
Beyond the Seen: Bounded Distribution Estimation for Open-Vocabulary Learning

Xiaomeng Fan^{1*} Yuchuan Mao^{1*} Zhi Gao^{1†}

Yuwei Wu^{1,2} Jin Chen¹ Yunde Jia^{2,1†}

¹Beijing Key Laboratory of Intelligent Information Technology,
School of Computer Science & Technology, Beijing Institute of Technology

²Guangdong Laboratory of Machine Perception and Intelligent Computing,
Shenzhen MSU-BIT University

<https://github.com/Beyond-the-Seen-NeurIPS>

Abstract

Open-vocabulary learning requires modeling the data distribution in open environments, which consists of both seen-class and unseen-class data. Existing methods estimate the distribution in open environments using seen-class data, where the absence of unseen classes makes the estimation error inherently unidentifiable. Intuitively, learning beyond the seen classes is crucial for distribution estimation to bound the estimation error. We theoretically demonstrate that the distribution can be effectively estimated by generating unseen-class data, through which the estimation error is upper-bounded. Building on this theoretical insight, we propose a novel open-vocabulary learning method, which generates unseen-class data for estimating the distribution in open environments. The method consists of a class-domain-wise data generation pipeline and a distribution alignment algorithm. The data generation pipeline generates unseen-class data under the guidance of a hierarchical semantic tree and domain information inferred from the seen-class data, facilitating accurate distribution estimation. With the generated data, the distribution alignment algorithm estimates and maximizes the posterior probability to enhance generalization in open-vocabulary learning. Extensive experiments on 11 datasets demonstrate that our method outperforms baseline approaches by up to 14%, highlighting its effectiveness and superiority.

1 Introduction

Open-vocabulary learning, an increasingly prominent task in computer vision, aims to recognize objects for both seen and unseen classes in open environments [63, 47, 78]. Effectively modeling the data distribution in open environments requires capturing both seen-class and unseen-class distributions. However, existing methods estimate open-environment distributions only based on seen-class data [46, 76, 75], and the absence of unseen classes makes it challenging to obtain accurate distribution estimation in open environments.

In this paper, we study how to learn beyond the seen classes in open environments. We derive distribution estimation theorems that prove the distribution can be estimated by generating unseen-class data, with an upper bound on the estimation error. Furthermore, these theorems reveal that narrowing the distribution gap between seen-class and generated unseen-class data tightens this upper bound, leading to more precise estimation. Motivated by these theoretical insights, it is desirable to generate unseen-class data that closely aligns with the seen-class data distribution, enabling a

*Equal contribution.

†Corresponding authors.

more accurate distribution estimation in open environments. To this end, we propose a novel open-vocabulary learning method, which generates unseen-class data for distribution estimation in open environments. The method is composed of a class-domain-wise data generation pipeline and a distribution alignment algorithm.

The class-domain-wise data generation pipeline consists of three key components: a hierarchy-guided unseen class predictor, a caption-based domain information generator, and a text-to-image model. These components collaboratively generate unseen-class data while minimizing distribution differences from the seen-class data. Specifically, the hierarchy-guided unseen class predictor leverages a hierarchical semantic tree, constructed from seen classes (leaf nodes) and their superclasses (parent nodes). This tree is expanded with candidate unseen classes sourced from WordNet [14] or large language models (LLMs) [58, 59]. The predictor selects the most relevant unseen class by identifying the nearest leaf node to ensure a minimal distribution distance to seen classes. The caption-based domain information generator extracts domain attributes, such as styles and backgrounds, from the seen-class data via image captioning. The text-to-image model (*e.g.*, Stable Diffusion [48]) is utilized to generate unseen-class data, with the guidance of the generated domain information and predicted unseen classes. Supported by our theoretical guarantees, this process significantly reduces the distribution gap, enabling accurate distribution estimation in open environments.

With the generated data, the distribution alignment algorithm is proposed to estimate and maximize the posterior probability of model outputs in open environments. We derive an evidence lower bound (ELBO) of the logarithmic posterior, which can be approximated as the expectation of logarithmic posterior on seen-class data minus the Kullback–Leibler (KL) divergence between the distributions of seen-class and generated unseen-class data. Hence, we employ a KL-based loss to minimize the distribution gap between the seen-class and generated unseen-class data. However, due to inherent variations in mini-batches, enforcing strict alignment in every iteration introduces misalignment, thereby degrading learning performance. To address this, we propose a sparse loss computation strategy that accumulates output distributions across iterations and then minimizes the alignment loss periodically. This approach effectively mitigates misalignment while ensuring that the posterior probability is maximized, thereby enhancing the generalization capability in open environments.

We evaluate the proposed method in open environments across two settings: base-to-base/base-to-new, and cross-dataset, using 11 image recognition datasets. Our method consistently outperforms the baseline on all datasets across two settings, demonstrating its effectiveness. Notably, on the EuroSAT dataset, our method achieves significant improvements of 14% and 9.48% in the base-to-new and cross-dataset settings, respectively, highlighting its effectiveness and superiority. Furthermore, ablation studies confirm the impact of the proposed class-domain-wise data generation pipeline and the distribution alignment algorithm. The results reveal that reducing the distribution distance between seen-class and generated unseen-class data indeed enhances performance in open environments, confirming the critical role of the proposed method. The contributions can be summarized as:

- We present a theoretical analysis that demonstrates the distributions in open environments can be effectively approximated by generating unseen-class data, with an upper bound on estimation error.
- We propose a novel open-vocabulary learning method, which introduces a class-domain-wise data generation pipeline to generate unseen-class data and a distribution alignment algorithm to accurately estimate and utilize the distribution in open environments for enhancing performance in open environments.

2 Related Work

2.1 Open-Vocabulary Learning

Existing open-vocabulary learning methods can be categorized into pre-training and prompt learning. Pre-training methods, such as CLIP [46] and ALIGN [22], train vision-language models (VLMs) on large-scale image-text pairs (*e.g.*, 400M to 1B) to learn rich multi-modal representations. Recent efforts focus on scaling datasets (*e.g.*, Datacomp [15], LAION-5B [54]) or improving training strategies (*e.g.*, caption diversity enhancement [32], fine-grained semantic alignment [34], masked cross-modal learning [56], scalable training optimization [62], multimodal guidance alignment [29]).

However, these methods often require retraining from scratch, which is resource-intensive in terms of time, data, and annotations.

Prompt learning methods address the retraining issue by introducing learnable prompt tokens at the input [17]. Initially successful in NLP tasks [30, 33, 38], these methods are adapted for vision-language models (VLMs). CoOp pioneers continuous prompt optimization in the language branch [76], while CoCoOp improves generalization by generating conditional prompts based on visual features [75]. VPT extends this to the visual branch by optimizing visual prompt tokens [23]. Recent advancements include multi-modal prompt fusion [24, 72, 68], regularization techniques [77, 31, 25, 42], and leveraging local VLM features to enhance performance [28, 57, 6]. These methods focus on estimating the open-environment distribution using seen-class data without theoretical guarantees for upper-bounded estimation error. In contrast to existing methods that rely solely on seen-class data, we explore learning beyond the seen by generating unseen-class data for accurate and bounded distribution estimation in open environments.

2.2 Learning from Synthetic Data

Synthetic data enhances performance across computer vision tasks like object detection [45, 51], semantic segmentation [8, 50], autonomous driving [1], and robotics [40, 71]. Recent text-to-image models, powered by diffusion techniques, generate high-quality images from text [52, 4, 48]. Existing methods combine descriptive prompts and classes to create synthetic images [12], which, when paired with real data, improve tasks like image classification [2, 11, 18, 36, 53, 60], object detection [7, 66, 74], and semantic segmentation [16, 26, 44, 64, 65, 69]. Unlike these methods that focus on generation of seen classes, our method generates images for unseen classes to precisely estimate the open-environment distribution.

3 Preliminaries

3.1 CLIP

We implement our method on CLIP [46] that consists of an image encoder $f_{\Phi_1}(\cdot)$ and a text encoder $g_{\Phi_2}(\cdot)$, where parameters are $\Phi = [\Phi_1, \Phi_2]$. Given an image x , the image encoder embeds x and adds a learnable class token to obtain the visual feature. Then, the text encoder projects the corresponding class label y wrapped within a text template to get the textual feature. Given image x and a ground-truth \bar{y} from all classes, CLIP computes the posterior probability as

$$p(\bar{y}|x, \Phi) = \frac{\exp(\text{sim}(g_{\Phi_2}(\bar{y}), f_{\Phi_1}(x))/\tau)}{\sum_i \exp(\text{sim}(g_{\Phi_2}(y_i), f_{\Phi_1}(x))/\tau)}, \quad (1)$$

where $\text{sim}(\cdot, \cdot)$ is the cosine similarity and τ is a temperature parameter. In this paper, we utilize the prompt learning to optimize CLIP. We add learnable language and visual prompts given as v_1 and v_2 to the textual and visual inputs, respectively. Prompts $v = [v_1, v_2]$ are optimized with the loss as

$$\mathcal{L}_{\text{CE}} = -\log \frac{\exp(\text{sim}(\tilde{f}_p, \tilde{g}_{p\bar{y}})/\tau)}{\sum_i \exp(\text{sim}(\tilde{f}_p, \tilde{g}_{py_i})/\tau)}, \quad (2)$$

where $\tilde{f}_p = f_{\Phi_1}([v_1, x])$, and $\tilde{g}_{py_i} = g_{\Phi_2}([v_2, y_i])$.

3.2 Open-Vocabulary Learning

Open-vocabulary learning requires recognizing objects of unseen classes and seen classes in open environments. We denote the data in open environments as \mathcal{D}_o , which consists of image-label pairs $\{(x_o, y_o)\}$. Accordingly, \mathcal{D}_o can be divided into two disjoint datasets, *i.e.*, the seen-class dataset $\mathcal{D}_s = \{(x_s, y_s)\}$ and the unseen-class dataset $\mathcal{D}_u = \{(x_u, y_u)\}$. The corresponding label sets are denoted as $Y_o = \{y_o\}$, $Y_s = \{y_s\}$, $Y_u = \{y_u\}$, which satisfy that $Y_u \cap Y_s = \emptyset$ and $Y_u \cup Y_s = Y_o$. A unique aspect of open-vocabulary recognition tasks is the inclusion of language vocabulary knowledge encoded in a large vocabulary space, such as the description of textual classes.

Open-vocabulary learning tasks aim to maximize the model outputs of posterior $p(\bar{y}|x_o, \Phi)$. Intuitively, the posterior distribution $p(\bar{y}|x_o, \Phi)$ is strong related to $p(\bar{y}|x_s, \Phi)$ and $p(\bar{y}|x_u, \Phi)$. Existing

methods tend to utilize seen-class data to model $p(\bar{y}|\mathbf{x}_s, \Phi)$, which is further to estimate $p(\bar{y}|\mathbf{x}_o, \Phi)$. Ignoring $p(\bar{y}|\mathbf{x}_u, \Phi)$ results in significant estimation errors that cannot be guaranteed to be bounded, further impacting the generalization in open environments.

4 Theoretical Analysis

To facilitate open-vocabulary learning, we explore learning beyond the seen classes for accurate distribution estimation in open environments. In this section, we demonstrate that the distribution in open environments can be estimated by generating unseen-class data $\mathcal{G}_u = \{(\mathbf{x}_e, \mathbf{y}_e)\}$. The label set of \mathcal{G}_u is denoted as $\mathbf{Y}_e = \{\mathbf{y}_e\}$ that satisfies $\mathbf{Y}_e \cap \mathbf{Y}_s = \emptyset$ and $\mathbf{Y}_e \cup \mathbf{Y}_s = \mathbf{Y}_o$. Our theoretical analysis is conducted from two perspectives, *i.e.*, the joint probability distribution and the posterior probability distribution.

The joint probability distribution in open environments is denoted as $p(\mathbf{x}_o, \mathbf{y}_o)$. Due to $\mathbf{Y}_e \cap \mathbf{Y}_s = \emptyset$ and $\mathbf{Y}_u \cup \mathbf{Y}_s = \mathbf{Y}_o$, $p(\mathbf{x}_o, \mathbf{y}_o)$ can be modeled as

$$p(\mathbf{x}_o, \mathbf{y}_o) = p(\mathbf{x}_s, \mathbf{y}_s) + p(\mathbf{x}_u, \mathbf{y}_u), \quad (3)$$

where $p(\mathbf{x}_s, \mathbf{y}_s)$ and $p(\mathbf{x}_u, \mathbf{y}_u)$ denote the joint probability distribution of seen classes and unseen classes, and $p(\mathbf{x}_s, \mathbf{y}_s)$ can be directly modeled from seen-class data. We propose that $p(\mathbf{x}_u, \mathbf{y}_u)$ can be estimated by generating unseen-class data $\mathcal{G}_u = \{(\mathbf{x}_e, \mathbf{y}_e)\}$, where this estimation error is upper bounded, as shown in Theorem 1.

Theorem 1. *With probability at least $1 - \delta$, we have the following,*

$$d(p(\mathbf{x}_e, \mathbf{y}_e), p(\mathbf{x}_u, \mathbf{y}_u)) \leq \sqrt{\frac{d(p(\mathbf{x}_e, \mathbf{y}_e), p(\mathbf{x}_s, \mathbf{y}_s))}{2m-1}} + \sqrt{\frac{\ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m-1}}, \quad (4)$$

where $d(\cdot, \cdot)$ denotes the distribution distance, and m denotes the size of seen-class dataset.

This theorem demonstrates that the distance between the joint distributions of unseen-class and generated unseen-class data has an upper bound. In Theorem 1, we can observe that as the distance between the joint distributions of generated unseen-class data and seen-class data $d(p(\mathbf{x}_e, \mathbf{y}_e), p(\mathbf{x}_s, \mathbf{y}_s))$ decreases, the upper bound in Eq. (4) decreases. This indicates that we can narrow the gap between the joint distributions of generated unseen-class data and unseen-class data by reducing the distance between the joint distributions of generated unseen-class data and seen-class data. Theorem 1 holds for any distribution distance (*e.g.*, KL divergence, total variation distance, or other forms).

In terms of posterior probability distribution, we demonstrate that the estimation error between $p(\bar{y}|\mathbf{x}_e, \Phi)$ and $p(\bar{y}|\mathbf{x}_u, \Phi)$ is upper bounded, which is presented in Theorem 2. Without loss of generality, we define the distribution distance as KL divergence for analysis.

Theorem 2. *Given the predicted classes $\mathbf{Y}_e = \{\mathbf{y}_e\}$. Suppose that predicted class \mathbf{Y}_e have any nonzero probability $p(\mathbf{Y}_e)$. With probability at least $1 - \delta$ over the m instances of generated unseen-class data $\{(\mathbf{x}_e, \mathbf{y}_e)\}$, we have that*

$$D_{KL}(p(\bar{y}|\mathbf{x}_u, \Phi) || p(\bar{y}|\mathbf{x}_e, \Phi)) \leq \sqrt{\frac{\ln \frac{1}{p(\mathbf{Y}_e)} + \ln \frac{1}{\delta}}{2m}} - \ln \frac{|\mathbf{Y}_e|}{|\mathbf{Y}_u|}. \quad (5)$$

Discussion. $p(\mathbf{Y}_e)$ is the probability assigned to \mathbf{Y}_e under the distribution over \mathbf{Y}_u . Since $\mathbf{Y}_e \subset \mathbf{Y}_u$, $p(\mathbf{Y}_e) > 0$ always holds without any constraint on \mathbf{Y}_e .

Theorem 2 demonstrate that the estimation of the posterior probability distribution in open environments has an upper bound. From Theorem 2, we can observe that $\lim_{m \rightarrow \infty} \sqrt{\frac{\ln \frac{1}{p(\mathbf{Y}_e)} + \ln \frac{1}{\delta}}{2m}} = 0$, which indicates that increasing the amount of samples m can decrease the approximation error. As the probability $p(\mathbf{Y}_e)$ and $|\mathbf{Y}_e|$ increase, the approximation error bound decreases, which conforms to common sense. The step-by-step derivations are presented in the appendix.

From Theorems 1 and 2, we can conclude that the distance between the distributions of generated unseen-class data and unseen-class data in open environments has an upper bound. This provides the theoretical guarantee for the open-environment distribution estimation.

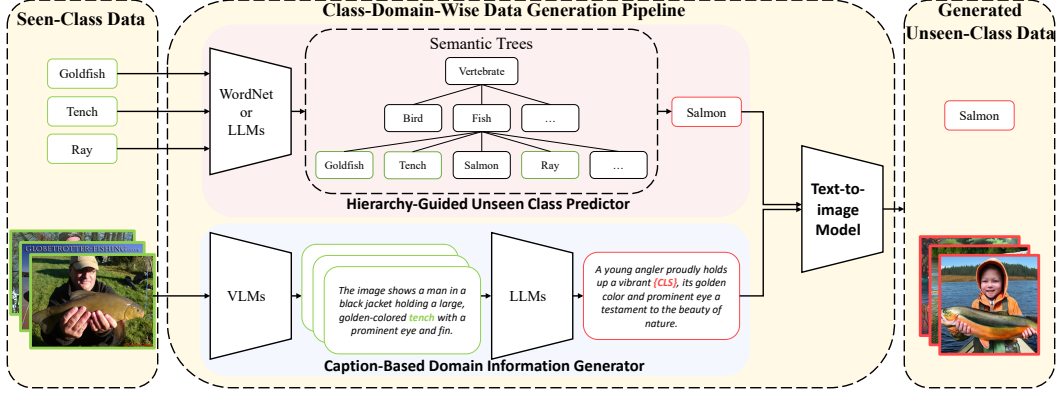


Figure 1: Formulation of Class-Domain-Wise Data Generation Pipeline

5 Method

Based on the theoretical analysis, we propose a novel open-vocabulary learning method that includes a class-domain-wise data generation pipeline to generate unseen-class data and a distribution alignment algorithm to estimate and utilize the estimated distribution for generalization.

5.1 Class-Domain-Wise Data Generation Pipeline

Inspired by Theorem 1 that indicates estimation error is related to the distribution distance between seen-class data and generated unseen-class data, our goal is to generate unseen-class data aligned with the seen-class data distribution. Our pipeline includes a **hierarchy-guided unseen class predictor** to identify classes close to seen-class data, a **caption-based domain information generator** to extract domain information of seen-class data, and a **text-to-image model**. These components work together to generate unseen-class data that align with the seen-class data distribution. The overall pipeline is shown in Figure 1.

5.1.1 Hierarchy-Guided Unseen Class Predictor

In open environments, the semantic structure of classes exhibits a prominent hierarchical nature, akin to the label space in ImageNet [10], which is also organized hierarchically. Motivated by this observation, the unseen class predictor leverages the semantic structure of seen classes to predict unseen classes.

We first construct a semantic tree for the seen classes \mathbf{Y}_s , where \mathbf{Y}_s are the leaf nodes and parent nodes represent superclasses derived from WordNet [14] or large language models (LLMs) [58, 59]. If WordNet contains the classes, superclasses are their hypernyms; otherwise, LLMs are queried. To predict unseen classes, we expand the semantic tree by adding missing hyponyms of the superclasses via WordNet or LLMs. These hyponyms, representing sibling nodes of the seen classes, serve as potential unseen class candidates. For each candidate, we compute its cosine similarity with seen classes using textual embeddings from text encoder $g_{\Phi_2}(\cdot)$, and select the top K_0 closest candidates as predicted unseen classes.

5.1.2 Caption-Based Domain Information Generator

Domain information (such as image style and scene information) plays important roles in the alignment of generated unseen-class data and seen-class data. We aim to capture the domain information from seen-class data by utilizing Vision-Language Models (VLMs) [37]. In doing so, we have to encounter two main issues. Firstly, the hallucination originated from VLMs may result in unmatched domain information. Secondly, domain information extracted from seen-class data may be limited, undermining the diversity of generated images.

To solve these issues, we utilize VLMs to generate class-specific captions for each classes, and then we calculate the similarity between these descriptions and the images of corresponding classes. By

selecting top K_1 results with the highest similarity, we ensure that the generated textual descriptions closely align with the image content, thereby effectively reducing hallucination issues. Then, the selected captions are summarized as top K_2 class-specific domain information by using LLMs.

5.1.3 Text-to-Image Model

The text-to-image model [49] is utilized to generate unseen-class data, with the guidance of the predicted unseen classes and the corresponding class-specific domain information.

Because the unseen classes are close to seen classes and domain information aligns with seen-class data, we narrow the distribution gap between the generated unseen-class data and the seen-class data.

5.2 Distribution Alignment

After generating unseen-class data $\mathcal{G}_u = \{(\mathbf{x}_e, \mathbf{y}_e)\}$, we estimate and maximize the posterior probability in open environments, thereby enhancing its generalization. The posterior probability distribution of model outputs in open environments satisfies that

$$p(\bar{\mathbf{y}}|\mathbf{x}_o, \Phi) \propto p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi). \quad (6)$$

The Evidence Lower Bound (ELBO) of the logarithmic posterior probability can be derived as

$$\log p(\bar{\mathbf{y}}|\mathbf{x}_o, \Phi) \geq \mathbb{E}[\log p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)] - D_{\text{KL}}(p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)||p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)), \quad (7)$$

where the proof is presented in the appendix. Due to the absence of $(\mathbf{x}_u, \mathbf{y}_u)$, we leverage the generated unseen-class data $(\mathbf{x}_e, \mathbf{y}_e)$ to estimate ELBO, *i.e.*,

$$\mathbb{E}[\log p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)] - D_{\text{KL}}(p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)||p(\bar{\mathbf{y}}|\mathbf{x}_e, \Phi)). \quad (8)$$

As Theorem 2 suggests that the KL divergence between $p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)$ and $p(\bar{\mathbf{y}}|\mathbf{x}_e, \Phi)$ is upper-bounded, this estimation in Eq. (8) is reasonable and practically effective. Thus, $\log p(\bar{\mathbf{y}}|\mathbf{x}_o, \Phi)$ in Eq. (54) can be maximized by minimizing $-\mathbb{E}[\log p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)]$ and $D_{\text{KL}}(p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)||p(\bar{\mathbf{y}}|\mathbf{x}_e, \Phi))$.

To this end, we design distribution alignment algorithm that adopts prompt learning to optimize CLIP. Specifically, we minimize the $\mathcal{L}_{\text{CE}}(\cdot)$ on \mathcal{D}_b in Eq. (2) to minimize $-\mathbb{E}[\log p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)]$. To minimize $D_{\text{KL}}(p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)||p(\bar{\mathbf{y}}|\mathbf{x}_e, \Phi))$, we introduce a KL-based loss for distribution alignment, which is formulated as

$$\mathcal{L}_{\text{KL}} = D_{\text{KL}}[p(\bar{\mathbf{y}}|\mathbf{x}_b, \Phi, \mathbf{v})||p(\bar{\mathbf{y}}|\mathbf{x}_e, \Phi, \mathbf{v})]. \quad (9)$$

By minimizing $\mathcal{L}_{\text{CE}}(\cdot)$ and $\mathcal{L}_{\text{KL}}(\cdot)$, $p(\bar{\mathbf{y}}|\mathbf{x}_o, \Phi)$ is maximized.

The proposed loss $\mathcal{L}_{\text{KL}}(\cdot)$ introduces additional challenges, *i.e.*, the data from unseen and seen classes in the mini-batch may differ significantly, and the loss could forcefully align their output distributions despite their large inherent differences. This misalignment between the distributions can, in turn, compromise the learning performance of both the base and unseen classes. To address the issue, we propose a sparse loss computation strategy that accumulates output distributions of seen-class data across iterations and then minimizes the alignment loss periodically. During each iteration, we save the output distributions of the seen-class data. For each batch of unseen-class data, we compute the similarity between the saved output distributions of seen-class data and that of the unseen-class data. Alignment is then performed on the top K_3 most similar distributions, which helps alleviate the misalignment problem by ensuring a more accurate and consistent alignment across batches.

In order to decrease the distribution distance between the generated data and real data in the feature spaces, we introduce a Maximum Mean Discrepancy (MMD) loss. Specifically, we first generate some extra seen-class data $\mathcal{G}_s = \{(\mathbf{x}'_s, \mathbf{y}_s)\}$ using the same generation pipeline as presented in Section 5.1. Then the MMD loss is formulated as

$$\mathcal{L}_{\text{MMD}} = \frac{1}{n^2} \sum_{i,j=1}^n K(\mathbf{x}_s^i, \mathbf{x}_s^j) + \frac{1}{n^2} \sum_{i,j=1}^n K(\mathbf{x}'_b^i, \mathbf{x}'_b^j) - \frac{2}{n^2} \sum_{i=1}^n \sum_{j=1}^n K(\mathbf{x}_i, \mathbf{x}'_j), \quad (10)$$

where n is batch size. $K(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$ represents Gaussian kernel. By minimizing the MMD loss, the distance of feature spaces between generated data and real data is decreased, further improving alignment. We also employ the sparse loss computation strategy for the MMD loss.

Overall, the loss function for updating parameters is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \alpha \mathcal{L}_{\text{KL}} + \beta \mathcal{L}_{\text{MMD}}, \quad (11)$$

where α and β are hyper-parameters. By minimizing $\mathcal{L}_{\text{total}}$, the posterior probability in open environments $p(\bar{\mathbf{y}}|\mathbf{x}_o, \Phi)$ is maximized, thereby improving capability in open environments. This algorithm is summarized in appendix.

Table 1: Results in base-to-new/base-to-base generalization setting. We bold the best results and underline the second-best results. H denotes the harmonic mean of performance on base and new.

Dataset		CLIP [46]	CoOp [76]	CoCoOp [75]	DePT [73]	TCP [70]	CuTCP [21]	DeKg [35]	PromptSRC (baseline)	Ours	Gain Δ
Average	Base	69.34	82.69	80.47	<u>85.19</u>	84.13	84.21	84.96	84.26	86.40	+2.14
	New	74.22	63.22	71.69	<u>76.17</u>	75.36	76.10	<u>76.38</u>	76.10	80.52	+4.42
	H	71.70	71.66	75.83	80.43	79.51	79.95	<u>80.44</u>	79.97	83.36	+3.39
ImageNet	Base	72.43	76.47	75.98	78.20	77.27	77.73	77.40	77.60	<u>77.91</u>	+0.31
	New	68.14	67.88	70.43	70.27	69.87	70.50	69.20	<u>70.73</u>	70.74	+0.01
	H	70.22	71.92	73.10	<u>74.02</u>	73.38	73.94	73.07	74.01	74.15	+0.14
Caltech	Base	96.84	98.00	97.96	98.57	98.23	98.47	98.64	98.10	98.97	+0.87
	New	94.00	89.81	93.81	94.10	94.67	<u>95.27</u>	95.20	94.03	95.85	+1.82
	H	95.40	93.73	95.84	96.28	96.42	<u>96.84</u>	<u>96.89</u>	96.02	97.38	+1.36
Pets	Base	91.17	93.67	95.20	<u>95.43</u>	94.67	95.07	94.47	95.33	96.01	+0.68
	New	97.26	95.29	97.69	<u>97.33</u>	97.20	<u>97.83</u>	97.76	97.30	98.27	+0.97
	H	94.12	94.47	<u>96.43</u>	<u>96.37</u>	95.92	<u>96.43</u>	96.09	96.30	97.12	+0.82
Cars	Base	63.37	78.12	70.49	80.80	80.80	80.23	<u>81.18</u>	78.27	82.93	+4.66
	New	74.89	60.40	73.59	<u>75.00</u>	74.13	74.27	74.75	74.97	80.81	+5.84
	H	68.65	68.13	72.01	<u>77.79</u>	77.32	77.13	<u>77.83</u>	76.58	81.86	+5.28
Flowers	Base	72.08	97.60	94.87	98.40	97.73	98.10	<u>98.58</u>	98.07	98.77	+0.70
	New	<u>77.80</u>	59.67	71.75	77.10	75.57	75.58	75.18	76.50	80.92	+4.42
	H	74.83	74.06	81.71	<u>86.46</u>	85.23	85.38	85.30	85.95	88.96	+3.01
Food	Base	90.10	88.33	90.70	<u>90.87</u>	90.57	90.47	90.73	90.67	91.39	+0.72
	New	91.22	82.26	91.29	<u>91.57</u>	91.37	<u>91.77</u>	91.55	91.53	92.99	+1.46
	H	90.66	85.19	90.99	<u>91.22</u>	90.97	91.11	91.14	91.10	92.18	+1.08
Aircraft	Base	27.19	40.44	33.41	<u>45.70</u>	41.97	42.43	45.20	42.73	48.98	+6.25
	New	36.29	22.30	23.71	<u>36.73</u>	34.43	36.37	35.09	<u>37.87</u>	44.03	+6.16
	H	31.09	28.75	27.74	<u>40.73</u>	37.83	39.17	39.51	40.15	46.37	+6.22
SUN	Base	69.36	80.60	79.74	<u>83.27</u>	82.63	83.00	82.52	82.67	83.64	+0.97
	New	75.35	65.89	76.86	<u>78.97</u>	78.20	78.23	78.30	78.47	80.15	+1.68
	H	72.23	72.51	78.27	<u>81.06</u>	80.35	80.55	80.35	80.52	81.86	+1.34
DTD	Base	53.24	79.44	77.01	<u>84.80</u>	82.77	83.00	83.80	83.37	85.53	+2.16
	New	59.90	41.18	56.00	61.20	58.07	59.40	59.66	<u>62.97</u>	71.50	+8.53
	H	56.37	54.24	64.85	71.09	68.25	69.24	69.70	<u>71.75</u>	77.89	+6.14
EuroSAT	Base	56.48	92.19	87.49	93.23	91.63	90.87	<u>94.02</u>	92.90	97.17	+4.27
	New	64.05	54.74	60.04	77.90	74.73	77.13	<u>81.69</u>	73.90	87.90	+14.00
	H	60.03	68.69	71.21	84.88	82.32	83.44	<u>87.42</u>	82.32	92.30	+9.98
UCF	Base	70.53	84.69	82.33	87.73	87.13	86.87	<u>88.06</u>	87.10	89.14	+2.04
	New	77.50	56.05	73.45	77.70	80.77	80.80	<u>81.77</u>	78.80	82.53	+3.73
	H	73.85	67.46	77.64	82.46	83.83	83.72	<u>84.80</u>	82.74	85.71	+2.97

6 Experiments

6.1 Experiment Settings

We evaluate our method on open-vocabulary benchmarks with 11 image recognition datasets, following the setting of the baseline method PromptSRC [25].

Datasets. We evaluate the proposed method on 11 image recognition datasets: ImageNet [10], Caltech101 (Caltech) [13], OxfordPets (Pets) [43], StanfordCars (Cars) [27], Flowers102 (Flowers) [41], Food101 (Food) [5], FGVC Aircraft (Aircraft) [39], SUN397 (SUN) [67], UCF101 (UCF) [55], DTD [9] and EuroSAT [19].

Benchmark Settings. We evaluate our method on two open-vocabulary learning benchmarks.

Table 2: Results of our method and state-of-the-art methods for cross-dataset evaluation.

	Source	Target										
	ImageNet	Caltech	Pets	Cars	Flowers	Food	Aircraft	SUN	DTD	EuroSAT	UCF	Average
CoOp	71.51	93.70	89.14	64.51	68.71	85.30	18.47	64.15	41.92	46.39	66.55	63.88
CoCoOp	71.02	94.43	90.14	65.32	71.88	86.06	22.94	67.36	45.73	45.37	68.21	65.74
ASPrompt	71.05	94.57	90.79	66.90	72.30	86.17	25.16	67.32	47.35	50.25	69.52	67.03
PromptSRC	71.27	93.60	90.25	65.70	70.25	86.15	23.90	67.10	46.87	45.50	68.75	65.81
Ours	71.22	93.87	90.46	67.36	72.88	86.61	25.14	67.68	48.27	54.98	69.44	67.68
Gain Δ	-0.05	+0.27	+0.21	+1.66	+2.63	+0.46	+1.24	+0.58	+1.40	+9.48	+0.72	+1.87

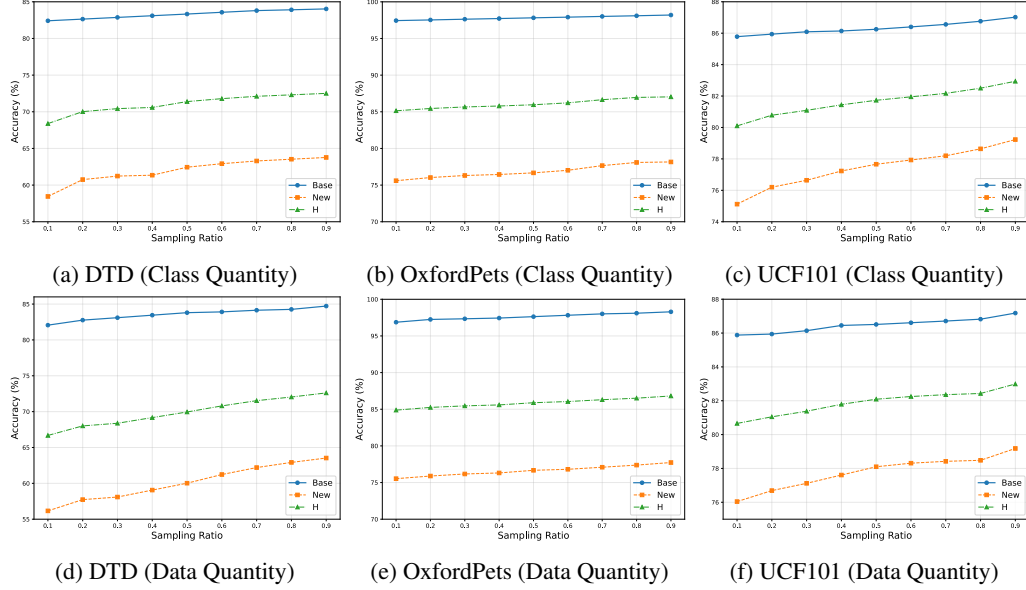


Figure 2: Ablation Studies on Quantity of Predicted Unseen Classes and Generated Images.

- **Base-to-base/Base-to-new generalization.** We equally split each dataset into base and new classes. The model is trained on base classes and evaluated on both base classes (base-to-base) and new classes (base-to-new) across all 11 datasets.
- **Cross-dataset evaluation.** To evaluate our method on the unseen classes under different domain environments, we adopt cross-dataset setting, where we train a model on ImageNet (source domain) and evaluate the model on the other 10 datasets (target domains) without any fine-tuning.

Implementation details. Following the setting of PromptSRC [25], we use a few-shot training strategy in all experiments at 16 shots which are randomly sampled for each class. We apply prompt learning on a pretrained ViT-B/16 based CLIP model and report results averaged over 3 runs. Other details such as hyper-parameters are provided in appendix.

6.2 Base-to-Base/Base-to-New Generalization

We compare our method with other prompt learning methods in Table 1. Results show that our method significantly improves the performance of PromptSRC in open environments for both base-to-base and base-to-new settings, demonstrating its effectiveness in open environments. Notably, on new classes, our method achieves up to a 14% performance improvement, indicating its ability to effectively estimate the distribution of unseen classes.

We also compare our method with other state-of-the-art methods: CLIP [46], CoOp [76], CoCoOp [75], DePT [73], TCP [70], CuTCP [21], and DeKg [35]. Results show that our method

Table 3: Ablation studies on class quality and data quality. “Acc” denotes the accuracy. “Dis” denotes the distribution distance of the generated unseen-class data and seen-class data.

Dataset		Ours		Class Quality				Data Quality					
		Acc ↑	Dis ↓	LowSim		w/o Tree		Picture		Photo		Image	
				Acc ↑	Dis ↓	Acc ↑	Dis ↓	Acc ↑	Dis ↓	Acc ↑	Dis ↓	Acc ↑	Dis ↓
Caltech	Base	98.97		98.26		97.93		98.26		98.19		98.06	
	New	95.85	9.99	94.32	10.49	93.67	13.16	94.21	11.84	94.11	11.95	93.78	12.12
	H	97.38		96.25		95.75		96.19		96.11		95.87	
Pets	Base	96.01		95.85		95.00		95.69		95.43		95.27	
	New	98.27	7.58	97.60	8.19	96.81	11.71	97.04	9.14	96.98	9.21	96.92	10.27
	H	97.12		96.72		95.90		96.36		96.20		96.09	
Cars	Base	82.93		78.94		77.86		77.99		78.71		78.24	
	New	80.81	8.92	75.29	9.33	73.36	13.78	74.75	10.65	75.04	10.08	74.92	10.48
	H	81.86		77.07		75.54		76.33		76.83		76.54	
Flowers	Base	98.77		98.29		97.15		97.82		97.91		97.63	
	New	80.92	6.29	77.52	6.88	75.04	10.29	76.88	7.99	77.09	7.84	76.17	8.22
	H	88.96		86.68		84.67		86.09		86.26		85.57	

Table 4: Ablation study on distribution alignment. “w/o da” denotes prompt learning without distribution alignment.

	Caltech	Pets	Cars	Flowers	Aircraft	DTD	EuroSAT	UCF								
	w/o da Ours	w/o da Ours	w/o da Ours	w/o da Ours	w/o da Ours	w/o da Ours	w/o da Ours	w/o da Ours								
Base	97.42	98.97	94.10	96.01	74.14	82.93	96.49	98.77	39.80	48.98	75.81	85.53	93.57	97.17	85.16	89.14
New	90.07	95.85	88.65	97.65	71.63	80.81	71.63	80.92	23.39	44.03	53.50	71.50	64.13	87.90	62.03	82.53
H	93.60	97.38	91.29	96.82	72.86	81.86	82.22	88.96	29.47	46.37	62.73	77.89	76.10	92.30	71.78	85.71

achieves the best performance, demonstrating its superiority in open environments, especially its generalization ability on unseen classes.

6.3 Cross-Dataset Evaluation

The comparison between our method and other prompt learning methods is presented in Table 2. Compared to PromptSRC, our method achieves the comparable performance on the source dataset. On the target datasets, our method significantly outperforms PromptSRC with a notable improvement of 9.48% on the Eurosat dataset, demonstrating that our method can improve the generalization on unseen classes across different domains, effectively working in open-vocabulary learning task.

6.4 Ablation Studies

6.4.1 Effectiveness of Class-Domain-Wise Data Generation Pipeline

We evaluate the effectiveness of two components with respect to this pipeline, *i.e.*, hierarchy-guided unseen class predictor and caption-based domain information generator.

Hierarchy-Guided Unseen Class Predictor. We evaluate the effectiveness by adjusting the quantity and quality of predicted unseen classes. As to the quantity, we randomly select $s_{cls} \times N$ classes from the N predicted unseen classes for training, where $s_{cls} < 1$ denotes sampling ratio. We conduct experiments on three datasets (DTD, Flowers102, UCF101), and we set s_{cls} as 0.1 to 0.9 in increments of 0.1 for each dataset. As shown in Figures 2a, 2b and 2c, we observe that increases of quantity lead to better performance, especially for new classes, which is consistent with Theorem 2.

As to the quality, we introduce “LowSim” that chooses candidate classes with lowest cosine similarity and “w/o Tree” that directly ask LLMs to query unseen classes. We calculate the distribution distance between the generated unseen-class data and the seen-class data. As shown in Table 3, our method

can significantly reduce the distribution distance and improve alignment with seen-class data. Results also reveal that alignment improves recognition, which is consistent with Theorem 1.

Caption-Based Domain Information Generator. We evaluate its effectiveness by adjusting the quantity and quality of generated images. As to the quantity, we randomly select $s_{imag} \times M$ images of each predicted unseen classes for training, where $s_{cls} < 1$ denotes sampling ratio and M denotes the amount of data for each classes. Figures 2d, 2e and 2f show that increase of quantity leads to better performance, especially for new classes, which is consistent with Theorem 2.

As to the quality, we modify the prompt template for data generation, without the domain information inferred from the generator. We inject "Picture", "Photo" and "Image" into stable diffusion model to generate images, which are denoted as the the prompt template "A picture of a {class}", "A photo of a {class}", and "An image of a {class}", respectively. Results are shown in Table 3. We observe that our method can significantly reduce the distribution distance and improve alignment with seen-class data. Results also reveal that this alignment improves recognition, which is consistent with Theorem 1.

To further evaluate the generated image quality, we computed CLIPScore [20]. CLIPScores on UCF101, DTD, SUN397, Caltech101, OxfordPets, and StanfordCars are 0.43, 0.42, 0.43, 0.43, 0.44, and 0.42, respectively. Results show that these images exhibit high semantic quality (CLIPScore > 0.35 is considered high quality [61, 3, 52]). We further conducted a user study and a GPT-4-based evaluation, where both human annotators and GPT-4 independently rated 200 randomly selected images on a 1–5 scale (5 is the highest quality). The resulting average scores of 4.67 (human) and 4.59 (GPT-4) confirm the high quality of the generated images.

6.4.2 Effectiveness of Distribution Alignment

We evaluate the effectiveness of the distribution alignment algorithm by directly using the generated images for prompt learning during training without the distribution alignment algorithm, denoted by "w/o pda". Results are shown in Table 5, which indicate that the proposed algorithm can improve the model performance in open environments by aligning the output distributions of model between generated data and real data.

6.5 Efficiency Analysis

We take Eurosat as an example for analysis. The training time and memory of our method requires at most 16.14 GB and 739.2 s. The baseline requires 6.12 GB and 101.25 s. The added time and memory mainly come from data generation. At inference time, no extra parameters or computation are introduced. Thus, our runtime (3.8 s) and memory (1.84 GB) are identical to the baseline.

7 Conclusion and Discussion

In this paper, we have investigated learning beyond the seen for bounded distribution estimation in the open-vocabulary task. We have demonstrated the distribution in open environments can be estimated by generating unseen-class data with upper-bounded estimation error, as evidenced by the constructed theoretical analysis. The proposed open-vocabulary learning method consists of a class-domain-wise data generation pipeline and a distribution alignment algorithm. The data generation pipeline generates unseen-class data via the introduced unseen class predictor and domain information generator, enabling accurate distribution estimation. With the generated data, the proposed distribution alignment algorithm can effectively estimate and maximize the posterior probability in open environments for improving generalization in open environments. Experiments on 11 datasets demonstrate that our method can generate unseen-class data for accurate distribution estimation, leading to consistent improvements in generalization across diverse open environments.

The generated data exists biases from the utilized LLMs and wordnet. In the future, we plan to design a multi-expert collaboration strategy that leverages diverse pretrained models and agreement-based selection to reduce dependence and bias on any single model. Moreover, to extend our method to support truly unknown classes, such as newly cartoon characters, we plan to integrate RAG mechanisms that dynamically retrieve emerging classes from external sources to enhance the ability of the model to generate up-to-date images, enabling the model to adapt in real time to new concepts.

Acknowledgements. This work was supported by the Shenzhen Science and Technology Program under Grant No. JCYJ20241202130548062, the Natural Science Foundation of Shenzhen under Grant No. JCYJ20230807142703006, the Natural Science Foundation of China (NSFC) under Grants No. 62406009, No. 62172041 and No. 62176021.

References

- [1] Hassan Abu Alhaija, Siva Karthik Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision*, 126:961–972, 2018.
- [2] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *Transactions on Machine Learning Research*.
- [3] Eslam Mohamed Bakr, Pengzhan Sun, Xiaoqian Shen, Faizan Farooq Khan, Li Erran Li, and Mohamed Elhoseiny. Hrs-bench: Holistic, reliable and scalable benchmark for text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20041–20053, 2023.
- [4] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3):8, 2023.
- [5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101—mining discriminative components with random forests. In *European Conference on Computer Vision*, pages 446–461, 2014.
- [6] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Plot: Prompt learning with optimal transport for vision-language models. *arXiv preprint arXiv:2210.01253*, 2022.
- [7] Kai Chen, Enze Xie, Zhe Chen, Yibo Wang, Lanqing Hong, Zhenguo Li, and Dit-Yan Yeung. Geodiffusion: Text-prompted geometric control for object detection data generation. *arXiv preprint arXiv:2306.04607*, 2023.
- [8] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1841–1850, 2019.
- [9] 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 (CVPR)*, 2014.
- [10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.
- [11] Lisa Dunlap, Alyssa Umino, Han Zhang, Jiezhi Yang, Joseph E Gonzalez, and Trevor Darrell. Diversify your vision datasets with automatic diffusion-based augmentation. *Advances in neural information processing systems*, 36:79024–79034, 2023.
- [12] Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. Scaling laws of synthetic images for model training... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7382–7392, 2024.
- [13] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, pages 178–178, 2004.
- [14] Christiane Fellbaum. Wordnet. *WordNet An Electronic Lexical Database*, page 69, 1998.
- [15] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruva Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36, 2024.

- [16] Rui Gong, Martin Danelljan, Han Sun, Julio Delgado Mangas, and Luc Van Gool. Prompting diffusion representations for cross-domain semantic segmentation. *arXiv preprint arXiv:2307.02138*, 2023.
- [17] Jindong Gu, Zhen Han, Shuo Chen, Ahmad Beirami, Bailan He, Gengyuan Zhang, Ruotong Liao, Yao Qin, Volker Tresp, and Philip Torr. A systematic survey of prompt engineering on vision-language foundation models. *arXiv preprint arXiv:2307.12980*, 2023.
- [18] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and XIAOJUAN QI. Is synthetic data from generative models ready for image recognition? In *The Eleventh International Conference on Learning Representations*.
- [19] 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*, pages 2217–2226, 2019.
- [20] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, 2021.
- [21] Min Huang, Chen Yang, and Xiaoyan Yu. Cutcp: Custom text generation-based class-aware prompt tuning for visual-language models. *Scientific Reports*, page 2681, 2025.
- [22] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.
- [23] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022.
- [24] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19113–19122, 2023.
- [25] Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15190–15200, 2023.
- [26] Neehar Kondapaneni, Markus Marks, Manuel Knott, Rogério Guimaraes, and Pietro Perona. Text-image alignment for diffusion-based perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13883–13893, 2024.
- [27] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, pages 554–561, 2013.
- [28] Marc Lafon, Elias Ramzi, Clément Rambour, Nicolas Audebert, and Nicolas Thome. Gallop: Learning global and local prompts for vision-language models. *arXiv preprint arXiv:2407.01400*, 2024.
- [29] Samuel Lavoie, Polina Kirichenko, Mark Ibrahim, Mido Assran, Andrew Gordon Wilson, Aaron Courville, and Nicolas Ballas. Modeling caption diversity in contrastive vision-language pretraining. In *Forty-first International Conference on Machine Learning*.
- [30] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021.
- [31] Juncheng Li, Minghe Gao, Longhui Wei, Siliang Tang, Wenqiao Zhang, Mengze Li, Wei Ji, Qi Tian, Tat-Seng Chua, and Yueting Zhuang. Gradient-regulated meta-prompt learning for generalizable vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2551–2562, 2023.
- [32] Juncheng Li, Xin He, Longhui Wei, Long Qian, Linchao Zhu, Lingxi Xie, Yueting Zhuang, Qi Tian, and Siliang Tang. Fine-grained semantically aligned vision-language pre-training. *Advances in neural information processing systems*, 35:7290–7303, 2022.

- [33] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021.
- [34] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23390–23400, 2023.
- [35] Yilun Li, Miaomiao Cheng, Xu Han, and Wei Song. Divergence-enhanced knowledge-guided context optimization for visual-language prompt tuning. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [36] Zheng Li, Yuxuan Li, Penghai Zhao, Renjie Song, Xiang Li, and Jian Yang. Is synthetic data from diffusion models ready for knowledge distillation? *arXiv preprint arXiv:2305.12954*, 2023.
- [37] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023.
- [38] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *AI Open*, 5:208–215, 2024.
- [39] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.
- [40] Arthur Moreau, Nathan Piasco, Dzmitry Tsishkou, Bogdan Stanciulescu, and Arnaud de La Fortelle. Lens: Localization enhanced by nerf synthesis. In *Conference on Robot Learning*, pages 1347–1356. PMLR, 2022.
- [41] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008.
- [42] Jinyoung Park, Juyeon Ko, and Hyunwoo J Kim. Prompt learning via meta-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26940–26950, 2024.
- [43] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012.
- [44] Duo Peng, Ping Hu, Qihong Ke, and Jun Liu. Diffusion-based image translation with label guidance for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 808–820, 2023.
- [45] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. Learning deep object detectors from 3d models. In *Proceedings of the IEEE international conference on computer vision*, pages 1278–1286, 2015.
- [46] 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*, pages 8748–8763. PMLR, 2021.
- [47] Shuhuai Ren, Aston Zhang, Yi Zhu, Shuai Zhang, Shuai Zheng, Mu Li, Alexander J Smola, and Xu Sun. Prompt pre-training with twenty-thousand classes for open-vocabulary visual recognition. *Advances in Neural Information Processing Systems*, 36:12569–12588, 2023.
- [48] 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*, pages 10684–10695, 2022.
- [49] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.
- [50] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban

- scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016.
- [51] Artem Rozantsev, Vincent Lepetit, and Pascal Fua. On rendering synthetic images for training an object detector. *Computer Vision and Image Understanding*, 137:24–37, 2015.
 - [52] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.
 - [53] Mert Bülent Sarıyıldız, Karteek Alahari, Diane Larlus, and Yannis Kalantidis. Fake it till you make it: Learning transferable representations from synthetic imagenet clones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8011–8021, 2023.
 - [54] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.
 - [55] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.
 - [56] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.
 - [57] Ximeng Sun, Ping Hu, and Kate Saenko. Dualcoop: Fast adaptation to multi-label recognition with limited annotations. *Advances in Neural Information Processing Systems*, 35:30569–30582, 2022.
 - [58] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
 - [59] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
 - [60] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*, 2023.
 - [61] Lifu Wang, Daqing Liu, Xincheng Liu, and Xiaodong He. Scaling down text encoders of text-to-image diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18424–18433, 2025.
 - [62] Longhui Wei, Lingxi Xie, Wengang Zhou, Houqiang Li, and Qi Tian. Mvp: Multimodality-guided visual pre-training. In *European conference on computer vision*, pages 337–353. Springer, 2022.
 - [63] Jianzong Wu, Xiangtai Li, Shilin Xu, Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, et al. Towards open vocabulary learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
 - [64] Weijia Wu, Yuzhong Zhao, Hao Chen, Yuchao Gu, Rui Zhao, Yefei He, Hong Zhou, Mike Zheng Shou, and Chunhua Shen. Datasetdm: Synthesizing data with perception annotations using diffusion models. *Advances in Neural Information Processing Systems*, 36:54683–54695, 2023.
 - [65] Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1206–1217, 2023.
 - [66] Zhenyu Wu, Lin Wang, Wei Wang, Tengfei Shi, Chenglizhao Chen, Aimin Hao, and Shuo Li. Synthetic data supervised salient object detection. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5557–5565, 2022.
 - [67] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010.

- [68] Zehao Xiao, Jiayi Shen, Mohammad Mahdi Derakhshani, Shengcai Liao, and Cees GM Snoek. Any-shift prompting for generalization over distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13849–13860, 2024.
- [69] Lihe Yang, Xiaogang Xu, Bingyi Kang, Yinghuan Shi, and Hengshuang Zhao. Freemask: Synthetic images with dense annotations make stronger segmentation models. *Advances in Neural Information Processing Systems*, 36, 2024.
- [70] Hantao Yao, Rui Zhang, and Changsheng Xu. Tcp: Textual-based class-aware prompt tuning for visual-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23438–23448, 2024.
- [71] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Tsung-Yi Lin, Alberto Rodriguez, and Phillip Isola. Nerf-supervision: Learning dense object descriptors from neural radiance fields. In *2022 international conference on robotics and automation (ICRA)*, pages 6496–6503. IEEE, 2022.
- [72] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*, 2022.
- [73] Ji Zhang, Shihan Wu, Lianli Gao, Heng Tao Shen, and Jingkuan Song. Dept: Decoupled prompt tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12924–12933, 2024.
- [74] Manlin Zhang, Jie Wu, Yuxi Ren, Ming Li, Jie Qin, Xuefeng Xiao, Wei Liu, Rui Wang, Min Zheng, and Andy J Ma. Diffusionengine: Diffusion model is scalable data engine for object detection. *arXiv preprint arXiv:2309.03893*, 2023.
- [75] 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*, pages 16816–16825, 2022.
- [76] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.
- [77] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15659–15669, 2023.
- [78] Zhen Zhu, Yiming Gong, and Derek Hoiem. Anytime continual learning for open vocabulary classification. In *European Conference on Computer Vision*, pages 269–285. Springer, 2025.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The abstract and introduction include the claims made in the paper.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We have discussed the limitations in Section 7.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: Specific details are provided in supplementary materials.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We provide the implementation details in Section 6 and supplementary materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We do not provide open access to the data and code at this time, but can publish part of them at the rebuttal stage if the reviewers need it. The complete data and code will be published after the paper is accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide the implementation details see Section 6 and supplementary materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We follow existing work in the areas we work in and do not provide statistical significance for fair comparisons.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: : We provide the computer resources in supplementary materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Our work conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: There is no societal impact of our work performed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper contains no such risky models or data.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We've cited the original paper of the code and model we used.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [No]

Justification: We will provide open access to part of the new assets at the rebuttal stage if the reviewers need it. The complete assets will be published after the paper is accepted.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The LLM is used only for writing and editing.

Guidelines.

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Proof of Theorem 3

In the manuscript, we present Theorem 3 to demonstrate that the estimation error of $p(\mathbf{x}_u, \mathbf{y}_u)$ is upper bounded. Here we provide further derivations of Theorem 3 of the manuscripts. To this end, We first review the defined setting in open-vocabulary learning task, present some lemmas, and their proofs, which are based on PAC-Bayesian Theorems [?, ?].

Setting. The data in open environments can be denoted as $\mathcal{D}_o = \{\mathcal{D}_s, \mathcal{D}_u\}$, which consists of image-label pairs $\{(\mathbf{x}_o, \mathbf{y}_o)\} = \{(\mathbf{x}_s, \mathbf{y}_s), (\mathbf{x}_u, \mathbf{y}_u)\}$. We assume a predicted unseen-class data distribution \mathbf{E} , where $(\mathbf{x}_e^i, \mathbf{y}_e^i)$ has the probability \mathbf{E}_i . Similarly, the distributions of training data, unseen data and data in open environments are denoted as \mathbf{S} , \mathbf{U} and \mathbf{O} , respectively.

Lemma 1. *With probability at least $1 - \delta$ over the training dataset of size m , we have the following,*

$$\sum \mathbf{S}_i e^{(2m-1)\gamma_i^2} \leq \frac{4m}{\delta}. \quad (12)$$

Proof. By the Chernoff bound we have $P(\gamma \geq x) \leq 2e^{-2mx^2}$. We now consider the density function $f(\gamma)$ maximizing $\int_0^\infty e^{(2m-1)\gamma^2} f(\gamma) d\gamma$ subject to the constraint that $\int_x^\infty f(\gamma) d\gamma \leq 2e^{-2mx^2}$. The maximum occurs when we have $\int_x^\infty f(\gamma) d\gamma = 2e^{-2mx^2}$ which is realized when $f(\gamma) = 8m\gamma e^{-2m\gamma^2}$. So we have the following.

$$\begin{aligned} \mathbb{E}_S e^{(2m-1)\gamma^2} &\leq \int_0^\infty e^{(2m-1)\gamma^2} f(\gamma) d\gamma \\ &= \int_0^\infty 8m\gamma e^{(2m-1)\gamma^2} e^{-2m\gamma^2} d\gamma \\ &= \int_0^\infty 8m\gamma e^{-\gamma^2} d\gamma \\ &= 4m, \end{aligned} \quad (13)$$

which suffices to the following.

$$\forall i \quad \mathbb{E}_S \left[e^{(2m-1)\gamma_i^2} \right] \leq 4m. \quad (14)$$

So we have the following.

$$\mathbb{E}_S \mathbb{E}_i e^{(2m-1)\gamma_i^2} \leq 4m. \quad (15)$$

By applying Markov's inequality on Eq. (15), we get the following.

$$P \left[\sum \mathbf{S}_i e^{(2m-1)\gamma_i^2} \geq \frac{4m}{\delta} \right] \leq \delta, \quad (16)$$

which suffices to lemma 1.

To prove lemma 6 we consider selecting a training data distribution \mathbf{S} and a predicted unseen-class data distribution \mathbf{E} . Lemma 1 implies that with probability at least $1 - \delta$ we have the following.

$$\sum \mathbf{S}_i e^{(2m-1)\gamma_i^2} \leq \frac{4m}{\delta}. \quad (17)$$

So to prove lemma 1 it now suffices to show that Eq. (17) plus $\ln \frac{1}{\delta} \leq 2m$ implies the following for all distributions \mathbf{E} such that $d(\mathbf{E}, \mathbf{S}) \leq 2m$.

$$\sum \mathbf{E}_i \gamma_i \leq \sum \mathbf{E}_i \sqrt{\frac{\ln \frac{\mathbf{E}_i}{\mathbf{S}_i} + \ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m-1}}. \quad (18)$$

To prove Eq. (18) for a given \mathbf{E} we select γ_i so as to maximize the quantity $\sum \mathbf{E}_i \gamma_i$ subject to the constraint Eq. (17). Using Lagrange multipliers we set the gradient of the constraint to be equal to a multiplier λ times the gradient of the objective function.

$$2(2m-1)\gamma_i e^{(2m-1)\gamma_i^2} \mathbf{S}_i = \lambda \mathbf{E}_i. \quad (19)$$

Eq. (18) is trivially true if $\mathbf{E}_i > 0$ but $\mathbf{S}_i = 0$ for some i . So we can assume without loss of generality that $\mathbf{S}_i > 0$ whenever $\mathbf{E}_i > 0$. This allows the above to be rewritten as follows.

$$2(2m-1)\gamma_i e^{(2m-1)\gamma_i^2} = \frac{\lambda \mathbf{E}_i}{\mathbf{S}_i}. \quad (20)$$

Note that $2(2m-1)\gamma e^{(2m-1)\gamma^2}$ is an unbounded monotonically increasing function of γ . We now define $\Delta_i(\lambda)$ to be the unique non-negative value satisfying the following.

$$2(2m-1)\Delta_i(\lambda) e^{(2m-1)\Delta_i^2(\lambda)} = \frac{\lambda \mathbf{E}_i}{\mathbf{S}_i}. \quad (21)$$

Now note that $\sum \mathbf{S}_i e^{(2m-1)\Delta_i^2(\lambda)}$ is an unbounded monotonically increasing function of λ . We now define λ^* to be the unique nonnegative value such that we have the following.

$$\sum \mathbf{S}_i e^{(2m-1)\Delta_i^2(\lambda^*)} = \frac{4m}{\delta}. \quad (22)$$

Note that $\Delta_i(0) = 0$ and $\sum \mathbf{S}_i e^{(2m-1)\Delta_i^2(0)} = 1 < \frac{4m}{\delta}$. So we must have $\lambda^* > 0$ and hence $\Delta_i(\lambda^*) > 0$ for $\mathbf{E}_i > 0$. \square

Lemma 2. For any γ_i satisfying Eq. (17), we have the following.

$$\sum \mathbf{E}_i \gamma_i \leq \mathbf{E}_i \Delta_i(\lambda^*). \quad (23)$$

Proof. Consider the following four situations:

(1) $\exists i \gamma_i < 0$

$\sum \mathbf{E}_i \gamma_i$ can be increased by replacing γ_i with $-\gamma_i$ for $\gamma_i < 0$. Hence we can assume without loss of generality that $\gamma_i \geq 0$.

(2) $\sum \mathbf{S}_i e^{(2m-1)\gamma_i^2} < \frac{4m}{\delta}$

$\sum \mathbf{E}_i \gamma_i$ can be increased by raising γ_i with $\mathbf{E}_i > 0$. Hence we can assume without loss of generality that $\sum \mathbf{S}_i e^{(2m-1)\gamma_i^2} = \frac{4m}{\delta}$.

(3) $\exists i \mathbf{E}_i = 0, \gamma_i > 0$

$\sum \mathbf{E}_i \gamma_i$ can be increased by setting $\gamma_i = 0$ with $\mathbf{E}_i = 0$ while raising γ_i with $\mathbf{E}_i > 0$. Hence we can assume without loss of generality that $\gamma_i = 0$ whenever $\mathbf{E}_i = 0$.

(4) $\exists j, k \mathbf{E}_j > 0, \mathbf{E}_k > 0,$

$$\frac{\mathbf{S}_j}{\mathbf{E}_j} \gamma_j e^{(2m-1)\gamma_j^2} > \frac{\mathbf{S}_k}{\mathbf{E}_k} \gamma_k e^{(2m-1)\gamma_k^2}$$

Eq. (19) is trivially true if $\mathbf{E}_i = 0$ and $\gamma_i = 0$. So we can rewrite Eq. (19) as follows.

$$\lambda_i = \frac{\mathbf{S}_j}{\mathbf{E}_i} 2(2m-1)\gamma_i e^{(2m-1)\gamma_i^2}. \quad (24)$$

From Eq. (24) we can get the following.

$$\lambda_j = \frac{\mathbf{S}_j}{\mathbf{E}_j} 2(2m-1)\gamma_j e^{(2m-1)\gamma_j^2} > \frac{\mathbf{S}_k}{\mathbf{E}_k} 2(2m-1)\gamma_k e^{(2m-1)\gamma_k^2} = \lambda_k. \quad (25)$$

For $\lambda_i > 0$, Eq. (19) can be rewritten as follows.

$$\mathbf{E}_i = \frac{\mathbf{S}_j}{\lambda_i} 2(2m-1)\gamma_i^2 e^{(2m-1)\gamma_i^2}. \quad (26)$$

So we have the following.

$$\sum \mathbf{E}_i \gamma_i = \sum \frac{\mathbf{S}_j}{\lambda_i} 2(2m-1)\gamma_i^2 e^{(2m-1)\gamma_i^2}. \quad (27)$$

Let $f(x_i)$ be the function mapping x_i to $\sum \frac{S_i}{\lambda_i} 2(2m-1)x_i e^{(2m-1)x_i}$.

$$f'(x_i) = 4m(2m-1) \sum \frac{S_i}{\lambda_i} x_i e^{(2m-1)x_i}. \quad (28)$$

For $x_j = x_k$, $f'(x_j) < f'(x_k)$.

$\sum \mathbf{E}_i \gamma_i$ can be increased by increasing γ_k and decreasing γ_j while holding $\sum S_i e^{(2m-1)\gamma_i^2}$ constant. Hence we can assume without loss of generality that there exists a value λ' such that for all indices i with $\mathbf{E}_i > 0$ we have $2(2m-1)\gamma_i e^{(2m-1)\gamma_i^2} = \frac{\lambda' \mathbf{E}_i}{S_i}$, which implies $\gamma_i = \triangle_i(\lambda')$. We also have $\sum S_i e^{(2m-1)\triangle_i^2(\lambda')} = \frac{4m}{\delta}$, which implies $\lambda' = \lambda^*$. So we have $\gamma_i = \triangle_i(\lambda^*)$ which implies the result.

Now proving lemma 1 suffices to bound $\sum \mathbf{E}_i \triangle_i(\lambda^*)$. Eq. (21) implies that for $\lambda \gg 1$ and $\mathbf{E}_i \geq S_i$ we have the following

$$\triangle_i(\lambda) \approx \sqrt{\frac{\ln \frac{\lambda \mathbf{E}_i}{S_i}}{2m-1}}. \quad (29)$$

□

This approximate relationship is made more precise in the following two lemmas.

Lemma 3. For $m \geq 1$, $S_i > 0$, $\mathbf{E}_i \geq S_i$ and $\lambda \geq e$, we have the following.

$$\triangle_i(\lambda) \leq \sqrt{\frac{\ln \frac{\lambda \mathbf{E}_i}{S_i}}{2m-1}}. \quad (30)$$

Proof. Let $g(x)$ be the function mapping x to $2(2m-1)x e^{(2m-1)x^2}$. By definition, $\triangle_i(\lambda)$ satisfies $g(\triangle_i(\lambda)) = \frac{\lambda \mathbf{E}_i}{S_i}$. Note that for $x \geq 0$ we have that $g(x)$ is a monotonically increasing function. Hence for $x \geq 0$ and $g(x) \geq \frac{\lambda \mathbf{E}_i}{S_i}$ we must have $\triangle_i(\lambda) \leq x$. Under the assumptions of lemma 3 we have $\ln \frac{\lambda \mathbf{E}_i}{S_i} \geq 1$ which implies the following.

$$g\left(\sqrt{\frac{\ln \frac{\lambda \mathbf{E}_i}{S_i}}{2m-1}}\right) = 2\sqrt{(2m-1)\ln \frac{\lambda \mathbf{E}_i}{S_i}} \frac{\lambda \mathbf{E}_i}{S_i} \geq \frac{\lambda \mathbf{E}_i}{S_i}. \quad (31)$$

□

Lemma 4. For $m \geq 1$, $S_i > 0$, $\mathbf{E}_i \geq S_i$ and $\lambda \geq e$, we have the following.

$$\triangle_i(\lambda) \geq \sqrt{\frac{\ln \frac{\lambda \mathbf{E}_i}{S_i} + \frac{1}{2} \ln m - \ln \ln \frac{\lambda \mathbf{E}_i}{S_i} - 2}{2m-1}}. \quad (32)$$

Proof. By an argument similar to that in the proof of lemma 3, to show that $\triangle_i(\lambda) \geq x$ it suffices to show that $g(x) \leq \frac{\lambda \mathbf{E}_i}{S_i}$. In particular we have the following.

$$\begin{aligned} & g\left(\sqrt{\frac{\ln \frac{\lambda \mathbf{E}_i}{S_i} + \frac{1}{2} \ln m - \ln \ln \frac{\lambda \mathbf{E}_i}{S_i} - 2}{2m-1}}\right) \\ &= 2\sqrt{(2m-1)\left(\ln \frac{\lambda \mathbf{E}_i}{S_i} + \frac{1}{2} \ln m - \ln \ln \frac{\lambda \mathbf{E}_i}{S_i} - 2\right)} \\ & \quad \cdot \frac{\lambda \mathbf{E}_i}{S_i} \frac{1}{\sqrt{m} \ln \frac{\lambda \mathbf{E}_i}{S_i} e^2} \\ &\leq 4\sqrt{m \ln \frac{\lambda \mathbf{E}_i}{S_i} \frac{\lambda \mathbf{E}_i}{S_i}} \frac{1}{\sqrt{m} \ln \frac{\lambda \mathbf{E}_i}{S_i} e^2} \\ &= \frac{\lambda \mathbf{E}_i}{S_i} \frac{1}{\sqrt{\ln \frac{\lambda \mathbf{E}_i}{S_i}}} \frac{4}{e^2} \\ &\leq \frac{\lambda \mathbf{E}_i}{S_i}. \end{aligned} \quad (33)$$

□

Lemma 5. For $d(\mathbf{E}, \mathbf{S}) \leq 2m$ and $\ln \frac{1}{\delta} \leq 2m$, we have $\lambda^* \leq \frac{64e^2m^{5/2}}{\delta}$.

Proof. Let $h(x)$ be the function mapping x to $\sum \mathbf{S}_i e^{(2m-1)\Delta_i^2(x)}$. The quantity λ^* is defined by $h(\lambda^*) = \frac{4m}{\delta}$. Since h is a monotonically increasing function, $h(x) \geq \frac{4m}{\delta}$ implies $\lambda^* \leq x$. Lemma 4 implies that for $x \geq e$ we have the following.

$$\begin{aligned}
h(x) &= \sum \mathbf{S}_i e^{(2m-1)\Delta_i^2(x)} \\
&\geq \sum \mathbf{S}_i \frac{x \mathbf{E}_i}{\mathbf{S}_i} \frac{1}{\sqrt{m} \ln \frac{x \mathbf{E}_i}{\mathbf{S}_i} e^2} \\
&= \frac{x}{e^2 \sqrt{m}} \sum \frac{\mathbf{E}_i}{\ln \frac{x \mathbf{E}_i}{\mathbf{S}_i}} \\
&\geq \frac{x}{e^2 \sqrt{m}} \frac{1}{\sum \mathbf{E}_i \ln \frac{x \mathbf{E}_i}{\mathbf{S}_i}} \\
&= \frac{x}{e^2 \sqrt{m} (d(\mathbf{E}, \mathbf{B}) + \ln x)} \\
&\geq \frac{x}{e^2 \sqrt{m} (2m + \ln x)}.
\end{aligned} \tag{34}$$

Now inserting $x = \frac{64e^2m^{5/2}}{\delta}$ we get the following.

$$\begin{aligned}
h\left(\frac{64e^2m^{5/2}}{\delta}\right) &\geq \frac{64m^2}{\delta(2m + \frac{5}{2} \ln m + \ln \frac{1}{\delta} + 8)} \\
&\geq \frac{64m^2}{\delta(2m + \frac{5}{2}m + 2m + 8m)} \\
&\geq \frac{4m}{\delta}.
\end{aligned} \tag{35}$$

□

Lemma 6. Without loss of generality, we define the distance between distributions as $d(\mathbf{P}, \mathbf{Q}) = \sum \mathbf{P}_i \ln \frac{\mathbf{P}_i}{\mathbf{Q}_i}$ for analysis. For $\ln \frac{1}{\delta} \leq 2m$ we have that with probability $1 - \delta$ over the training dataset of size m the following holds for all distributions satisfying $d(\mathbf{E}, \mathbf{S}) \leq 2m$.

$$d(\mathbf{E}, \mathbf{O}) - d(\mathbf{E}, \mathbf{S}) \leq \sum \mathbf{E}_i \sqrt{\frac{\ln \frac{\mathbf{E}_i}{\mathbf{S}_i} + \ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m - 1}}. \tag{36}$$

Proof. Note that $d(\mathbf{E}, \mathbf{O}) - d(\mathbf{E}, \mathbf{S}) = \sum \mathbf{E}_i (\ln \frac{\mathbf{E}_i}{\mathbf{O}_i} - \ln \frac{\mathbf{E}_i}{\mathbf{S}_i}) \leq \sum \mathbf{E}_i \gamma_i$, where γ_i abbreviates $|\ln \frac{\mathbf{E}_i}{\mathbf{O}_i} - \ln \frac{\mathbf{E}_i}{\mathbf{S}_i}|$, the lemma can be viewed as an upper bound of $\sum \mathbf{E}_i \gamma_i$.

From the above-mentioned lemmas, we have that

$$\begin{aligned}
d(\mathbf{E}, \mathbf{O}) - d(\mathbf{E}, \mathbf{S}) &\leq \sum \mathbf{E}_i \gamma_i \\
&\leq \sum \mathbf{E}_i \Delta_i(\lambda^*) \\
&\leq \sum \mathbf{E}_i \Delta_i\left(\frac{64e^2m^{5/2}}{\delta}\right) \\
&\leq \sum \mathbf{E}_i \sqrt{\frac{\ln \frac{64e^2m^{5/2}\mathbf{E}_i}{\delta \mathbf{S}_i}}{2m - 1}} \\
&\leq \sum \mathbf{E}_i \sqrt{\frac{\ln \frac{\mathbf{E}_i}{\mathbf{S}_i} + \ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m - 1}}.
\end{aligned} \tag{37}$$

□

Lemma 7. Denote the $d(\cdot, \cdot)$ as the distribution distance. With probability at least $1 - \delta$, we have the following,

$$\begin{aligned} d(p(\mathbf{x}_o, \mathbf{y}_o), p(\mathbf{x}_e, \mathbf{y}_e)) &\leq d(p(\mathbf{x}_e, \mathbf{y}_e), p(\mathbf{x}_s, \mathbf{y}_s)) + \sqrt{\frac{d(p(\mathbf{x}_e, \mathbf{y}_e), p(\mathbf{x}_s, \mathbf{y}_s))}{2m-1}} \\ &\quad + \sqrt{\frac{\ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m-1}}, \end{aligned} \quad (38)$$

where m denotes the size of training dataset.

Proof. By applying Jensen's inequality on lemma 6, we can get Theorem 3.

$$\begin{aligned} d(\mathbf{E}, \mathbf{O}) &\leq d(\mathbf{E}, \mathbf{S}) + \sum \mathbf{E}_i \sqrt{\frac{\ln \frac{\mathbf{E}_i}{\mathbf{S}_i} + \ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m-1}} \\ &\leq d(\mathbf{E}, \mathbf{S}) + \sqrt{\frac{d(\mathbf{E}, \mathbf{S}) + \ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m-1}} \\ &\leq d(\mathbf{E}, \mathbf{S}) + \sqrt{\frac{d(\mathbf{E}, \mathbf{S})}{2m-1}} + \sqrt{\frac{\ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m-1}}. \end{aligned} \quad (39)$$

□

From Lemma 7, we can obviously observe that the distribution distance of the generated unseen-class data and the open-environment has an upper bound, which indicates that the rationality of generating unseen-class data for distribution estimation in open environments. Obviously, we also can observe that this upper bound is strongly related to the distribution distance between the generated unseen-class data and the seen-class data. The conclusion of Lemma 7 is same with Theorem 3. This also motivates us to construct the proposed open-vocabulary method. From Lemma 7, we can directly obtain Theorem 3.

Theorem 3. Denote the $d(\cdot, \cdot)$ as the distribution distance. With probability at least $1 - \delta$, we have the following,

$$\begin{aligned} d(p(\mathbf{x}_u, \mathbf{y}_u), p(\mathbf{x}_e, \mathbf{y}_e)) &\leq \sqrt{\frac{d(p(\mathbf{x}_e, \mathbf{y}_e), p(\mathbf{x}_s, \mathbf{y}_s))}{2m-1}} \\ &\quad + \sqrt{\frac{\ln \frac{1}{\delta} + \frac{5}{2} \ln m + 8}{2m-1}}, \end{aligned} \quad (40)$$

where m denotes the size of training dataset.

B Proof of Theorem 4

In the manuscripts, we present Theorem 4 to demonstrate that the estimation error of $p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)$ is upper bounded. Here we provide specific derivations of Theorem 4.

Theorem 4. Given the predicted classes $\mathbf{Y}_e = \{\mathbf{y}_e\}$. Suppose that predicted class \mathbf{Y}_e have any nonzero probability $p(\mathbf{Y}_e)$. With probability at least $1 - \delta$ over the m instances of generated unseen-class data $\{(\mathbf{x}_e, \mathbf{y}_e)\}$, we have that

$$D_{KL}(p(\bar{\mathbf{y}}|\mathbf{x}_e, \Phi) || p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)) \leq \sqrt{\frac{\ln \frac{1}{p(\mathbf{Y}_e)} + \ln \frac{1}{\delta}}{2m}} - \ln \frac{|\mathbf{Y}_e|}{|\mathbf{Y}_u|}. \quad (41)$$

Proof. We denote $p^*(\mathbf{y}) = p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)$, $\bar{\mathbf{y}} \in \mathbf{Y}_u$. For analysis, we define $P(\mathbf{Y}_e) = \sum_{\mathbf{y} \in \mathbf{Y}_e} p^*(\mathbf{y})$, and we define the conditional distribution as $P_e(\mathbf{y}) = p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi) = \frac{p^*(\mathbf{y})}{P(\mathbf{Y}_e)}$, $\bar{\mathbf{y}} \in \mathbf{Y}_e$. $P_e(\mathbf{y})$ satisfies

that $\sum_{\mathbf{y} \in \mathbf{Y}_e} P_e(\mathbf{y}) = 1$. We also denote that $\hat{p}(\mathbf{y}) = p(\bar{\mathbf{y}}|\mathbf{x}_e, \Phi)$. In this way, the KL divergence between $\hat{p}(\mathbf{y})$ and $p^*(\mathbf{y})$ is computed as

$$D_{\text{KL}}(\hat{p}(\mathbf{y})||p^*(\mathbf{y})) = \sum_{\mathbf{y} \in \mathbf{Y}_e} \hat{p}(\mathbf{y}) \ln \frac{\hat{p}(\mathbf{y})}{p^*(\mathbf{y})}. \quad (42)$$

$\frac{\hat{p}(\mathbf{y})}{p^*(\mathbf{y})}$ can be formulated as

$$\frac{\hat{p}(\mathbf{y})}{p^*(\mathbf{y})} = \frac{\hat{p}(\mathbf{y})}{P_e(\mathbf{y})} \frac{P_e(\mathbf{y})}{p^*(\mathbf{y})} = \frac{\hat{p}(\mathbf{y})}{P_e(\mathbf{y})} \frac{1}{P(\mathbf{Y}_e)}, \quad (43)$$

and thus

$$\ln \frac{\hat{p}(\mathbf{y})}{p^*(\mathbf{y})} = \ln \frac{\hat{p}(\mathbf{y})}{P_e(\mathbf{y})} - \ln P(\mathbf{Y}_e). \quad (44)$$

By substituting Eq. (44) into Eq (42), we have that

$$\begin{aligned} D_{\text{KL}}(\hat{p}(\mathbf{y})||p^*(\mathbf{y})) &= \sum_{\mathbf{y} \in \mathbf{Y}_e} \hat{p}(\mathbf{y}) \left(\ln \frac{\hat{p}(\mathbf{y})}{P_e(\mathbf{y})} - \ln P(\mathbf{Y}_e) \right) \\ &= \sum_{\mathbf{y} \in \mathbf{Y}_e} \hat{p}(\mathbf{y}) \ln \frac{\hat{p}(\mathbf{y})}{P_e(\mathbf{y})} - \sum_{\mathbf{y} \in \mathbf{Y}_e} \hat{p}(\mathbf{y}) \ln P(\mathbf{Y}_e) \\ &= D_{\text{KL}}(\hat{p}(\mathbf{y})||P_e(\mathbf{y})) - \ln P(\mathbf{Y}_e). \end{aligned} \quad (45)$$

By the Chernoff bound, for the generated unseen-class data $\{(\mathbf{x}_e, \mathbf{y}_e)\}$, we have that

$$P[D_{\text{KL}}(\hat{p}(\mathbf{y})||P_e(\mathbf{y})) \geq \delta] \leq e^{-2mt^2}. \quad (46)$$

From the union bound, and the classes are countable, we have that

$$P[\exists \mathbf{Y}_e : D_{\text{KL}}(\hat{p}(\mathbf{y})||P_e(\mathbf{y})) \geq t] \leq \sum_{\mathbf{Y}_e} P[D_{\text{KL}}(\hat{p}(\mathbf{y})||P_e(\mathbf{y})) \geq t]. \quad (47)$$

Assign the probability $p(\mathbf{Y}_e)$ for \mathbf{Y}_e , and thus $\sum_{\mathbf{Y}_e} p(\mathbf{Y}_e) = 1$. In this way, we have that

$$P[\exists \mathbf{Y}_e : D_{\text{KL}}(\hat{p}(\mathbf{y})||P_e(\mathbf{y})) \geq \delta] \leq \sum_{\mathbf{Y}_e} e^{-2mt^2} = \sum_{\mathbf{Y}_e} p(\mathbf{Y}_e) \frac{e^{-2mt^2}}{p(\mathbf{Y}_e)}. \quad (48)$$

Letting $e^{-2mt^2} = p(\mathbf{Y}_e)\delta$, we have that

$$P[\exists \mathbf{Y}_e : D_{\text{KL}}(\hat{p}(\mathbf{y})||P_e(\mathbf{y})) \geq t] \leq \sum_{\mathbf{Y}_e} p(\mathbf{Y}_e)\delta = \delta, \quad (49)$$

where t is computed as

$$t = \sqrt{\frac{\ln \frac{1}{p(\mathbf{y}_e^i)} + \ln \frac{1}{\delta}}{2m}}. \quad (50)$$

In this way, we can derive that

$$P \left[D_{\text{KL}}(\hat{p}(\mathbf{y})||P_e(\mathbf{y})) \geq \sqrt{\frac{\ln \frac{1}{p(\mathbf{y}_e^i)} + \ln \frac{1}{\delta}}{2m}} \right] \leq \delta, \quad (51)$$

Therefore, with probability at least $1 - \delta$, we have that

$$D_{\text{KL}}(\hat{p}(\mathbf{y})||P_e(\mathbf{y})) \leq \sqrt{\frac{\ln \frac{1}{p(\mathbf{y}_e^i)} + \ln \frac{1}{\delta}}{2m}}. \quad (52)$$

We substitute Eq. (52) into Eq. (45). With probability at least $1 - \delta$, we have that

$$D_{\text{KL}}(p(\bar{\mathbf{y}}|\mathbf{x}_e, \Phi)||p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)) \leq \sqrt{\frac{\ln \frac{1}{p(\mathbf{Y}_e)} + \ln \frac{1}{\delta}}{2m}} - \ln \frac{|\mathbf{Y}_e|}{|\mathbf{Y}_u|}. \quad (53)$$

□

C Proof of ELBO in Eq. (7) in the Manuscripts

Proposition 1. *The Evidence Lower Bound (ELBO) of the logarithmic posterior probability can be derived as*

$$\log p(\bar{\mathbf{y}}|\mathbf{x}_o, \Phi) \geq \mathbb{E}[\log p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)] - D_{\text{KL}}(p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi) || p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)), \quad (54)$$

Proof. The probability $p(\bar{\mathbf{y}}|\mathbf{x}_o)$ can be modeled as a function $f(\mathbf{y})$. Therefore, the logarithmic probability $\log p(\bar{\mathbf{y}}|\mathbf{x}_o)$ is computed as

$$\log p(\bar{\mathbf{y}}|\mathbf{x}_o) = \log f(\mathbf{y}) = \log \left(\int f(\tilde{\mathbf{y}}) \delta(\tilde{\mathbf{y}} - \mathbf{y}) d\tilde{\mathbf{y}} \right), \quad (55)$$

where $\delta(\cdot)$ is a Dirac function, and $\tilde{\mathbf{y}}$ is a intermediate variable. The last equality holds since the proposition of the Dirac function. We introduce a variational distribution $q(\tilde{\mathbf{y}})$. Then, $\log f(\mathbf{y})$ holds that

$$\begin{aligned} \log f(\mathbf{y}) &= \log \left(\int f(\tilde{\mathbf{y}}) \delta(\tilde{\mathbf{y}} - \mathbf{y}) d\tilde{\mathbf{y}} \right) = \log \left(\int f(\tilde{\mathbf{y}}) \frac{q(\tilde{\mathbf{y}})}{q(\tilde{\mathbf{y}})} \delta(\tilde{\mathbf{y}} - \mathbf{y}) d\tilde{\mathbf{y}} \right) \\ &= \log \left(\int q(\tilde{\mathbf{y}}) \frac{f(\tilde{\mathbf{y}}) \delta(\tilde{\mathbf{y}} - \mathbf{y})}{q(\tilde{\mathbf{y}})} d\tilde{\mathbf{y}} \right). \end{aligned} \quad (56)$$

From the Jensen inequality, we can derive that

$$\log f(\mathbf{y}) \geq \int q(\tilde{\mathbf{y}}) \log \frac{f(\tilde{\mathbf{y}}) \delta(\tilde{\mathbf{y}} - \mathbf{y})}{q(\tilde{\mathbf{y}})} d\tilde{\mathbf{y}} = \mathbb{E}_{q(\mathbf{y})} \left[\log \frac{f(\mathbf{y}) \delta(\tilde{\mathbf{y}} - \mathbf{y})}{q(\tilde{\mathbf{y}})} \right]. \quad (57)$$

To guarantee the boundedness of the ELBO, we ignore the Dirac function. Eq. (57) can be further modeled as

$$\log f(\mathbf{y}) \geq \mathbb{E}_{q(\mathbf{y})} [\log f(\mathbf{y})] - \mathbb{E}_{q(\mathbf{y})} [\log q(\mathbf{y})]. \quad (58)$$

Then, substituting $f(\mathbf{y}) = \log p(\bar{\mathbf{y}}|\mathbf{x}_o, \Phi)$ and $q(\mathbf{y}) = p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)$, we have that

$$\begin{aligned} \log p(\bar{\mathbf{y}}|\mathbf{x}_o, \Phi) &\geq \mathbb{E}_{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} [\log p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)] + \mathbb{E}_{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} [\log p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)] \\ &\quad - \mathbb{E}_{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} [\log p(\bar{\mathbf{y}}|\Phi)] - \mathbb{E}_{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} [\log p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)] \\ &= \mathbb{E}_{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} [\log p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)] + \mathbb{E}_{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} \left[\log \frac{p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)}{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} \right] \\ &\quad - \mathbb{E}_{p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)} [\log p(\bar{\mathbf{y}}|\Phi)]. \end{aligned} \quad (59)$$

The probability term $p(\bar{\mathbf{y}}|\Phi)$ can be ignored because it appears as a constant term in the variational lower bound (ELBO). It depends only on the model parameters Φ and is independent of the input data \mathbf{x} . As such, it does not affect the gradient computation or the update of model parameters Φ . Since this term does not contribute to the optimization process, it can be safely omitted, simplifying the derivation. In many works, derivation of ELBO commonly omit such constant terms, as they do not affect the optimization objective and can be safely ignored to simplify the computation [?, ?, ?]. Therefore, we model Eq. (59) as

$$\log p(\bar{\mathbf{y}}|\mathbf{x}_o, \Phi) \geq \mathbb{E} [\log p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi)] - D_{\text{KL}}(p(\bar{\mathbf{y}}|\mathbf{x}_s, \Phi) || p(\bar{\mathbf{y}}|\mathbf{x}_u, \Phi)). \quad (60)$$

□

D Details and Illustration of Class-Domain-Wise Data Generation Pipeline

D.1 Specific Details of Hierarchy-Guided Unseen Class Predictor

The hierarchy-guided unseen class predictor identifies the potential unseen classes, which are close to training classes. This is achieved by constructing a hierarchical semantic tree, where leaf nodes represent training classes and parent nodes represent their superclasses. The tree is expanded by adding leaf nodes of the candidate unseen classes sourced from WordNet or LLMs. As illustrated in Figure 1, given the training classes ‘‘Goldfish,’’ ‘‘Tench,’’ and ‘‘Ray,’’ a hierarchical semantic tree is constructed where these classes are set as leaf nodes and their superclass ‘‘Fish’’ is set as a parent node. After LLMs are queried, ‘‘Salmon’’ is added as a leaf node of the candidate unseen class under

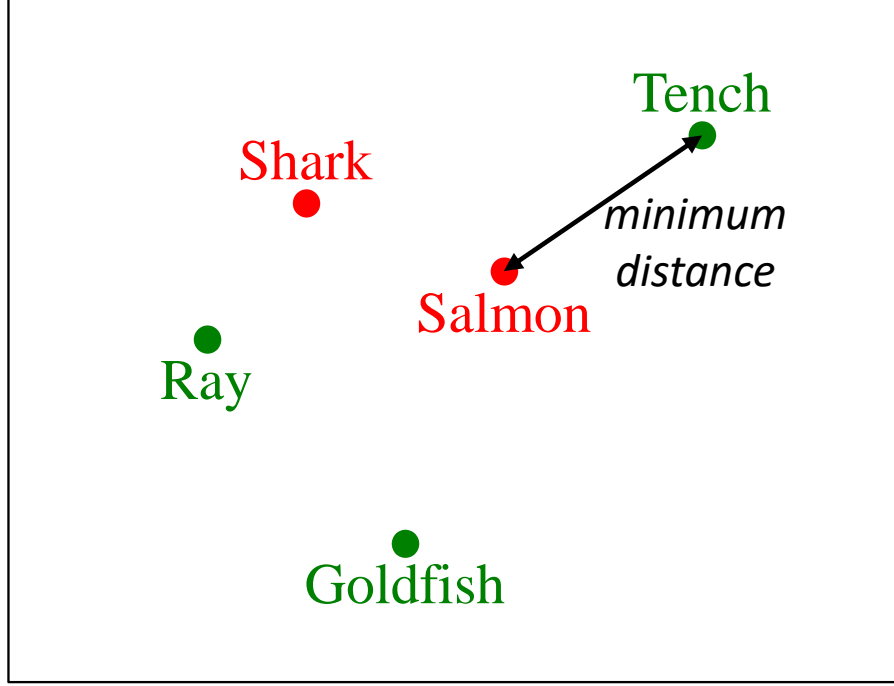


Figure 3: Distance between Classes

the “Fish” superclass. To select the closest candidate unseen class, the predictor computes the cosine similarity between the textual embeddings of candidate unseen classes and given training classes from the text encoder of a pretrained CLIP. The top K closest candidates are chosen as the identified potential unseen classes. Take the classes shown in Figure 3 as an example, the cosine similarity between the textual embeddings of “Tench” and candidate unseen classes “Salmon”, “Shark” is computed. The candidate unseen class with the highest similarity “Salmon” is chosen as a predicted unseen class.

D.2 Specific Details of Caption-Based Domain Information Generator

The caption-based domain information generator extracts contextual attributes, such as styles and backgrounds, from the training data to ensure the generated unseen data align with the visual characteristics of training data. This is achieved by generating class-specific captions for each training class using VLMs. The generator then computes the similarity between these captions and the corresponding data, selecting the top K_1 captions with the highest similarity to mitigate hallucination issues. These selected captions are further summarized into top K_2 class-specific domain information using LLMs. Finally, the predicted unseen classes and the summarized domain information are combined into textual prompts, which guide the data generation process using a text-to-image model such as Stable Diffusion. In Figure 1, for instance, captions for training images of “Tench” are first generated using VLMs. The generator then calculates the similarity between these captions and the corresponding images of “Tench”. The top 3 captions with the highest similarity are selected, which describe the domain information such as “helded by man”, “golden-colored” and “prominent eye”. These captions are summarized into top 1 class-specific domain information using LLMs, which is presented in the caption template in the red box. Finally, the predicted unseen class “Salmon” is inserted into the template to create an image caption, which is then used as input for Stable Diffusion to generate images of “Salmon”.

D.3 Prompt Templates

In the manuscripts, we propose to utilize LLMs and VLMs to identify potential unseen classes and extract domain information of training data. Here we provide the utilized 3 prompt templates.

As to the unseen class predictor, we aim to query the Hypernym of training classes by the following prompt. We first construct the in-context examples for accurate results, as shown in Template 1.

Template 1. *{class} denotes the training class.*

Q: What is the Hypernym category of {class1}?

A: {class2} is the Hypernym category of {class1}.

The prompt is given in Template 2.

Template 2. *{class} denotes the training class.*

Q: What is the Hypernym category of {class3}?

Then, with the generated Hypernym, we leverage LLMs to identify potential unseen classes using LLMs, where the in-context examples and prompts are shown in template 3 and template 4, respectively.

Template 3. *{class} denotes the training class.*

Q: What is the Hyponym category of {class1}?

A: {class2} is the Hyponym category of {class1}.

Template 4. *{class} denotes the training class.*

Q: What is the Hyponym category of {class3}?

We leverage template 5 to generate class-specific captions for each training class using VLMs.

Template 5. *{class} denotes the training class.*

user prompt:

This is an image of {class}. Summarize the main style, scene, and key elements of this image in one sentence.

We leverage template 6 to summarize captions into class-specific domain information using LLMs

Template 6. *{class} denotes the predicted unseen class.*

system prompt:

As a caption summarizer, your task is to transform the provided captions from their original category to a new specified category and condense them into a concise set of 3 distinct one-sentence captions. Make sure the new captions maintain coherence with the original style but reflect the characteristics of the new target category. Each caption must capture a unique artistic style or visual theme. Only generate the transformed one-sentence captions—no introductions, explanations, or comments. The output should strictly follow this format:

1. [Caption 1]

2. [Caption 2]

3. [Caption 3]

user prompt:

Transform and condense the following captions into 3 new one-sentence captions describing {class}, each focusing on a distinct artistic style or visual theme.

E Distribution Alignment

In this paper, we adopt prompt learning method to optimize the pretrained model. We propose a distribution alignment algorithm which aligns the output distributions of model on seen-class data and generated unseen-class data to maximize the logarithmic posterior probability in open environments. The proposed algorithm is summarized in Algorithm 1.

F Experiment Details in Manuscripts

In this section, we provide more specific details of experiments in the manuscripts. Specifically, we present the specific implementation details, details and extra analysis in ablation studies of the manuscripts.

Algorithm 1 Distribution Alignment Algorithm

Input: Parameters Φ , Data $\mathcal{D}_s, \mathcal{G}_u, \mathcal{G}_s$, Epoch E , batch-size B

Output: Optimized Prompts v_{max_iter}

Initialize: $e = 0, v \leftarrow v_0, S \leftarrow \{\}$

```
1: while  $e \leq E$  do
2:   for  $i = 1, 2, \dots, |\mathcal{D}_s|/B$  do
3:     Compute the posterior probability of model output in the current batch of the seen-class
       dataset  $\mathcal{D}_s^i$ .
4:     Accumulate the output posterior probability for distribution alignment into  $S$ .
5:     if  $i \% 8 == 0$  then
6:        $S_{KL} = \{\}$ .
7:       for  $j = 1, 2, \dots, |\mathcal{G}_u|/B$  do
8:         Compute the KL divergence between the accumulated posterior probability on
           the seen-class data and the mini-batch of generated unseen-class data  $d_{kl} =$ 
            $D_{KL}[p(\bar{y}|\mathbf{x}_s, \Phi, v) || p(\bar{y}|\mathbf{x}_e, \Phi, v)]$ .
9:         Update the set as  $S_{KL}.append(d_{kl})$ .
10:      end for
11:      Compute  $\mathcal{L}_{KL}$  based on top  $K_3$  smallest in the set  $S_{KL}$  as  $\mathcal{L}_{KL} = \frac{1}{K_3} \sum_{top K_3} d_{kl}$ .
12:       $S_{mmd} = \{\}$ .
13:      for  $m = 1, 2, \dots, |\mathcal{G}_s|/B$  do
14:        Compute MMD loss  $l_{mmd}$  based on the generated unseen-class data and the seen-class
          data on the current batch, and save them into  $S_{mmd}$ .
15:      end for
16:      Compute  $\mathcal{L}_{MMD}$  based on top  $K_3$  smallest  $l_{mmd}$  as  $\mathcal{L}_{MMD} = \frac{1}{K_3} \sum_{top K_3} l_{mmd}$ .
17:      Compute total loss  $\mathcal{L}_{total} = \mathcal{L}_{CE} + \alpha \mathcal{L}_{KL} + \beta \mathcal{L}_{MMD}$ .
18:      Backward and update the prompt  $v$  using  $\mathcal{L}_{CE}$  and  $\mathcal{L}_{total}$ .
19:      Clear saved data  $S = \{\}$ .
20:    else
21:      Compute  $\mathcal{L}_{CE}$  on the mini-batch of seen-class dataset  $\mathcal{D}_s^i$ .
22:      Backward and update the prompt  $v$  using  $\mathcal{L}_{CE}$ .
23:    end if
24:  end for
25: end while
26: return The updated prompts  $v$ .
```

F.1 Implementation details

For base-to-base/base-to-new generalization, we train each model for 20 epochs using 4 token prompts in the first 9 transformer layers on both visual and text branch. For cross-dataset evaluation, we train the source model for 4 epochs using 4 prompts in the first 3 transformer layers on both visual and text branch. Prompts are randomly initialized with a normal distribution except the text prompts of the first layer which are initialized with the word embeddings of “a photo of a”. The SGD optimizer is adopted, and the learning rate is set as 0.0025. Hyperparameters for the class-domain-wise data generation pipeline and distribution alignment are determined empirically. Specifically, We set $\alpha = 1$, $\beta = 1$, K_0 as 1, K_1 as 8, K_2 as 3 and K_3 as 1. The corresponding hyperparameters are fixed across all datasets and benchmarks.

For LLMs and VLMs, we use Doubao-pro-128k to identifies the potential unseen classes, use LLaVA-v1.6-Vicuna-13B [37] to generate class-specific captions for each training class, use Llama-v3.1-Instruct-8B [59] to summarize captions into class-specific domain information, and use Stable Diffusion v2.1 [48] as the text-to-image model to generate unseen-class data.

Experiments are performed on an NVIDIA A40 GPU, with at most 18 hours 20 GPU memory required to complete training across 11 datasets.

Table 5: Ablation study on sparse loss computation strategy. “w/o spa” denotes distribution alignment algorithm without sparse loss computation strategy.

	Caltech		Pets		Cars		Flowers		Food		Aircraft		DTD		EuroSAT		UCF	
	w/o spa	Ours	w/o spa	Ours	w/o spa	Ours	w/o spa	Ours	w/o spa	Ours	w/o spa	Ours	w/o spa	Ours	w/o spa	Ours	w/o spa	Ours
Base	98.52	98.97	93.36	96.01	81.88	82.93	96.68	98.77	91.14	91.39	46.40	48.98	82.41	85.53	95.95	97.17	87.07	89.14
New	94.21	95.85	91.33	97.65	79.48	80.81	78.09	80.92	92.56	92.99	39.71	44.03	61.72	71.50	76.92	87.90	79.61	82.53
H	96.32	97.38	92.33	96.82	80.66	81.86	86.39	88.96	91.84	92.18	42.80	46.37	70.58	77.89	85.39	92.30	83.17	85.71

F.2 Details and Extra Analysis of Ablation Study

In this subsection, we present the details in ablation studies in hierarchy-guided unseen class predictor and caption-based domain information generator. Then, we present the extra analysis in ablation studies of the manuscripts.

F.2.1 Details of Hierarchy-Guided Unseen Class Predictor

To evaluate the effectiveness of hierarchy-guided unseen class predictor, we first investigate how the quantity of predicted unseen classes impacts the model performance in open environments. Specifically, we introduce a class sampling ratio s_{cls} to control the quantity of predicted unseen classes used for training. Assume that we initially generate N unseen classes for a given dataset, with each class containing M images, we randomly select $s_{cls} \times N$ classes from the predicted N unseen classes and then train the model using $s_{cls} \times N \times M$ images of these classes, discarding the remaining classes and their images.

Next, we investigate how the quality of predicted unseen classes impacts the model performance in open environments. “Low Similarity” denotes that when predicting unseen classes, we compute the cosine similarity to textual seen classes for each candidate class and choose the one with the lowest cosine similarity as the predicted unseen class. “w/oTree” denotes that instead of constructing a hierarchical semantic tree to predict unseen classes, we directly ask LLMs to provide a predicted unseen class corresponding to the given base classes.

F.2.2 Details of Caption-Based Domain Information Generator

To evaluate the effectiveness of caption-based domain information generator, we first investigate how the quantity of generated unseen-class images impacts the model performance in open environments. Similarly, we introduce a image sampling ratio s_{img} to control the number of generated images of predicted unseen classes used for training. For a given dataset, we initially predict N unseen classes with each class containing M generated images. Based on the image sampling ratio s_{img} , we randomly select $s_{img} \times M$ images from each predicted unseen class. The selected $N \times s_{img} \times M$ images from N unseen classes are then used for training, while the remaining images are discarded.

Next, we investigate how the quality of generated images from predicted unseen classes impacts the model performance in open environments. We modify the prompts used in our unseen image generator to control the quality of image generation. In this work, we use the prompts “A picture of a category”, “A photo of a category”, and “An image of a category” as templates for the Stable Diffusion model when generating images of unseen classes, respectively.

F.3 Extra Analysis

The ablation studies provide a comprehensive analysis of the relationship between distribution distance and accuracy, which aligns with the theoretical analysis. Recall that Theorem 3 shows that the estimation error between the unseen-class data distribution and the generated unseen-class data distribution is upper-bounded, and reducing the distribution gap between generated unseen-class data and seen-class data tightens this bound, thereby improving model performance in open environments. Experimental results from the ablation studies validate this theoretical claim. Specifically, as the distribution distance decreases, accuracy consistently improves across various datasets, particularly for new classes. This confirms that reducing the distribution gap between generated unseen-class and seen-class data leads to more accurate estimation of unseen-class data distribution, enhancing the model’s generalization ability in open-vocabulary learning tasks.

G Robust Analysis

G.1 Hyperparameter Analysis

We add experiments to analyze the sensitivity of hyperparameters K_0, K_1, K_2, K_3 in the data generation pipeline and α, β in the loss function. Specifically, we conduct experiments on the EuroSAT dataset by adjusting these hyperparameters. Results on K_0, K_1, K_2, K_3 are shown in Table 6 7 8 9, respectively. Results of α, β are presented in Table 10. Results show that the performance remains relatively stable with varying hyperparameter values, indicating that the method is only minimally sensitive to hyperparameter variation.

Table 6: K_0 Analysis

	$K_0 = 1$	$K_0 = 5$	$K_0 = 10$	$K_0 = 20$
Base	97.17	96.64	96.43	96.24
New	87.90	87.07	85.85	85.84
H	92.30	91.61	90.83	90.57

Table 7: K_1 Analysis

	$K_1 = 8$	$K_1 = 6$	$K_1 = 4$	$K_1 = 2$
Base	97.17	97.02	96.95	96.83
New	87.90	86.31	86.0	85.85
H	92.30	91.35	91.15	91.01

Table 8: K_2 Analysis

	$K_2 = 3$	$K_2 = 2$	$K_2 = 1$
Base	97.17	96.92	96.79
New	87.90	86.59	85.31
H	92.30	91.47	90.69

Table 9: K_3 Analysis

	$K_3 = 1$	$K_3 = 2$	$K_3 = 3$
Base	97.17	96.33	96.60
New	87.90	85.36	86.31
H	92.30	90.51	91.16

Table 10: α/β analysis ($\alpha + \beta = 1$)

α	β	Base	New	H
0.3	0.7	96.88	86.15	91.20
0.4	0.6	96.86	86.26	91.25
0.5	0.5	97.17	87.90	92.30
0.6	0.4	96.81	87.06	91.68

G.2 Robustness to Long-tailed Setting and Noise

Robustness to long-tailed setting. We construct a long-tail distribution setting by removing a portion of samples from selected classes and conduct comparative experiments (with the number of samples per class being 16, 16, 16, 10, and 2, respectively). In this setting, we evaluate the performance of our current top alignment strategy against a sample duplication strategy designed for long-tail settings. The results in Table 11 demonstrate that in the long-tailed setting, the sample duplication strategy can improve performance.

Table 11: Experiments on long-tailed benchmark

	Our method (with top K_g alignment)	Our method (with top K_g alignment + sample duplication)
Base	95.79	97.00
New	85.39	87.39
H	90.29	91.94

Table 12: Robustness evaluation to the noisy semantic tree

	Correct Superclasses	Wrong Superclasses	PromptSRC (Baseline)
Base	97.17	96.17	92.90
New	87.90	86.49	73.90
H	92.30	91.07	82.32

Robustness to noise. To verify the robustness, we conduct an experiment where we select incorrect superclasses on the EuroSAT dataset. Results in Table 12 show that the performance slightly degrades, demonstrating the robustness of our method.

H Extra Ablation Studies

In this section, we present extra ablation studies for validating the effectiveness of sparse loss computation strategy in distribution alignment algorithm. We demonstrate the effectiveness of the sparse loss computation strategy by conducting experiments on the distribution alignment algorithm without it, denoted as 'w/o spa'. The results, shown in Table 13, reveal that the sparse loss computation strategy significantly improves performance, particularly on the new classes. Notably, on the Pets, Cars, DTD, and EuroSAT datasets, the strategy achieves improvements of 6.32%, 9.18%, 9.78%, and 10.98% on the new classes compared to 'w/o spa'. These results further confirm the effectiveness of the proposed strategy.

I Visualization

In this section, we visualize the unseen-class images generated to demonstrate the effectiveness of the proposed class-domain-wise data generation pipeline. We compare the proposed method with three prompt templates for the text-to-image model mentioned in the ablation studies, namely, 'A picture of a class', 'A photo of class', and 'An image of a class'. We use the images generated based on the Caltech101 dataset for analysis.

We use the caption-based domain information generator to capture the class-specific domain information of seen-class data. This domain information is then used to generate the corresponding seen-class data via a text-to-image model. For visualization, we adopt the seen classes 'motorbike' and 'barrel'. As shown in Figures 4 and 5, compared to the three commonly used prompt templates, the generated seen-class data from the proposed pipeline better align with the seen-class data in terms of both style and scene information. This demonstrates that our pipeline is more effective at capturing the domain information of seen-class images for data generation.

Regarding the generation of unseen-class data, we use the hierarchy-based unseen class predictor to infer that the unseen classes 'car' and 'drum' are closest to 'motorbike' and 'barrel', respectively. The captured class-specific domain information and inferred unseen classes are then used to generate the unseen-class images via a text-to-image model. We compare the generated unseen-class data from our pipeline with data generated using the three commonly used prompt templates. As shown in Figures 4 and 5, the generated unseen-class data align better with the seen-class data. For example, in Figure 4, the car generated by our pipeline reflects the style of the seen-class data, and the realistic scene depicted in the generated images mirrors the scene in the seen-class data. These results further demonstrate that our pipeline effectively captures the domain information of seen-class images, and the generated unseen-class images align closely with seen-class data, confirming the effectiveness of the proposed pipeline.

Table 13: Ablation studies on class quality and data quality. “Acc” denotes the accuracy. “Dis” denotes the distribution distance of the generated unseen-class data and seen-class data.

Dataset		Ours		Class Quality				Data Quality					
		Acc \uparrow	Dis \downarrow	LowSim		w/o Tree		Picture		Photo		Image	
				Acc \uparrow	Dis \downarrow	Acc \uparrow	Dis \downarrow	Acc \uparrow	Dis \downarrow	Acc \uparrow	Dis \downarrow	Acc \uparrow	Dis \downarrow
Caltech	Base	98.97		98.26		97.93		98.26		98.19		98.06	
	New	95.85	9.99	94.32	10.49	93.67	13.16	94.21	11.84	94.11	11.95	93.78	12.12
	H	97.38		96.25		95.75		96.19		96.11		95.87	
Pets	Base	96.01		95.85		95.00		95.69		95.43		95.27	
	New	98.27	7.58	97.60	8.19	96.81	11.71	97.04	9.14	96.98	9.21	96.92	10.27
	H	97.12		96.72		95.90		96.36		96.20		96.09	
Cars	Base	82.93		78.94		77.86		77.99		78.71		78.24	
	New	80.81	8.92	75.29	9.33	73.36	13.78	74.75	10.65	75.04	10.08	74.92	10.48
	H	81.86		77.07		75.54		76.33		76.83		76.54	
Flowers	Base	98.77		98.29		97.15		97.82		97.91		97.63	
	New	80.92	6.29	77.52	6.88	75.04	10.29	76.88	7.99	77.09	7.84	76.17	8.22
	H	88.96		86.68		84.67		86.09		86.26		85.57	
Food	Base	91.39		90.90		90.45		90.56		90.60		90.81	
	New	92.99	9.01	92.25	9.49	91.79	13.80	91.83	11.24	91.97	11.10	92.02	10.94
	H	92.18		91.57		91.12		91.19		91.28		91.41	
Aircraft	Base	48.98		43.88		42.56		42.98		42.80		43.64	
	New	44.03	8.63	37.97	8.89	34.85	15.79	36.23	12.86	35.51	13.64	36.89	12.04
	H	46.37		40.71		38.32		39.32		38.82		39.98	
DTD	Base	85.53		83.91		81.48		83.22		82.87		83.57	
	New	71.50	7.67	65.22	8.01	57.73	8.84	63.77	8.38	62.32	8.73	63.89	8.20
	H	77.89		73.39		67.58		72.21		71.14		72.41	
EuroSAT	Base	97.17		94.71		91.05		91.83		92.12		91.62	
	New	87.90	11.48	80.49	11.71	65.28	12.39	68.08	11.88	71.67	11.82	67.41	12.07
	H	92.30		87.02		76.04		78.19		80.62		77.67	
UCF	Base	89.14		86.97		85.88		86.25		86.66		86.14	
	New	82.53	11.70	77.99	12.29	76.53	13.39	77.45	12.70	77.88	12.67	77.29	13.06
	H	85.71		82.23		80.94		81.61		82.04		81.47	

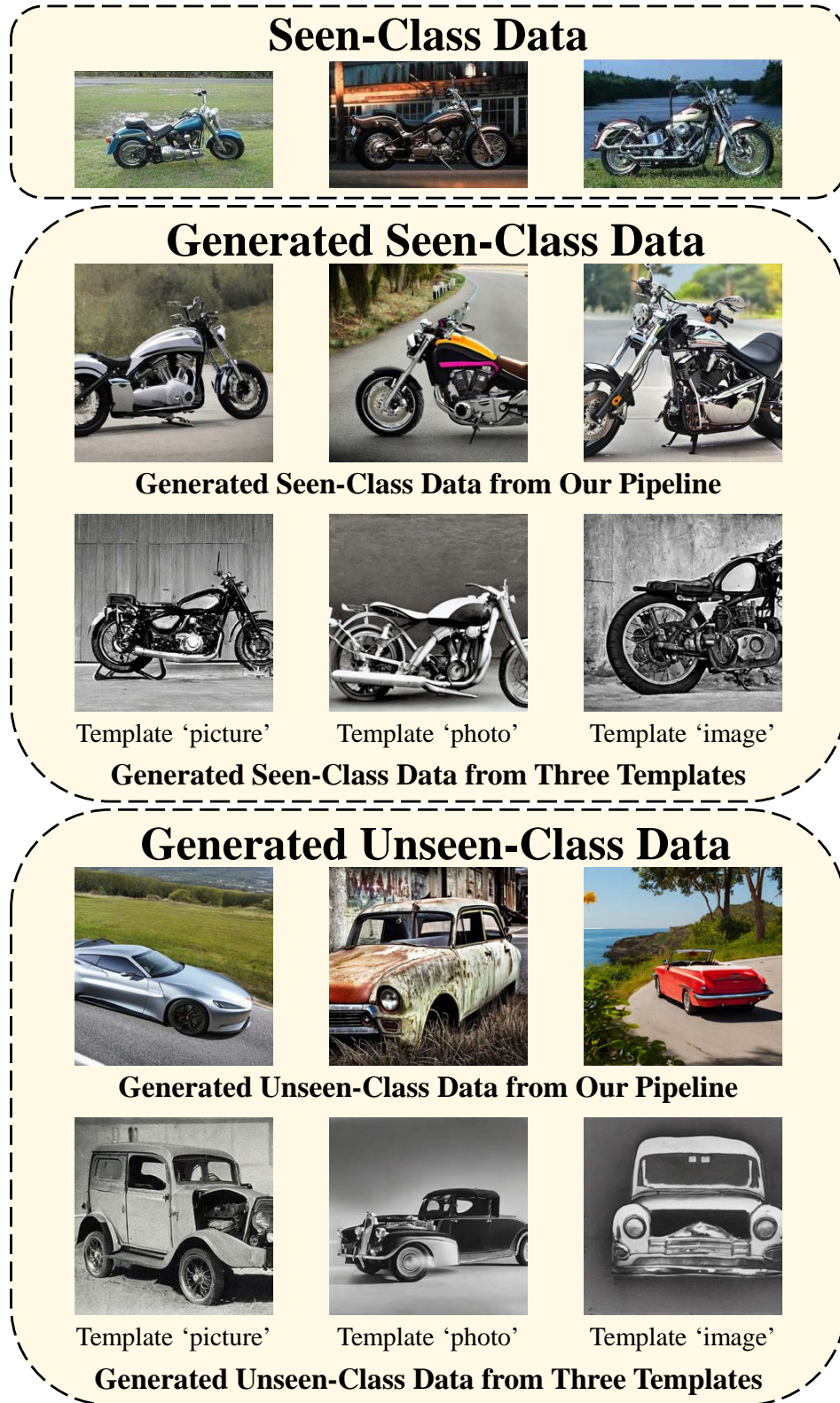


Figure 4: Comparison between the images generated with class-domain-wise data generation pipeline and three prompt templates mentioned in ablation studies. The seen class is 'motorbike' and the inferred unseen class is 'car'.

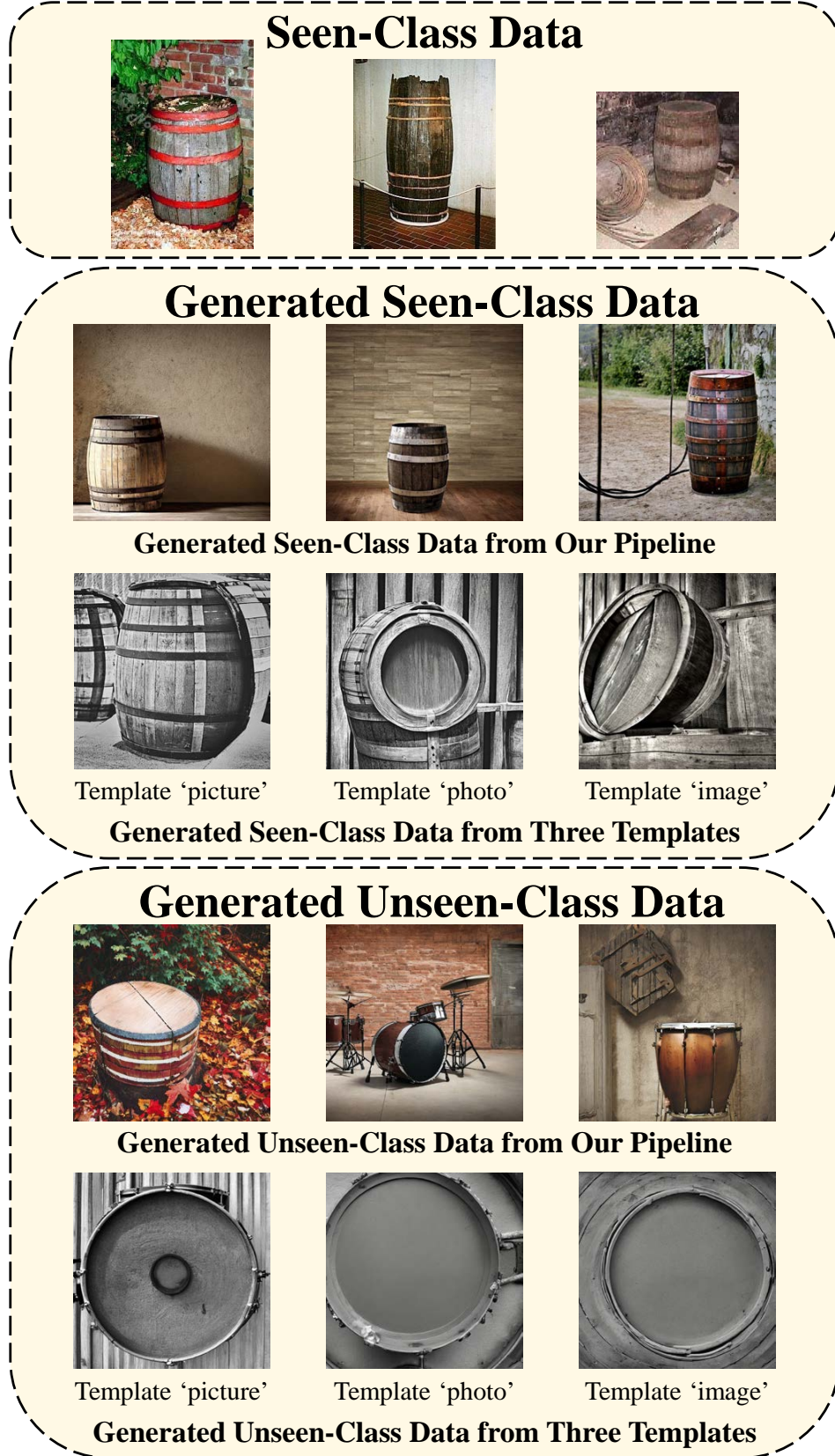


Figure 5: Comparison between the images generated with class-domain-wise data generation pipeline and three prompt templates mentioned in ablation studies. The seen class is 'barrel' and the inferred unseen class is 'drum'.