of synthetic data generation strategy; 4) ablation study of BN strategy.

Synthetic data can boost few-shot learning and the positive impact of synthetic data will gradually diminish with the increase of real data shots. As shown in Figure 1 (results of more datasets are in the Appendix), with only few-shot real images for training, our implemented **CT w. init** (classifier weights initialized from CLIP text embeddings) performs comparably with the state-of-the-art CLIP tuning methods **Tip Adapter** (Zhang et al., 2022) and **CoOp** (Zhou et al., 2022b). **CT w. Syn** represents our results of applying synthetic data with mix training, real image as guidance, and freezing BN strategies. With the help of generated synthetic data, **CT w. Syn** achieves noticeable performance gains upon **CT w. init**, and achieves a new state-of-the-art few-shot learning performance across different datasets. We argue that for data-scarce few-shot classification, synthetic data could help address the insufficient data problem to boost performance. Besides, we notice that the boost from synthetic data gradually diminishes as the real shot number increases, which further validates that synthetic data work complementarily with real-world data for few-shot classification.

Mix Training fits few-shot learning with synthetic data. Now that we have two parts of data, *i.e.* few-shot real data and synthetic data, we could either 1) **phase-wise** train on each part of data with two training phases, or 2) adopt **mix training** that simultaneously utilizes two parts of data to update the model in each iteration. We provide the results in Table 7: we study on the EuroSAT dataset and use synthetic data generated from the **RG** method; under different shot number settings, mix training performs consistently better than two phase-wise strategies. We suggest that mix training

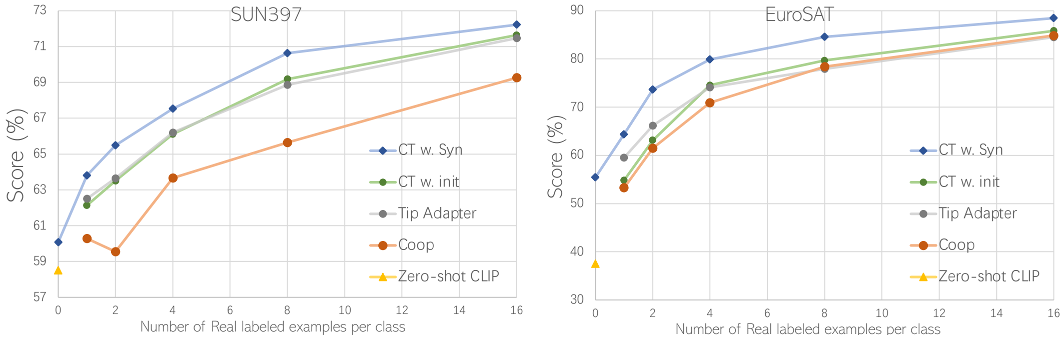


Figure 1: Results for few-shot image recognition. Results on all 8 datasets are provided in Appendix.

could help learn better classifiers since each part could function as a regularization for the other: synthetic data help alleviate instabilities brought by limited real samples, and real data help address the noise and domain gap of synthetic data.

Employing real data as guidance can alleviate domain differences and boost performance. We compare three strategies of synthetic data generation for few-shot tasks. As shown in Table 5, both **RF** and **RG** provide performance gains upon **B** which is the best strategy in the zero-shot setting. This demonstrates the importance of utilizing the domain knowledge from few-shot images for preparing the synthetic data. Further, **RG** significantly outperforms **RF**, yielding the best performance. This shows utilizing real data as guidance of the diffusion process help reduce the domain gap (visual illustrations in the Appendix).

B	RF	RG
87.1	87.33	88.47

Table 5: Ablation for Basic strategy (**B**), Real Filtering (**RF**), Real Guidance (**RG**) on EuroSAT, 16 shot.

Train data	Freeze BN?	Test Acc
Real		75.31
Real	✓	**85.63**
Syn		44.73
Syn	✓	**55.37**

Table 6: **Frozen BN works better** for 16-shot settings on EuroSAT.

M-shot	Phase-wise		Mix
	syn → real	real → syn	training
1	63.01	63.32	**64.36**
2	72.24	72.85	**73.62**
4	78.88	79.21	**79.88**
8	83.64	83.99	**84.57**
16	87.10	87.44	**88.47**

Table 7: **Mix training works better** for few-shot tasks on EuroSAT.

Frozen BN works better. Lastly, we investigate batch normalization (BN) strategies for our few-shot settings with synthetic data. As shown in Table 6, for both real and synthetic data, freezing the BN layers yields much better performance. We analyze that for real data, it is hard to get a good estimation of BN statistics when the number of images is limited. As for synthetic data, we attribute this to the statistical difference between different domains. Hence, we freeze BN layers during tuning for few-shot settings.

Summary. Synthetic data from text-to-image generation models could readily benefit few-shot learning and achieve a new state-of-the-art few-shot classification performance with strategies we present in this paper. However, the positive impact of synthetic data will diminish as more shots of real data are available which further confirms our previous claim that synthetic data are still not as effective as real data in training classification models.

3.3 Is Synthetic data ready for Pre-training?

Finally, we study whether synthetic data are effective in large-scale pre-training whose aim is to learn transferable representation. We also present effective strategies to better leverage synthetic data for model pre-training.

Pre-training for Transfer Learning. Recently, it has become a common practice to first pre-train models on large-scale datasets to obtain a well-trained feature extractor and then fine-tune the pre-trained models on a downstream task with labeled data (a.k.a. transfer learning). There have been various successful pre-training methods, including supervised pre-training (Joulin et al., 2016; Li

Syn data size	*w/o* ImageNet-1K pretrain	*w.* ImageNet-1K pretrain
/	78.83	84.50
1.2M	83.90 (+5.07)	84.90 (+0.40)
2.4M	85.03 (+6.20)	85.32 (+0.82)
3.6M	85.24 (+6.41)	85.52 (+1.02)

Table 8: **Downstream-aware synthetic pre-training.** Transfer results on CIFAR-100.

et al., 2017; Mahajan et al., 2018; Sun et al., 2017; Kolesnikov et al., 2020), self-supervised pre-training (Chen et al., 2020a; He et al., 2020; Caron et al., 2020; Grill et al., 2020; Chen & He, 2021; Zbontar et al., 2021; Ye et al., 2019), and semi-supervised pre-training (Xie et al., 2020; Pham et al., 2021).

Synthetic data for Pre-training. Since data amount and diversity play important roles in pre-training, we adopt the synthetic data generation strategy **LE** solely to maximize the scale of synthetic pre-training data. We study two settings for generating synthetic data for pre-training: 1) downstream-aware, where we have access to the label space of the downstream task, and thus we generate synthetic data according to the label space of the downstream task; 2) downstream-agnostic, where we have no access to downstream tasks in the pre-training stage, and we turn to a relatively general and diverse label space such as ImageNet-1K. For pre-training methods, we experiment with supervised pre-training and self-supervised pre-training methods.

Experiment Setup. We compare synthetic pre-trained models with models of random initialization and models of ImageNet-1K pre-training in terms of their transfer learning abilities. For downstream-aware settings: we conduct supervised pre-training on synthetic data generated according to CIFAR-100 label space and then transfer to CIFAR-100 through finetuning for evaluation.

For downstream-agnostic settings: we perform supervised pre-training and self-supervised pre-training (we adopt Moco v2 (Chen et al., 2020b) framework for its simplicity and reproducibility) on synthetic data generated from ImageNet-1K label space and evaluate the transfer performance by finetuning the pretrained models on a object detection dataset – PASCAL VOC (Everingham et al., 2010). Further, we experiment with ImageNet-2K label space (original ImageNet-1K and another non-overlapping 1K label names randomly selected from ImageNet-21K) to study the factors of data diversity and amount in synthetic pre-training. We use ResNet-50 as the backbone.

Results for Downstream-aware settings. We generate synthetic data of different sizes from CIFAR-100 label space, *i.e.* 1×, 2×, 3× ImageNet-1K data size, concretely 1.2M, 2.4M, 3.6M. We pre-train the model on the generated synthetic labeled set in a supervised manner, and then perform evaluation after finetuning the model on CIFAR-100. As shown in Table 8, with an equivalent amount of data as that of ImageNet-1k (1.2M), namely 1×, synthetic data for pre-training can largely reduce the gap between training from scratch (78.83%) and ImageNet- pre-trained model (84.50%). Moreover, with 2× and 3× synthetic data, pre-training on synthetic data outperforms ImageNet-1K pre-training with a noticeable margin. In addition, when we initialize the model from ImageNet-1K pre-trained weights when pre-training the model on synthetic dataset, we obtain extra boosts upon both results.

We conclude that for downstream-aware synthetic pre-training, synthetic data deliver close performance as that of ImageNet-1K pretraining with the same amount of data, synthetic data amount helps improve the results to outperforming ImageNet-1K pre-training, and synthetic pre-training could further benefit from ImageNet-1K pre-training.

Results for Downstream-agnostic settings. We first experiment with ImageNet-1K label space with 1× or 2× ImageNet-1K data size, *i.e.* 1.2M/2.4M IN-1K Syn. We perform supervised pre-training and self-supervised pre-training (*i.e.* Moco v2) on the generated synthetic data, and evaluate the pre-training results by transferring to the PASCAL VOC detection task. As it is too costly to validate all settings (*e.g.*, it takes more than 1 week to train Moco v2 on 4.0M synthetic data), we select several representative settings of interest to validate the effectiveness of synthetic data without hurting our conclusion.

As shown in Table 9 and 10, with 1.2M IN-1K Syn, both supervised pre-training (79.00%) and self-supervised pre-training (81.55%) could largely approach their IN-1K Real counterparts (super.:81.3%; self-super.:82.44%) and largely outperforms the result without pre-training (66.08%).

Data	pre-trained on IN-1k?	Syn. images amount			
		0	1.2M	2.4M	4.0M
(None)		66.08	-	-	-
IN-1K Syn		-	79.00	80.00	-
IN-2K Syn		-	-	80.54	80.72
(None)	✓	81.30	-	-	-
IN-1K Syn	✓	-	-	**81.78**	-
IN-2K Syn	✓	-	-	**81.87**	**81.91**

Table 9: Results for object detection on PASCAL VOC with **supervised pre-training**, all results are reported in AP_{50}.

Data	pre-trained on IN-1k?	Syn. images amount			
		0	1.2M	2.4M	4.0M
(None)		66.08	-	-	-
IN-1K Syn		-	81.55	82.13	-
IN-2K Syn		-	-	82.22	82.29
(None)	✓	82.44	-	-	-
IN-1K Syn	✓	-	-	**82.47**	-

Table 10: Results for object detection on PASCAL VOC when pre-trained with **Moco v2**, all results are reported in AP_{50}.

When increasing the data amount to 2.4M, the transferred results further increase, and the unsupervised pre-training method, *i.e.* Moco v2, performs better in utilizing our synthetic data thanks to its independence of labels, yielding a 82.13% transferred performance which surpasses supervised pre-training on IN-1K Real (81.30%) and is on par with its Moco v2 counterpart at IN-1K Real (82.44%). Next, we expand the label space by adding another 1K categories, producing IN-2K Syn. The enlarged diversity and data amount further bridge the gap between synthetic pre-training results and IN-1K Real pre-training results. Noticeably, the unsupervised pre-trained model Moco v2 (82.29%) largely approaches the IN-1K Real counterpart (82.44%) with negligible performance drop of 0.15%. Furthermore, when initialized from IN-1K Real pre-trained weights, both supervised and self-supervised pre-training improve upon both pure real data and synthetic data for pre-training.

Conclusion. In terms of transfer abilities, synthetic data from text-to-image generation models show surprisingly promising results for model pre-training, which is comparable to the standard ImageNet pre-training. We conclude our findings as follows:

1. Data amount has positive impacts on synthetic pre-training; performance could be improved by increasing synthetic data size, but the performance gradually saturates as the amount of data increases.

2. Synthetic data for pre-training is orthogonal to real data for pre-training.

3. For downstream-aware synthetic pre-training, we significantly outperform IN-1K Real (1.2M) pre-training with 2.4M/3.6M synthetic data on CIFAR-100.

4. For downstream-agnostic synthetic pre-training, we achieve comparable results with ImageNet (IN-1k) Real pre-training; self-supervised pre-training performs better than supervised pre-training. Besides, increasing the label space size could further improve the performance.

4 Conclusion

We systematically investigate whether synthetic data from current state-of-the-art text-to-image generation models are readily applicable for image recognition. Our extensive experiments demonstrate that synthetic data are beneficial for classifier learning in zero-shot and few-shot recognition, bringing significant performance boosts and yielding new state-of-the-art performance. Further, current synthetic data show strong potential for model pre-training, even surpassing the standard ImageNet pre-training. We also point out limitations and bottlenecks for applying synthetic data for image recognition, hoping to arouse more future research in this direction.

Limitations. In all investigated settings, we observe improved performance as the data amount and diversity (label space) increases. However, due to our limited computational resource, we are not able to further scale up data amount, which may take months to train one model. Besides, we are also not able to investigate larger model sizes and advanced architectures in the current investigation which is also worth exploring in the future. We present more discussions on limitations and future directions in the appendix.

REFERENCES

Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L. Alexander, David W. Jacobs, and Peter N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Victor Besnier, Himalaya Jain, Andrei Bursuc, Matthieu Cord, and Patrick Pérez. This dataset does not exist: training models from generated images. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2020.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv:2006.09882*, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020a.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

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

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. URL http://lmb.informatik.uni-freiburg.de/Publications/2015/DFIB15.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021.

Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv:2006.07733*, 2020.

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

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

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. *ICCV*, 2021.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23: 47–1, 2022.

Ali Jahanian, Xavier Puig, Yonglong Tian, and Phillip Isola. Generative models as a data source for multiview representation learning. *arXiv preprint arXiv:2106.05258*, 2021.

Armand Joulin, Laurens Van Der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *ECCV*, 2016.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, 2020.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Ang Li, Allan Jabri, Armand Joulin, and Laurens van der Maaten. Learning visual n-grams from web data. In *ICCV*, 2017.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. Commongen: A constrained text generation challenge for generative commonsense reasoning. *arXiv preprint arXiv:1911.03705*, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11461–11471, 2022.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.

S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2021.

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 preprint arXiv:2112.10741*, 2021.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729. IEEE, 2008.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *CVPR*, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NIPS*, 2015.

Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pp. 102–118. Springer, 2016.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–10, 2022a.

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 preprint arXiv:2205.11487*, 2022b.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022c.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pp. 10506–10518, 2019.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7959–7971, 2022.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020.

Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, 2019.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv:2103.03230*, 2021.

Renrui Zhang, Zhang Wei, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. *arXiv preprint arXiv:2207.09519*, 2022.

Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10145–10155, 2021.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16816–16825, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, pp. 1–12, 2022b.

A Appendix

In this appendix, we first provide more details of the diffusion models and text-to-image generation in Sec. A.1. Then, we illustrate how we achieve Real Guidance (**RG**) strategy in Sec. A.2. Next, we provide the full results of few-shot image classification on all 8 datasets in Sec. A.3. Besides, we show successful and failed examples of synthesized images from the language enhancement strategy in Sec. A.4. Moreover, in Sec. A.5, we elaborate on the implementation of the soft-target cross-entropy loss by offering an example code. Further, visualization of different strategies of synthetic data in few-shot settings is provided in Sec. A.6. Finally, we provide the implementation details in Sec. A.7.

A.1 Details of text-to-image diffusion model

In this section, we provide a brief review and derivation of the basic denoising diffusion probabilistic model (Ho et al. (2020)) alongside with the text-to-image engine in GLIDE (Nichol et al. (2021)).

A.1.1 Denoising diffusion probabilistic model

Denoising diffusion probabilistic model (DDPM) learns the data distribution through introducing a series of latent variables and matching the joint distribution. Formally, given a sample from the data distribution $x_0 \sim q(\mathbf{x}_0)$, a forward process $q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)=\prod_{t=1}^T q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)$ progressively perturbs the data with Gaussian kernels $q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right):=\mathcal{N}\left(\sqrt{1-\beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}\right)$, producing increasingly noisy latent variables $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$. Notably, x_t can be directly sampled from x_0 thanks to the closed form:

$$q\left(\mathbf{x}_t \mid \mathbf{x}_0\right)=\mathcal{N}\left(\mathbf{x}_t ; \sqrt{\bar{\alpha}_t} \mathbf{x}_0,\left(1-\bar{\alpha}_t\right) \mathbf{I}\right) \tag{1}$$

where $\alpha_t:=1-\beta_t$ and $\bar{\alpha}_t:=\prod_{s=1}^t \alpha_s$. In general, the forward process variances β_t are fixed and increased linearly from $\beta_1=10^{-4}$ to $\beta_T=0.02$. Besides, T should be large (*e.g.*, 1000) enough to ensure $q\left(\mathbf{x}_T \mid \mathbf{x}_0\right) \approx \mathcal{N}(0, \mathbf{I})$. Diffusion model aims to model the joint distribution $q\left(\mathbf{x}_{0:T}\right)$ which naturally involves a tractable sampling path for the marginal distribution $q\left(\mathbf{x}_0\right)$.

Specifically, the candidate distribution is formulated as a Markov chain with parameterized transition kernels:

$$p_\theta\left(\mathbf{x}_{0:T}\right):=p\left(\mathbf{x}_T\right) \prod_{t=1}^T p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right), p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right):=\mathcal{N}\left(\mathbf{x}_{t-1} ; \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)\right) \tag{2}$$

The training is thus achieved by optimizing a variational bound of negative log likelihood:

$$\mathrm{E}_{q(\mathbf{x}_0)}\left[-\log p_\theta\left(\mathbf{x}_0\right)\right] \leq \mathrm{E}_{q(\mathbf{x}_{0:T})}\left[-\log \frac{p_\theta\left(\mathbf{x}_{0:T}\right)}{q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)}\right]=: L \tag{3}$$

The loss term L can be rewritten as:

$$\mathbb{E}_q[\underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_T \mid \mathbf{x}_0\right) \| p\left(\mathbf{x}_T\right)\right)}_{L_T}+\sum_{t>1} \underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)\right)}_{L_{t-1}} \underbrace{-\log p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)}_{L_0}] \tag{4}$$

In practice, the core optimization terms are $L_{t-1}(t>1)$ that can be analytically calculated since both two terms compared in the KL divergence are Gaussians, *i.e.*,:

$$q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)=\mathcal{N}\left(\mathbf{x}_{t-1} ; \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right), \tilde{\beta}_t \mathbf{I}\right), p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right):=\mathcal{N}\left(\mathbf{x}_{t-1} ; \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)\right) \tag{5}$$

where $\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right):=\frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1-\bar{\alpha}_t} \mathbf{x}_0+\frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t} \mathbf{x}_t$ and $\tilde{\beta}_t:=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t$. Ho et al. (2020) fix $\boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)=\sigma_t^2 \mathbf{I}$ during training, where σ_t^2 is set to be β_t or $\tilde{\beta}_t$. Through reparameterization trick (Kingma & Welling (2013)) and empirical simplification (Ho et al. (2020)), the final training term is performed as follows:

$$L_{\text{simple}}(\theta):=\mathbb{E}_{t, \mathbf{x}_0, \epsilon}\left[\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t} \mathbf{x}_0+\sqrt{1-\bar{\alpha}_t} \boldsymbol{\epsilon}, t\right)\right\|^2\right] \tag{6}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and t is uniformly sampled between 1 and T.

After training, started from an initial noise map $x_T \sim p(\mathbf{x}_T) = \mathcal{N}(0, \mathbf{I})$, new images can be then generated via iteratively sampling from $p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$ using the following equation:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)\right) + \sigma_t \mathbf{z}, \text{ where } \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{7}$$

A.1.2 Text-to-image generation

The text-to-image diffusion model extends the basic unconditional diffusion model by changing the target distribution $q(\mathbf{x_0})$ into a conditional one $q(\mathbf{x_0} \mid \mathbf{c})$, where $\mathbf{c}$ is a natural language description. The derivation of the training terms and sampling procedure are similar to Sec. A.1.1, except that a conditioning signal $\mathbf{c}$ is included. Besides, following the improved DDPM (Nichol & Dhariwal (2021)), $\mathbf{\Sigma}_\theta$ is also estimated in GLIDE (Nichol et al. (2021)).

Especially, GLIDE employs a coarse-to-fine two-stage generation framework (Nichol & Dhariwal (2021); Saharia et al. (2022c)) with two guidance techniques for balancing mode coverage and sample fidelity, namely classifier guidance (Dhariwal & Nichol (2021)) and classifier-free guidance (Ho & Salimans (2022)). Classifier guidance mainly relies on an extra trained noise CLIP model to provide feedback at intermediate sampling steps. Classifier-free guidance, on the other hand, randomly drops the text prompt with a fixed probability p during the training, which can be viewed as a joint training of an unconditional model $\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t \mid \emptyset\right)$ (*i.e.*, $\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t\right)$) and a conditional model $\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t \mid \mathbf{c}\right)$. At each sampling step, the model's output is actually performed using an extrapolation as follows:

$$\hat{\boldsymbol{\epsilon}}_\theta\left(\mathbf{x}_t \mid \mathbf{c}\right) = \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t \mid \emptyset\right) + s \cdot \left(\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t \mid \mathbf{c}\right) - \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t \mid \emptyset\right)\right) \tag{8}$$

where s is a guidance scale that can trade off sampling quality and diversity. In our work, we use classifier-free guidance with default setting $s = 3$ for all experiments since it achieves better results than CLIP guidance. To speed up the sampling process, DDIM (Song et al. (2020)) is utilized which allows the model to produce high-quality images within few seconds. We follow the default settings in GLIDE and set $T = 100$ in the coarse stage and $T = 27$ in the upsampler stage.

A.2 Details of real guidance (RG) strategy

We elaborate how we use few-shot in-domain real images to guide the generation process for few-shot settings. In a normal text-to-image generation process, a pure noisy image $x_T \sim \mathcal{N}(0, \mathbf{I})$ would be sampled first as the initialization of the reverse path. Then, the pretrained GLIDE model iteratively predicts a less noisy image x_{t-1} $(t = T, T-1, ..., 1)$ using the given text prompt c and the noisy latent image x_t as inputs. In our case, we add noise to a reference image x_0^{ref} such that the noise level corresponds to a certain time-step $t_\star$:

$$x_{t_\star}^{ref} = \sqrt{\bar{\alpha}_{t_\star}} x_0^{ref} + \sqrt{1-\bar{\alpha}_{t_\star}}\epsilon \tag{9}$$

Then, rather than sampling from time-step T, we initialize the noisy latent variable as $x_{t_\star}^{ref}$ and begin our denoising process from time-step $t_\star$, as illustrated in Algorithm 1. Note that the GLIDE model adopts a coarse-to-fine two-stage generation framework and involves classifier-free guidance. However, we omit them in Algorithm 1 for simplicity since our image-guidance strategy only modifies the start point and leaves the other settings unchanged. In this way, the generated images can share similar in-domain properties, and thus helping to close the domain gap. While small $t_\star$ could synthesis images which are more similar to the reference image, it results in low diversity, which harms the classifier's learning. In the case of a large $t_\star$, $x_{t_\star}^{ref}$ retains too little information from x_0^{ref}, causing the generated image to deviate from the domain. In our experiments, we conduct different trade-offs considering different few-shot settings. Empirically, we set $t_\star$ as 15, 20, 35, 40, and 50 for shot 16, 8, 4, 2, and 1, respectively.

A.3 Main results on Few-shot tasks

Due to the limit of pages and space, we provide the results for few-shot tasks of all 8 datasets in Figure A.2. We observe similar results on various datasets that synthetic data could boost few-shot learning performance and achieve a new state-of-the-art performance.

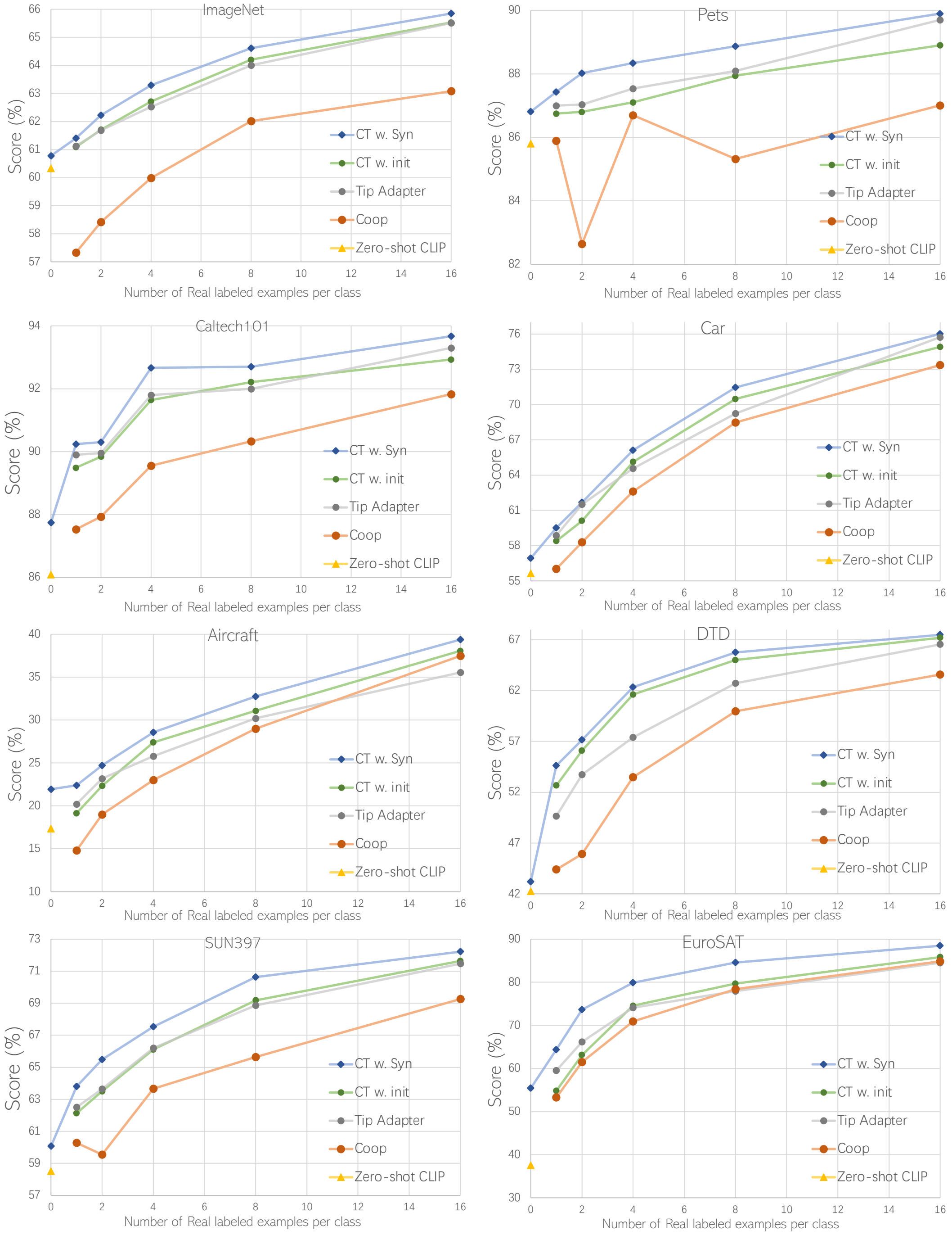
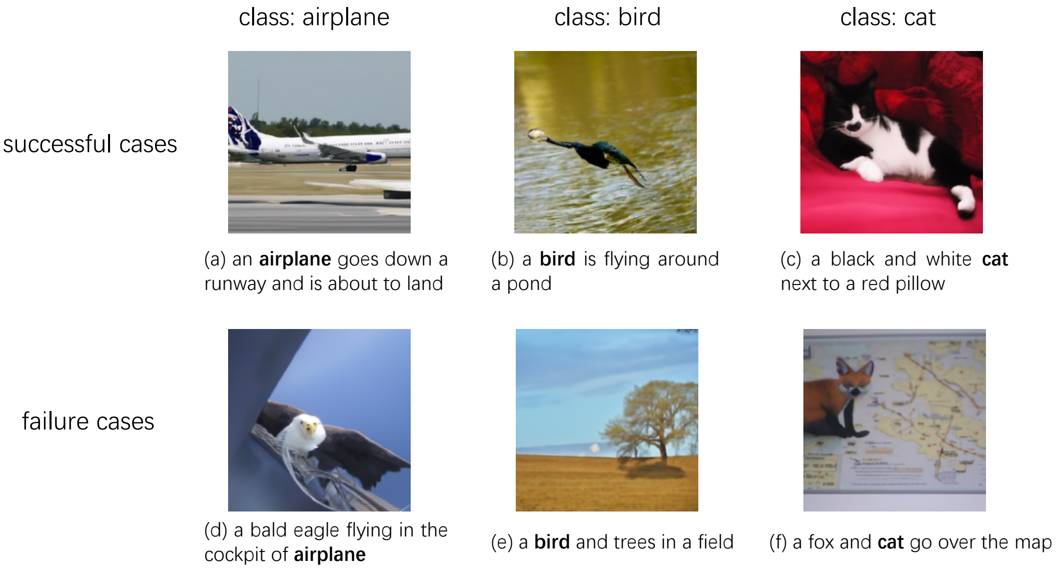


Figure A.2: Main results for Few-shot tasks on all 8 datasets.

Algorithm 1 Real Guidance (**RG**) Strategy

```
Input: Reference image $x_0^{ref}$, text prompt $c$ and GLIDE model $(\mu_\theta, \Sigma_\theta)$.
Output: Generated image $x_0$
# Noisy variable initialization
Select a time-step $t_\star \sim 1, 2, 3, ..., T$ and random noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$
Obtain initial noisy image $x_{t_\star} := x_{t_\star}^{ref}$ according to Eq. 9
# Random Sampling (could be replaced by DDIM for speed-up)
for $s$ from $t_\star$ to 1 do
    $\mu, \Sigma \leftarrow \mu_\theta\left(x_s, s, c\right), \Sigma_\theta\left(x_s, s, c\right)$
    $x_{s-1} \leftarrow$ sample from $\mathcal{N}\left(\mu, \Sigma\right)$
end for
return $x_0$
```



(a) an **airplane** goes down a runway and is about to land

(b) a **bird** is flying around a pond

(c) a black and white **cat** next to a red pillow

(d) a bald eagle flying in the cockpit of **airplane**

(e) a **bird** and trees in a field

(f) a fox and **cat** go over the map

Figure A.3: Examples of synthesized images from Language Enhancement strategy.

A.4 Examples of synthesized images from Language Enhancement strategy

Here, we provide successful and failed examples of synthesized images from the language enhancement strategy. As shown in Figure A.3 (a) $\sim$ (c), language enhancement could introduce more diversity into the language prompts and lead to more diversified synthesized images for each class, such as introducing "runway" in (a), "pond" in (b), and "red pillow" in (c). However, we also observe failure cases after the language enhancement process. As we can see in Figure A.3 (d) $\sim$ (f), after introducing some other items into the language prompts, the focus of the generated images may move to other objects rather than the target class. In some extreme failure cases we show here, the generated images may even not contain the desired class object.

A.5 Soft-target cross-entropy loss

Example code for soft-target cross-entropy loss is shown below.

```
def soft_target_cross_entropy(logits, target, labels, T=2):
    # T: temperature for soft targets.
    loss_func_CE = torch.nn.CrossEntropyLoss()
    CE = loss_func_CE(logits, labels)
    soft_targets = torch.softmax(target/T, dim=1)
    SCE = torch.sum(-soft_labels * F.log_softmax(x, dim=-1), dim=-1)
    loss = 0.5 * CE + 0.5 * SCE
    return loss
```

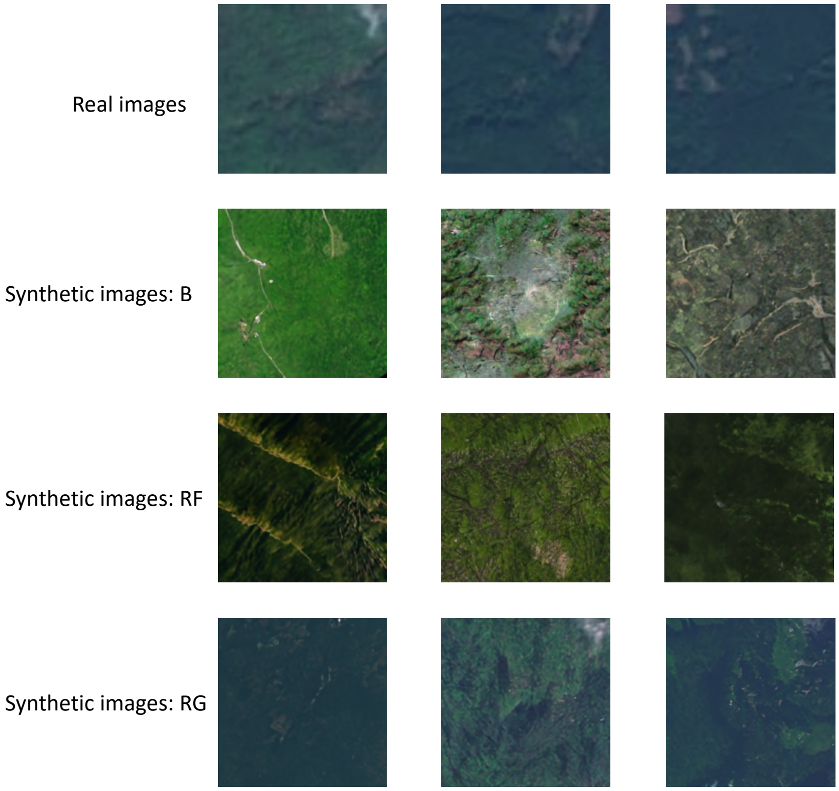


Figure A.4: Visualization of different strategies of synthetic data in few-shot settings.

A.6 Visualization of different strategies of synthetic data in few-shot settings

We provide the visual illustration of synthesized images by different strategies, *i.e.* basic (**B**), real filtering (**RF**), and real guidance (**RG**) as well as real images of the same class for comparison. Here, we take the "forest" class in EuroSAT dataset as an example. As shown in Figure A.4, both **RF** and **RG** strategies produce images with reduced domain gaps from the real images of the target domain. Further, **RG** significantly approaches the real images better than **RF**, demonstrating the effectiveness of the proposed **RG** method.

A.7 Implementation details

A.7.1 Zero-shot setting

For text-to-image generation process, we adopt the default hyperparameters from the official GLIDE text-to-image code. The input text of the basic strategy is "a photo of a [CLASS]", and the input text of language enhancement strategy is "a photo of a [SENTENCE]". For language enhancement, we adopt an off-the-shelf word-to-sentence T5 model pre-trained on "Colossal Clean Crawled Corpus" dataset (Raffel et al., 2020) and finetuned on CommonGen dataset (Lin et al., 2019). We generate 2000 synthetic images for each class in **B** and **LE**, and use a threshold of 1/N in **CF** where N is the number of classes. For **LE**, we generate 200 sentences for each class name.

For training on synthetic data for zero-shot recognition, we use AdamW (Loshchilov & Hutter, 2017) optimizer and an initial learning rate of 0.002 that is decayed by the cosine annealing rule. We train for 30 epochs, and use weight decay of 0.1 and batch size of 512. For image preprocessing, we resize the image's short side to 224 while keeping the original aspect ratio.

A.7.2 Few-shot setting

For the text-to-image generation process in few-shot settings, our basic strategy and Real filtering strategy both apply the same process as in the zero-shot settings; and for our Real guidance strategy, the generation process is illustrated in Sec. A.2. For synthetic image number, we generate 800 images per class for **RG** method to approximately match the number of images in **B** and **RF**.

For training in the few-shot settings, we again use AdamW optimizer, weight decay of 0.1 and the cosine annealing rule. We use batch size of 32 for few-shot real images and 512 for synthetic images. For phase-wise training, we train for 30 epochs for each stage and use an initial learning rate of 0.002. For mix training, we train for 30 epochs and use an initial learning rate of 0.001, where the loss values from real data and synthetic data are added at a 1:1 ratio in each iteration. For image preprocessing, we adopt the same strategy as zero-shot settings.

A.7.3 Pre-training setting

For pre-training settings, we adopt the **LE** strategy in zero-shot settings for generating massive amounts of diversified synthetic pre-training data. For downstream-aware settings, we generate synthetic data by language prompts constructed from CIFAR-100 label space through the word-to-sentence model, and we generate synthetic data of 1.2M, 2.4M, 3.6M. For downstream-agnostic settings, we generate from a generic label space of ImageNet-1K or ImageNet-2K, and data amount of 1.2M, 2.4M, 4M.

For training details of downstream-aware synthetic pre-training on CIFAR-100 label space, we use AdamW optimizer, weight decay of 0.9, batch size of 512, training epochs of 90, and the cosine annealing rule for adjusting learning rate. For the initial learning rate, we use $1e$-4 when pre-training from random initialization, and $1e$-5 when pre-training from ImageNet pre-trained weights. For data augmentation, we adopt random cropping, resizing, and random horizontal flip. For transfering on CIFAR-100 dataset, we train for 200 epochs and use a SGD optimizer with an initial learning rate of 0.003, which is multiplied by 0.2 at 60, 120, and 160 epochs. We use batch size of 128 and weight decay of $5e$-4.

For downstream-agnostic synthetic supervised pre-training, we train for 90 epochs and use a SGD optimizer with an initial learning rate of 0.2 for training from random initialization and 0.001 for training from ImageNet pre-trained weights. We multiply the initial learning rate by 0.1 every 30 epochs. We use batch size of 512 and weight decay of $1e$-4. For data augmentation, we also adopt random cropping, resizing, and random horizontal flip.

For downstream-agnostic synthetic self-supervised pre-training, *i.e.* Moco v2, we follow the hyper-parameters of the original implementation of the paper when training from random initialization. When initialized from ImageNet pre-trained weights, we use a small initial learning rate of 0.003 and keep other hyperparameters the same.

For transfer evaluation of object detection on PASCAL VOC 2012, we use the Faster R-CNN (Ren et al., 2015) detector and the backbones are initialized by the pre-trained weights. All the setups follow the evaluation protocols in Moco (He et al., 2020).