Hierarchical Prompting for Diffusion Classifiers

Wenxin Ning[1], Dongliang Chang[2⋆], Yujun Tong[1], Zhongjiang He[3], Kongming Liang[1], and Zhanyu Ma[1]

[1] School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China
[2] Department of Automation, Tsinghua University, Beijing, China
[3] China Telecom Artificial Intelligence Technology Co. Ltd, Beijing, China
[1]{wenxinning, tongyujun, liangkongming, mazhanyu}@bupt.edu.cn
[2]changdongliang@pris-cv.cn [3]hezhongj_1@163.com

Abstract. Recently, large-scale pre-trained text-to-image models like Stable Diffusion have demonstrated unparalleled capabilities, revolutionizing many tasks. Recent studies have found that these advanced generative models can be applied to discriminative tasks, showing strong accuracy and robustness in zero-shot recognition. However, the current pipeline suffers from impractical inference speed (about 1 minute per image). In this paper, we introduce Hierarchical Prompt Learning, a simple and effective pipeline to achieve high-speed classification for diffusion generators. Our method first proposes a hierarchical evaluation strategy, leveraging prior class tree taxonomy to reduce unnecessary class modeling. To handle the excessive sampling steps, we employ prompt learning, a parameter-efficient technique, to adapt downstream task-specific knowledge into the conditional text embedding. This allows our method to efficiently sample diffusion models in just **25 steps** while maintaining high accuracy. The proposed hierarchical evaluation achieves up to **3.5x speedups** compared to previous diffusion classifiers, and the combination with prompt learning achieves up to **20x speedups**. Beyond efficiency, our method also maintains high performance in zero-shot and few-shot scenarios, both in-distribution and out-of-distribution. Moreover, our visualization analysis sheds light on what our diffusion prompts learn, providing insights into the model's decision-making process. Codes are available at https://github.com/PRIS-CV/Hierarchical-Prompting-for-Diffusion-Classifiers.

Keywords: Diffusion classifier · Zero/Few-shot classification · Prompt learning.

1 Introduction

Currently, the main paradigm for classification tasks relies on discriminative models. These models have achieved high-speed inference and human-level accuracy [14, 21]. However, they often focus on the most discriminative patterns

⋆ indicates corresponding author.

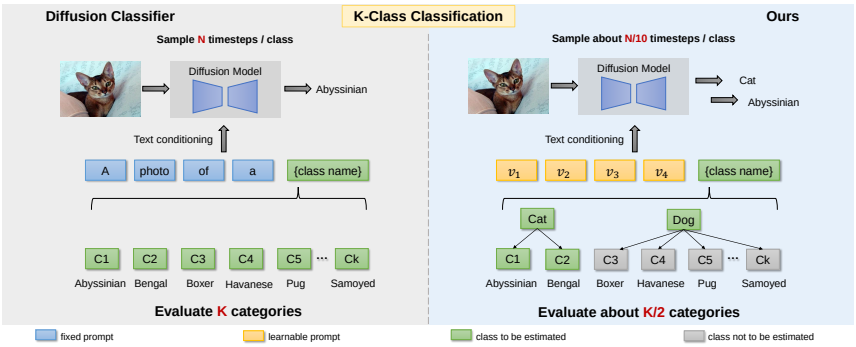

Fig. 1. Comparison of our method (right) with diffusion classifier (left). Take Pets dataset [36] as an example, for Diffusion Classifier, N timesteps are sampled for each class (total K classes) to compute its prediction score. For our method, hierarchical evaluation reduces the number of classes that need to be estimated (green cells), and prompt learning reduces the number of sampling timesteps for each class. They simultaneously increase the speed of inference.

(e.g., the furry sharp ears of a cat) to achieve such performance, which comes at the expense of generalization ability. As a result, they tend to perform poorly on out-of-distribution data and are prone to shortcut learning (e.g., associating green grass with cows) [18, 29].

Conversely, generative models offer a more comprehensive data distribution modeling [19]. Recent advancements in diffusion models have excelled in text-to-image generation, image-to-image translation and other challenging tasks [12, 15, 23, 24, 47]. Models like Imagen [41], DALL-E 2 [39], and Stable Diffusion [40] generate realistic, high-resolution images from diverse text prompts. This suggests a potentially more detailed and comprehensive understanding of objects, leading to robust recognition abilities. Leveraging this, some pioneers have adapted large-scale pre-trained generative models for downstream discriminative tasks [11, 32], achieving accurate zero-shot classification. For instance, Li et al. [32] model each class's log-likelihosod (exactly the variational lower bound) to score the image. These methods iteratively noise and denoise test inputs using each class as a text prompt to condition the model, and select the class with the best denoising ability. This paradigm exhibits excellent out-of-distribution performance and robust recognition ability. However, the slow inference speed (about 1 minute per image) significantly hinders its practical application.

The slow and costly inference can be attributed to two main reasons. First, unlike traditional discriminative models that compute all class probabilities in a single forward pass, diffusion classifier models individually model the conditional likelihood of each class, leading to inference time proportional to the number of classes. Second, approximating each class probability needs an excessive number of sampling steps. Inherent in the diffusion iterative inference process, generative classifiers also suffer from this stochastic denoising process over hundreds of timesteps, which would otherwise result in lower accuracy. To address these

two problems, we introduce Hierarchical Prompt Learning in this paper, as illustrated in Fig. 1. We first propose a hierarchical evaluation strategy. The key insight is to drop apparently "wrong" categories to avoid evaluating all classes. However, effectively pruning "wrong" categories is crucial yet non-trivial. To this end, we introduce prior information to generate category hierarchical trees based on category taxonomy. During inference, our evaluation is divided into multi-granularities based on the hierarchical tree, from coarse to fine. For example, in Fig. 1, once recognizing "cat" at the parent hierarchy level, there is no need to evaluate classes belonging to "dog". This hierarchical prediction architecture saves inference time by pruning distant sub-categories, leaving more computational space for complex fine-grained categories.

Another key contribution is that we find embedding task-specific category knowledge through prompting can greatly alleviate the requirement for massive sampling steps to achieve high accuracy. For many large discriminative multi-modal models, such as CLIP [38], BLIP [33], and ChatGPT [5], the text input, also known as a prompt, has proven to have a significant influence on downstream performance. Consequently, prompt learning [4, 17, 35, 50] has emerged to optimize downstream performance through end-to-end training with few downstream samples. Specifically, it replaces prompts such as "`a photo of a {CLASS}.`" with continuous learnable word embeddings (Fig. 1, right). Inspired by this, we combine this technique with diffusion classifiers, reducing the number of sampling timesteps for each class by more than 10 times compared to previous baselines while achieving slightly better performance. Moreover, prompt learning, as a parameter-efficient fine-tuning technique, introduces negligible training costs, requiring only a few extra downstream samples.

Comprehensive experiments are conducted to validate the effectiveness of our method. We select fine-grained visual classification tasks as our testbeds due to their natural hierarchical taxonomy. Our approach can also be adapted to other downstream classification tasks by developing hierarchical trees based on task-specific granularity. Compared to previous state-of-the-art methods, our approach achieves 7-20x speedups and an average accuracy improvement of 12.28% across all datasets. Furthermore, our method demonstrates significantly stronger robustness than the baseline against domain shifts. Additionally, through visualization of the generation process, we analyze the model's decision-making process and explain how our diffusion prompts enhance discriminative ability.

2 Related Work

Generative models for discriminative tasks In recent years, many works have explored how generative models can be applied to discriminative tasks. Some methods use generative models for data augmentation and then train discriminative tasks on augmented training datasets [1, 26, 44, 49]. Some methods extract features from generative models for direct use or fine-tuning in discriminative tasks [2, 13, 20, 37, 43, 48]. Xu et al. [48] utilize a pre-trained text diffusion model to extract features and combine it with a discriminative model for open

vocabulary panoptic segmentation. The above methods use the generative model as an auxiliary task to the discriminative model. Another category of inversion-based methods use generative models directly for the discriminative task [11,32]. Li et al. [32] show that zero-shot classification can be achieved by utilizing density estimates derived from large-scale text-to-image diffusion models such as Stable Diffusion [40]. These types of methods have stronger multimodal compositional reasoning ability and robustness [32]. However, the inference speed of such methods is very slow and there is still a gap with the accuracy of discriminative methods on the zero-shot classification task. Our work is based on the inversion-based methods with improvements in inference speed and performance on downstream classification tasks.

Parameter-Efficient Fine-tuning Parameter-Efficient Fine-tuning (PEFT) focuses on selectively updating a limited number of parameters in a large pretrained model for downstream tasks. They add small feedforward networks between layers in the pretrained model [3, 25], or apply techniques to select and fine-tune a sparse set of model parameters on labeled training datasets [27, 28]. The goal is to keep performance while updating as few parameters as possible.

Prompt learning is one type of PEFT that involves adding instructions to inputs and pre-training the language model to enhance downstream tasks. Manually defined prompts are commonly used to provide guidance to large pre-trained models like CLIP [38], GPT-3 [5], etc. However, it is impractical and sub-optimal to find the best manually defined prompt. To solve these problems, several methods have been proposed to automatically optimize continuous vectors in the word embedding space for large-scale vision-language models like CLIP. Zhou et al. [50] replace hand-crafted prompts with a set of learnable prompts inferred from labeled few-shot samples, namely CoOp. Nayak et al. [35] learn attribute and object soft prompts for compositional zero-shot learning. In terms of generation, Gal et al. [16] propose textual inversion to generate images with new objects or styles by learning new words in the textual embedding space of pre-trained text-to-image models. Despite its success in discriminative models, prompt learning has not been applied to generative models (similarly technique merely applied to the customized generation [16]).To best of our knowledge, we are the first to combine the diffusion classifier and prompting for the classification tasks.

Fine-grained visual classification Fine-grained visual classification (FGVC) aims to achieve a refined classification of subclasses within a large class, where instances have small inter-class variations and large intra-class variations [6]. Many works [7–10, 42] use hierarchical labelling framework of FGVC due to its hierarchical nature with different levels of concept abstraction. Chang et al. [7] use specific classification heads for different levels and limit gradient flow to update only the parameters in each head, enabling distinction between coarse-grained and fine-grained features. Chen et al. [9] employ a hierarchical tree to establish relationships between parent and child labels, ensuring mutual exclusivity of classes at the same level. Our approach also uses multi-granularity labels to con-

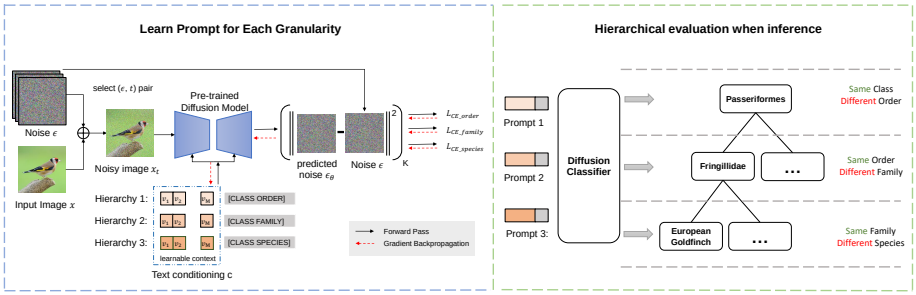

Fig. 2. Overview of our Hierarchical Prompt Learning method. For training, we input images into the diffusion model and use each class's learnable prompt as a control condition to get similarity scores. Next, we compute the cross-entropy loss from similarity scores. Finally, learnable prompts are updated by backpropagating the loss. For inference, we use the hierarchical evaluation strategy. At each level, condition the diffusion model with learned prompt and select the class with the highest similarity score at this level. According to the hierarchical tree, go to the next level whose leaf nodes continue the previous process.

struct a hierarchical tree. Instead of exploring the relationship between features of different levels, we use the hierarchical information for inference acceleration.

3 Method

In this section, we introduce Hierarchical Prompt Learning, as inllustrated in Fig. 2, which is combined with hierarchical evaluation (Sec. 3.2) and prompt learning (Sec. 3.3).

3.1 Prerequisites

Diffusion Generative Classifier In this section, we describe how current methods [11,32] convert a conditional diffusion model into a zero-shot classifier. Let's denote a dataset $\{(\mathbf{x}^1, \mathrm{y}^1), \ldots, (\mathbf{x}^M, \mathrm{y}^M)\}$ with M labeled samples, where each image belongs to one of K classes. We can then derive conditional prompts $[\mathbf{c}_k] := \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_K\}$ from the labels y_k (e.g., "A photo of [class]").

Given a test input $\mathbf{x}$, the goal is to use a diffusion generator parameterized by θ to predict the most likely class. According to Bayes' theorem, we can turn the likelihood $p_\theta(\mathbf{c}_k \mid \mathbf{x})$ estimated by generator G into predictive probabilities for the classification task :

$$p_\theta(\mathbf{c}_k \mid \mathbf{x}) = \frac{p_\theta(\mathbf{x} \mid \mathbf{c}_k) p(\mathbf{c}_k)}{p(\mathbf{x})}. \tag{1}$$

Since $p(\mathbf{x})$ is the same for all classes, it can be ignored for the purpose of classification, and we can focus on computing $p_\theta(\mathbf{x} \mid \mathbf{c}_k) p(\mathbf{c}_k)$. Assuming a uniform prior $p(\mathbf{c}_k)$, the classification problem then simplifies to finding the class k that maximizes $p_\theta(\mathbf{x} \mid \mathbf{c}_k)$:

$$\tilde{y} = \arg\max_{k} p_\theta(\mathbf{x} \mid \mathbf{c}_k). \tag{2}$$

Since log-likelihoods cannot be obtained directly from the diffusion model, existing methods use approximate evidence lower bound (ELBO) as an alternative [11, 32]. In the diffusion model, the approximate ELBO can be expressed as [23]:

$$p_\theta(\mathbf{x} \mid \mathbf{c}_k) \geq -\mathbb{E}_{t,\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c})\|^2\right] + C, \tag{3}$$

where x_t is the noised sample by adding Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ to the clean input x with timestep $t \sim [1, 1000]$, and $\epsilon_\theta(\mathbf{x}_t, \mathbf{c})$ is the predicted noise through the network and C is a constant [23].

By plugging Eq. 3 into Eq. 2, the prediction of the model is:

$$\begin{aligned} \tilde{y} \leftarrow \tilde{\mathbf{c}} &= \arg\max_{\mathbf{c}_k} \log p_\theta(\mathbf{x} \mid \mathbf{c}_k) \\ &\approx \arg\min_{\mathbf{c}_k} \mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_k)\|^2]. \end{aligned} \tag{4}$$

In practice, each expectation in Eq. 4 is computed using an unbiased Monte Carlo estimate by sampling N pairs (t_i, ϵ_i), with $t_i \sim [1, 1000]$ and $\epsilon_i \sim \mathcal{N}(0, I)$, and computing:

$$S_{\mathbf{c}_k}(\mathbf{x}) = -\mathbb{E}_{t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{c}_k)\|^2] \approx -\frac{1}{N}\sum_{i=1}^{N} \|\epsilon_i - \epsilon_\theta(\mathbf{x}_{t_i}, \mathbf{c}_k)\|^2 . \tag{5}$$

We denote it as prediction similarity score $S_{\mathbf{c}_k}$ for each class k. It can be simply summarized as the mean squared error between the prediction noises of the network and the original sampling noises added to the image, with higher scores representing higher similarity.

Efficiency Improvement Strategy If using diffusion model for classification according to the above process directly, $K \times N$ trails are required for each image, where K is the number of classes and N is the number of sampling pairs in the set $\{(t_i, \epsilon_i)\}_{i=1}^{N}$ to estimate the prediction scores. For accurate prediction, hundreds of pairs (e.g., 250-500) are usually required, resulting in very slow inference.

To reduce inference time, Diffusion Classifier [32] uses a adaptive evaluation strategy. The strategy first roughly classifies all the categories and then finely classifies the most likely part of the categories. Specifically, evaluation is split into a series of stages, in each of which the classes with the highest prediction score are selected to the next stage. Taking two stages for Pets dataset as an example, in the first stage, a smaller amount of N_1 (25 times) is used to evaluate each class, and then the K^{top} (top 5) classes with the highest prediction scores are selected to the second stage with a larger amount of N_2 (250 times). This allows more computational resources to be allocated to the more reasonable classes. However, in case of large number of classes, the inference is still slow. For example, it takes 61 seconds to classify an image of CUB-200-2011 dataset with 200 classes. Our proposed strategy in Sec. 3.2 further reduces the inference time.

3.2 Hierarchical evaluation when inference

Despite the adaptive evaluation strategy used in Diffusion Classifier [32], the inference time still increases roughly linearly with the number of categories. Adaptive evaluation requires to evaluate all categories in the first stage, which is time-consuming and unnecessary. To further reduce the number of categories to be estimated, our approach utilizes a smarter hierarchical evaluation strategy using multi-granularity labels. We construct hierarchical classification tree based on multi-granularity labels. When inference, first classify the coarse-grained level and then the fine-grained level according to the structure of the hierarchical tree. In this way, the number of categories to be classified is reduced and the scope of classification is refined.

Take two levels in CUB [45] dataset as an example, "order" (coarse) and "species" (fine) granularity. When inference, the conditional prompts $[\mathbf{c}_{order}]$ are first synthesized based on the order level labels $[\mathrm{y}_{order}]$ of K^{order} categories. Then the prediction class at the order level can be obtained by finding highest prediction score $S_{\mathbf{c}_k}(\mathbf{x})$ (i.e. the Monte-Carlo estimate for condition $\mathbf{c}_k$ on image $\mathbf{x}$ in Eq. 5):

$$\tilde{y}_{order} \leftarrow \tilde{\mathbf{c}}_{order} = \underset{\mathbf{c}_k \in [\mathbf{c}_{order}]}{\operatorname{argmax}} S_{\mathbf{c}_k}(\mathbf{x}), \tag{6}$$

where $[\mathbf{c}_{order}] := \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{K^{order}}\}$.

Next, find the sub-classes $[\mathrm{y}_{species}]$ of $K^{species}$ categories belonging to prediction $\tilde{y}_{order}$ in the hierarchical tree and transform them to prompts $[\mathbf{c}_{species}]$. Similarly, we can get the prediction class at the species level:

$$\tilde{y}_{species} \leftarrow \tilde{\mathbf{c}}_{species} = \underset{\mathbf{c}_j \in [\mathbf{c}_{species}]}{\operatorname{argmax}} S_{\mathbf{c}_j}(\mathbf{x}), \tag{7}$$

where $[\mathbf{c}_{species}] := \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_{K^{species}}\}$.

Discussion By hierarchical classification during inference, $(K^{order} + K^{species})$ categories for which the conditional likelihood are required to be computed when classifying a given image. As for adaptive evaluation strategy, $(K + K^{top})$ categories are required. Clearly, due to the introduction of the hierarchical prior, $(K^{order} + K^{species})$ are much smaller than $(K + K^{top})$. When keeping the number of sampling steps the same for each category, the number of trials our method required is fewer than adaptive evaluation strategy, which improves the speed of inference. Here we just take two levels as an example. For the same hierarchical structure, when the number of levels is higher, theoretically more categories can be eliminated. We also conduct experiments to illustrate this phenomenon in Sec. 4.3. However, although the number of categories to be predicted is reduced, the large number of sampling steps per category still leads to a long sampling time. In Sec. 3.3 we will further discuss how our method reduces the large number of sampling steps.

3.3 Learn prompt for each granularity

To further reduce the number of sampling steps per category, we introduce prompt learning, which is a parameter-efficient learning technique for fine-tuning pre-trained diffusion models. We replace hand-crafted prompts with learnable context, bringing task-relevant knowledge into the control prompts. With the inclusion of learned prompts, compared to previous diffusion classifiers, the number of sampling pairs in the set $\{(t_i, \epsilon_i)\}_{i=1}^{N}$ can be reduced to $1/10$ or even less while increasing the accuracy. Specifically, the conditional prompt for class k given to the diffusion model is designed as:

$$\mathbf{c}_k = \{v_1^k, v_2^k, \ldots, v_M^k, y_k\}, \tag{8}$$

where each $v_m^k \in \mathbb{R}^d (m \in \{1, \ldots, M\})$ is a learnable vector with the same dimension as word embeddings, M is a hyperparameter specifying the number of context tokens to be tuned[1], and $y_k (k \in \{1, \ldots, K\})$ is the label. Since diffusion classifiers require comparing differences across classes, to better capture class-specific information, we choose to design specific context vectors for each class. In this way, v_m^k is not the same when k changes.

Next, input the conditional prompt $\mathbf{c}_k$ into the Eq. 5 and get prediction score $S_{\mathbf{c}_k}$ for each class y_k on image $\mathbf{x}$.

Note that for each image, we only randomly select one (t_i, ϵ_i) from the small number of sampling pairs $\{(t_i, \epsilon_i)\}_{i=1}^{N}$ (e.g., $N = 25$) used for inference to compute the prediction score. This allows the information under the sampling steps that need to be applied when inference to be learned into the prompts at a faster rate.

For diffusion classifiers, probabilities for classes are not directly obtained, but are estimated by expectation in Eq. 5. Although we can theoretically get the probabilities by applying softmax to the expectation, in practice the probabilities are very close and noisy. To get calibrated probabilities, we convert these expectation into probabilities by applying softmax with temperature:

$$p_\theta(y = \mathrm{y}_k \mid \mathbf{x}) = \frac{\exp(-S_{\mathbf{c}_k}(\mathbf{x})/\tau)}{\sum_{\mathbf{c}_j \in [\mathbf{c}_K]} \exp(-S_{\mathbf{c}_j}(\mathbf{x})/\tau)}, \tag{9}$$

where τ is the temperature parameter.

Finally, conditional prompts at each granularity are learned by minimizing the cross-entropy loss on the training dataset. Take two levels as an example, "order" (coarse) and "species" (fine) granularity. The final loss can be defined as:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}}(p_\theta(y_{order} \mid \mathbf{x}), \mathrm{y}_{\mathrm{order}}) + \mathcal{L}_{\mathrm{CE}}(p_\theta(y_{species} \mid \mathbf{x}), \mathrm{y}_{\mathrm{species}}), \tag{10}$$

where $p_\theta(y_{order} \mid \mathbf{x})$ and $p_\theta(y_{species} \mid \mathbf{x})$ denote the prediction probabilities of the "order" and "species" level respectively, $\mathcal{L}_{\mathrm{CE}}$ denotes the cross-entropy loss.

During inference, We apply the learned prompts to the hierarchical evaluation strategy. The fine-tuned learnable vectors and class names are recomposed as prompts for each granularity.

[1] M is set to 16, the same as CoOp [50]. During inference, the additional computation introduced by the learnable text is minimal and does not affect inference speed.

4 Experiments

In Sec. 4.1 we test our approach in the fine-grained visual classification task. In Sec. 4.2, we conduct experiments to test the out-of-distribution robustness. In Sec. 4.3, we conduct ablation study. In Sec. 4.4, we do further analysis comparing the speed of our method with the baseline and analyzing why the learned prompts are useful through visualization.

Implementation details We use Stable Diffusion v1.5, a latent text-to-image diffusion model. It utilizes the pre-trained text encoder from CLIP to encode text and a pre-trained variational autoencoder to map images to a latent space. With 860M parameters, the model takes 512x512-resolution images as input.

When inference, the number of sampling timesteps for each class is set to 25. During training, we randomly select 1 timestep from the 25 sampling timesteps for each image. We fix the length of the learnable context vectors to 16 and initialize them with Gaussian noise. Learnable context vectors are trained by minimizing the cross-entropy loss with SGD for 30 epochs. Initial learning rate is set to 0.001, which is decayed by the cosine annealing rule. All experiments are conducted on a single NVIDIA A800. In Sec. 4.1, We follow the few-shot evaluation protocol used in CoOp [50], training with 1, 2, 4, 8, and 16 shots respectively, and testing over the full test sets. In Sec. 4.2, as for fine-grained domain generalization, we train with 16 shots on CUB [45] and CUB-Paintings [46], respectively, and test on another dataset. As for robustness of corruption, we train with 16 shots on CIFAR-10 [31] and test on CIFAR-10-C [22].

4.1 Fine-Grained Visual Classification

Datasets We evaluate our proposed method on fine-grained visual classification task with four widely used FGVC datasets: CUB-200-2011 [45], Aircraft [34], Stanford Cars [30] and Oxford Pets [36]. To construct a taxonomy of label hierarchy for CUB-200-2011 and Stanford Cars, we learn from the work of Chang et al. [7] and use GPT-3 [5] to check for corrections. Details on the datasets and how to apply the method to other datasets are in the supplementary material.

Baselines We compare our method with the following zero-shot or few-shot models. **Diffusion Classifier** [32] is a zero-shot classifier using density estimates from large-scale text-to-image diffusion models such as Stable Diffusion without any additional training. In fairness, we choose to use the results of experiments with Stable Diffusion v1.5, the same as ours. **CLIP ResNet-50** [38] with manual prompts is a strong discriminative zero-shot model. **CoOp** [50] is a few-shot classification model, which resembles soft-prompting method applied to VLMs with limited labeled examples. Our approach differs from CLIP and CoOp in several aspects such as training data, model size, and model structure of the pre-trained model. Thus, the comparison is not absolutely fair.

Table 1. Comparison with Diffusion Classifier baseline on accuracy and computation speed. Our zero-shot method represents only adding hierarchical evaluation strategy to baseline. Our few-shot method represents adding hierarchical evaluation strategy and prompt learning to baseline. Time denotes inference time per image (s).

Model		Aircraft			Stanford Cars			CUB-200-2011			Oxford Pets		
Type	shots	Acc	Time	Speedup	Acc	Time	Speedup	Acc	Time	Speedup	Acc	Time	Speedup
baseline [32]	zero-shot	29.10	47.13	1.0x	77.64	53.04	1.0x	45.63	61.20	1.0x	87.30	17.50	1.0x
Ours	zero-shot	30.03	18.76	2.5x	77.58	17.67	3.0x	44.03	17.49	3.5x	86.11	8.50	2.1x
Ours	16	47.78	4.15	11.4x	84.17	2.61	20.3x	70.47	8.91	6.9x	86.37	1.96	8.9x

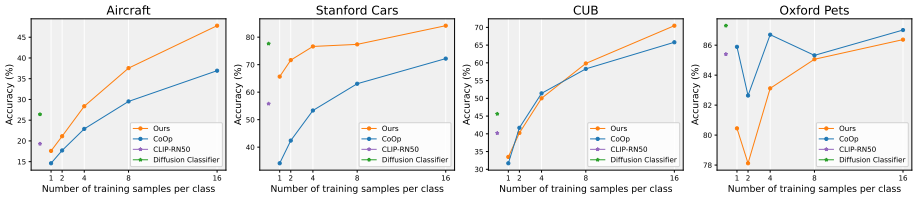

Fig. 3. Main results of few-shot learning on the 4 datasets.

Comparison with Diffusion Classifier Our experimental results in Table 1 demonstrate the improvement of our method on the inference speed and accuracy of Diffusion Classifier baseline. When the hierarchical evaluation strategy is used, the inference speed can be improved by 2-3.5x on the four datasets while the accuracy keeping basically the same. Combining prompt learning with it, in order to further improve the speed, we choose to reduce the number of sampling steps for each category to 1/10 of the baseline or even more. Compared to the baseline, hierarchical prompting (our few-shot method) improves inference speedup by 11x, 20x, 7x, and 9x on the Aircraft, Stanford Cars, CUB-200-2011 and Oxford Pets respectively. Accuracy can definitely be improved due to our fine-tuning in the downstream dataset. However, we can achieve a boost with less training data. In the 16-shots setting, we achieve improvement by 18.68% on Aircraft, 6.53% on Stanford Cars, and 24.84% on CUB-200-2011. And it is comparable on Oxford Pets with much reduced inference time. Note that in the case of increasing the number of sampling timesteps per class, our accuracy can be much higher.

Comparison with disciminative methods Table 2 shows the comparison between our method and other discriminative models. Our zero-shot method significantly outperforms CLIP ResNet-50. As for few-shot classification, the average improvement over the four datasets is 7.29% compared to CoOp at 16-shots setting. Our few-shot method can achieve 14.5% and 10.62% higher than CoOp on the Aircraft and Stanford Cars datasets and is competitive on the CUB-200-2011 and Oxford Pets datasets. Fig. 3 exhibits the comparison results of accuracy at 1, 2, 4, 8, and 16 shots.

Table 2. Percent accuracies for fine-grained visual classification compared with other discriminative few-shot/ zero-shot classification methods.

Model		Aircraft	Stanford Cars	CUB-200-2011	Oxford Pets	Average
Type	Shots					
CLIP-RN50 [38]	zero-shot	19.30	55.80	40.21	85.40	50.18
Ours	zero-shot	30.03	77.58	44.03	86.11	59.44
CoOp [50]	16	33.28	73.55	65.80	87.01	64.91
Ours	16	47.78	84.17	70.47	86.37	72.20

4.2 Robustness to Out-of-distribution

Compared to discriminative methods, diffusion classifier methods show stronger out-of-distribution (OOD) classification performance [11, 32]. We find that our method can achieve better results on out-of-distribution datasets on top of Diffusion Classifier baseline. In this section, we compare the out-of-distribution robustness of our method with baseline.

Datasets **(i)** For fine-grained datasets, we choose two domains: CUB and CUB-Paintings dataset. CUB-Paintings is a dataset of bird paintings with the same class list (200 classes) as CUB. It includes watercolors, oil paintings, pencil drawings stamps, and cartoons. **(ii)** In order to comprehensively evaluate the model's robustness to a given type of corruption, we select CIFAR-10 as source dataset and CIFAR-10-C as target dataset. CIFAR-10-C dataset offers multiple corrupted versions of the original CIFAR-10 test set including noise, blur, weather, and digital distortions. Overall, there are 15 different types of perturbations and corruptions, each available in 5 severity levels, enabling the study of performance under increasing data shifts.

Results Table 3 shows the results of domain generalization on fine-grained datasets. Our method achieves better performance across two transfer tasks. We raise average accuracy from the baseline of 32.05% to 47.12%, a boost of more than 15 percent. As for the experiments to explore robustness to different corruptions, Fig. 4 shows improved effective robustness on the CIFAR-10-C distribution shift. Compared with baseline, our method improves by 6.74% on average across 5 severity levels under 15 different types of corruptions. Two experimental results show that our method improves the robustness to out-of-distribution changes.

Table 3. Effect of context length on accuracy.

Model	C→P	P→C	Avg
baseline [32]	25.51	38.58	32.05
Ours	37.79	56.62	47.21

4.3 Ablation Study

Effect of the number of sampling timesteps per class The number of sampling timesteps per category is one factor that affects the inference speed of

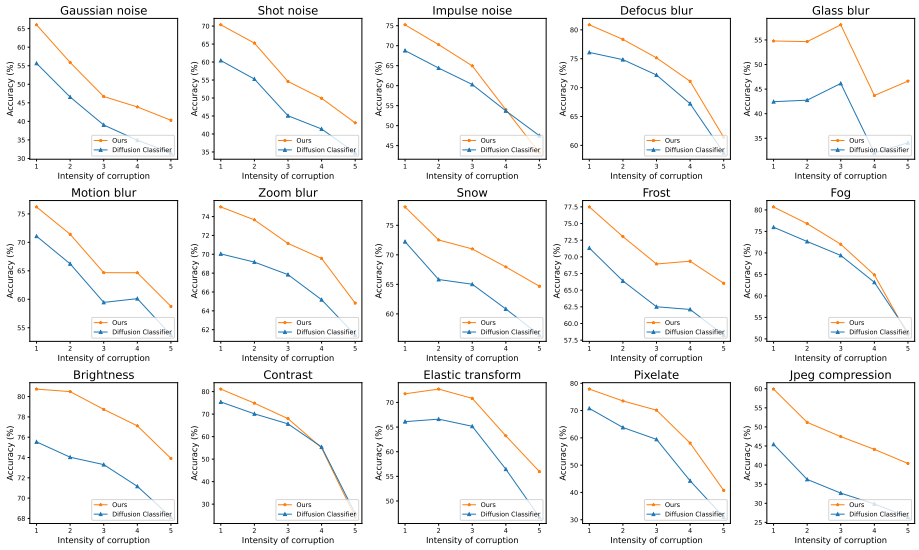

Fig. 4. Robust accuracy on CIFAR-10-C for our method and Diffusion Classifier. Our method shows better performance.

diffusion classifier models. Fig. 5 illustrates the changing behavior of the accuracy and inference time of our method with the number of sampling timesteps per category. We test the results on the Oxford Pets dataset when the number of sampling steps is 1, 5, 25, 120 and 250. It can be seen that as the number of sampling steps per category increases, the accuracy and the time increases gradually. This is due to the fact that selecting more samples used for Monte Carlo estimation improves its estimation accuracy but also increases the computational time. We can also see that the increase in accuracy is more drastic when the number of sampling steps is less than 25. Therefore, in the final experiment we choose a number of sampling timesteps per category of 25.

Effect of the number of hierarchies To explore the impact of hierarchical structure, we test with 1, 2, and 3 hierarchical structures on the Aircraft dataset. Fig. 6 demonstrates the decrease in inference time and accuracy as the number of hierarchies increases. As the number of hierarchies increases, the number of classes that needs to be estimated decreases resulting in faster inference. However, the accuracy decreases due to the accumulation of errors in the upper hierarchy. For example, when the number of hierarchies changes from 2 to 3, the speed of inference achieves 1.6x speed-ups and the accuracy decreases by 5.94%. Thus it's necessary to find the trade-off between performance and inference time when determining the number of hierarchies.

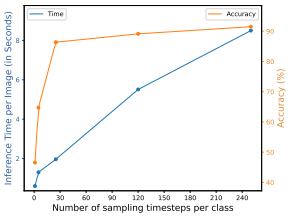

Fig. 5. Investigation on the effect of the number of sampling timesteps per class.

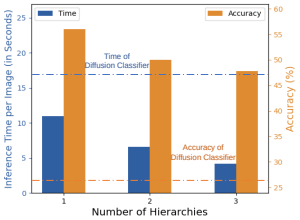

Fig. 6. Investigation on the effect of the number of hierarchies.

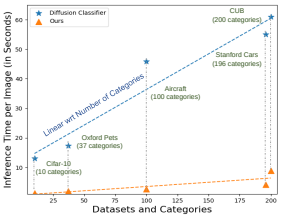

Fig. 7. Comparison of inference speed across different datasets.

Effect of the design of learnable prompts **(i)** We experiment with different context lengths (4, 8, and 16) on the Pets dataset to assess their impact. As depicted in Table 4, increasing the context length generally enhances accuracy, likely because more parameters are trained, enabling prompts to capture additional semantic information relevant to the task. However, the gains between different context lengths are modest, suggesting potential for parameter reduction without significant loss in effectiveness. **(ii)** We investigate the effectiveness of class-specific prompts compared to unified prompts on the Pets dataset. Results indicate that using class-specific prompts boosts accuracy by 1.02% over unified prompts. Class-specific prompts facilitate learning distinct characteristics of each category, thereby improving differentiation between categories.

Table 4. Effect of context length on accuracy.

Context Length	Acc(%)
4	85.11
8	85.39
16	86.37

4.4 Further Analysis

How does inference speed vary across datasets? We compare the inference time of our method with Diffusion Classifier as the number of categories in the dataset grows in Fig. 7. Diffusion Classifier requires to do the forward pass for each category separately to obtain its conditional likelihood. In this way, the inference time basically increases linearly with the number of categories in the dataset. For our method, on the other hand, the inference time is greatly reduced due to the introduction of hierarchical evaluation and prompt learning. Although still showing a linear relationship, the growth is substantially reduced. We also see that the speedup of our method is often more pronounced when the number of categories is larger.

Why are learned prompts useful? Compared to discriminative models, diffusion classifier models offer clearer visual insights into decision-making processes. In this section, we explain the role of learned prompts by image generation. Using Stable Diffusion v1.5, we apply prompt control to edit images based on both hand-crafted (e.g., "a photo of a {CLASS}, a type of bird.") prompts

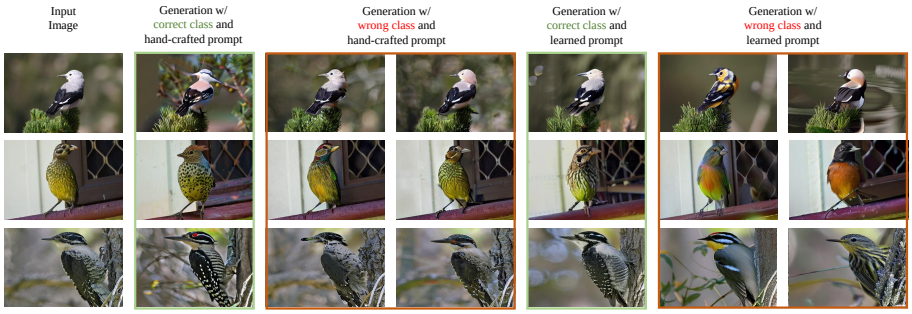

Fig. 8. Comparison of image generation for hand-crafted prompts and learned prompts.

and learned prompts in our method. As shown in Fig. 8, images generated with hand-crafted prompts (column 2) often diverge from the input image (column 1), whereas those generated with learned prompts (column 5) align more closely. For incorrect classifications, images generated with hand-crafted prompts (columns 3, 4) closely resemble the input (column 1), while those with learned prompts (columns 6, 7) highlight features of the wrong class (e.g., color and feather patterns of a hooded merganser in row 1, column 6), thereby magnifying the disparity from the input image. This demonstrates that learned prompts in our method effectively capture class-specific details, improving alignment with corresponding classes and enhancing scores for correct classifications while reducing scores for incorrect ones.

5 Discussion and Conclusion

In this paper, we introduce Hierarchical Prompt Learning, a method leveraging hierarchical evaluation and prompt learning techniques. We achieve significant speedups and maintains high accuracy for diffusion classifiers in zero-shot and few-shot classification, both in-distribution and out-of-distribution. Visualization analysis shows the effectiveness of what we have learned about prompts. However, Although achieving results, it requires additional training costs, which we aim to mitigate in future research. Finally, while our method has significantly improved in speed over previous diffusion classifiers, it is still not comparable to other discriminative methods (CLIP takes about 60ms to recognize an image). Using Consistency Models to improve speed may be an avenue for future work.

Acknowledgments. This work was supported by the National Nature Science Foundation of China (Grant 62225601, U23B2052, 62406171, 62476029), in part by the Beijing Natural Science Foundation Project No. L242025, in part by the Youth Innovative Research Team of BUPT No. 2023YQTD02, in part by the China Postdoctoral Science Foundation No. 2023M741961, in part by the Postdoctoral Fellowship Program of CPSF No. GZB20240359, and in part by BUPT Excellent Ph.D. Students Foundation No. CX2023112.

References

1. Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., Fleet, D.J.: Synthetic data from diffusion models improves imagenet classification. arXiv preprint arXiv:2304.08466 (2023)
2. Baranchuk, D., Rubachev, I., Voynov, A., Khrulkov, V., Babenko, A.: Label-efficient semantic segmentation with diffusion models. arXiv preprint arXiv:2112.03126 (2021)
3. Basu, S., Hu, S., Massiceti, D., Feizi, S.: Strong baselines for parameter-efficient few-shot fine-tuning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 11024–11031 (2024)
4. Bose, S., Jha, A., Fini, E., Singha, M., Ricci, E., Banerjee, B.: Stylip: Multi-scale style-conditioned prompt learning for clip-based domain generalization. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 5542–5552 (2024)
5. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
6. Chang, D., Ding, Y., Xie, J., Bhunia, A.K., Li, X., Ma, Z., Wu, M., Guo, J., Song, Y.Z.: The devil is in the channels: Mutual-channel loss for fine-grained image classification. IEEE Transactions on Image Processing **29**, 4683–4695 (2020)
7. Chang, D., Pang, K., Zheng, Y., Ma, Z., Song, Y.Z., Guo, J.: Your" flamingo" is my" bird": Fine-grained, or not. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11476–11485 (2021)
8. Chang, D., Tong, Y., Du, R., Hospedales, T., Song, Y.Z., Ma, Z.: An erudite fine-grained visual classification model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7268–7277 (2023)
9. Chen, J., Wang, P., Liu, J., Qian, Y.: Label relation graphs enhanced hierarchical residual network for hierarchical multi-granularity classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4858–4867 (2022)
10. Chen, J., Chang, D., Xie, J., Du, R., Ma, Z.: Cross-layer feature based multi-granularity visual classification. In: 2022 IEEE International Conference on Visual Communications and Image Processing (VCIP). pp. 1–5. IEEE (2022)
11. Clark, K., Jaini, P.: Text-to-image diffusion models are zero shot classifiers. Advances in Neural Information Processing Systems **36** (2024)
12. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)
13. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. arXiv preprint arXiv:1605.09782 (2016)
14. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
15. Du, R., Chang, D., Hospedales, T., Song, Y.Z., Ma, Z.: Demofusion: Democratising high-resolution image generation with no $$$. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6159–6168 (2024)
16. Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A.H., Chechik, G., Cohen-Or, D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. arXiv preprint arXiv:2208.01618 (2022)

17. Ge, C., Huang, R., Xie, M., Lai, Z., Song, S., Li, S., Huang, G.: Domain adaptation via prompt learning. IEEE Transactions on Neural Networks and Learning Systems (2023)
18. Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A.: Shortcut learning in deep neural networks. Nature Machine Intelligence **2**(11), 665–673 (2020)
19. Harshvardhan, G., Gourisaria, M.K., Pandey, M., Rautaray, S.S.: A comprehensive survey and analysis of generative models in machine learning. Computer Science Review **38**, 100285 (2020)
20. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16000–16009 (2022)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
22. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261 (2019)
23. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
24. Höllein, L., Müller, N., Novotny, D., Tseng, H.Y., Richardt, C., Zollhöfer, M., Nießner, M., et al.: Viewdiff: 3d-consistent image generation with text-to-image models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5043–5052 (2024)
25. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: International conference on machine learning. pp. 2790–2799. PMLR (2019)
26. Islam, K., Zaheer, M.Z., Mahmood, A., Nandakumar, K.: Diffusemix: Label-preserving data augmentation with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 27621–27630 (2024)
27. Karimi Mahabadi, R., Henderson, J., Ruder, S.: Compacter: Efficient low-rank hypercomplex adapter layers. Advances in Neural Information Processing Systems **34**, 1022–1035 (2021)
28. Kim, M., Hospedales, T.: Bayestune: Bayesian sparse deep model fine-tuning. Advances in Neural Information Processing Systems **36** (2024)
29. Koh, P.W., Sagawa, S., Marklund, H., Xie, S.M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R.L., Gao, I., et al.: Wilds: A benchmark of in-the-wild distribution shifts. In: International conference on machine learning. pp. 5637–5664. PMLR (2021)
30. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 554–561 (2013)
31. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
32. Li, A.C., Prabhudesai, M., Duggal, S., Brown, E., Pathak, D.: Your diffusion model is secretly a zero-shot classifier. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2206–2217 (2023)
33. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International conference on machine learning. pp. 12888–12900. PMLR (2022)

34. Maji, S., Rahtu, E., Kannala, J., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151 (2013)
35. Nayak, N.V., Yu, P., Bach, S.H.: Learning to compose soft prompts for compositional zero-shot learning. arXiv preprint arXiv:2204.03574 (2022)
36. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and dogs. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3498–3505. IEEE (2012)
37. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016)
38. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
39. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)
40. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: 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)
41. Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., Norouzi, M.: Palette: Image-to-image diffusion models. In: ACM SIGGRAPH 2022 conference proceedings. pp. 1–10 (2022)
42. Shi, W., Gong, Y., Tao, X., Cheng, D., Zheng, N.: Fine-grained image classification using modified dcnns trained by cascaded softmax and generalized large-margin losses. IEEE transactions on neural networks and learning systems **30**(3), 683–694 (2018)
43. Singh, M., Duval, Q., Alwala, K.V., Fan, H., Aggarwal, V., Adcock, A., Joulin, A., Dollár, P., Feichtenhofer, C., Girshick, R., et al.: The effectiveness of mae pre-pretraining for billion-scale pretraining. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5484–5494 (2023)
44. Trabucco, B., Doherty, K., Gurinas, M., Salakhutdinov, R.: Effective data augmentation with diffusion models. arXiv preprint arXiv:2302.07944 (2023)
45. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
46. Wang, S., Chen, X., Wang, Y., Long, M., Wang, J.: Progressive adversarial networks for fine-grained domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9213–9222 (2020)
47. Wang, Z., Jiang, Y., Zheng, H., Wang, P., He, P., Wang, Z., Chen, W., Zhou, M., et al.: Patch diffusion: Faster and more data-efficient training of diffusion models. Advances in neural information processing systems **36** (2024)
48. Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., De Mello, S.: Open-vocabulary panoptic segmentation with text-to-image diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2955–2966 (2023)
49. Yu, T., Xiao, T., Stone, A., Tompson, J., Brohan, A., Wang, S., Singh, J., Tan, C., Peralta, J., Ichter, B., et al.: Scaling robot learning with semantically imagined experience. arXiv preprint arXiv:2302.11550 (2023)
50. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. International Journal of Computer Vision **130**(9), 2337–2348 (2022)