Progressive Pseudo-label Self-balancing Towards Unsupervised Vision-Language Models Adaptation

Si Qin[1], **Yaxin Hou**[1], **Jiawei Tang**[1], **Yuheng Jia**[1,2*]
[1]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[2]Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China
{siqin,yaxin,230259148,yhjia}@seu.edu.cn

Abstract

Enhancing Vision-Language Models (VLMs) on downstream tasks with unlabeled data has attracted increasing attention. Constructing a pseudo-labeled dataset by pseudo-labeling is a common method. Nonetheless, owing to the inherent class-wise prediction bias in VLMs, they tend to generate long-tailed pseudo-labels that are inconsistent with the true label distribution. Motivated by this, we first divide all classes into pseudo-head, pseudo-middle, and pseudo-tail classes based on the pseudo-label quantity. Accordingly, we propose a **P**rogressive **P**seudo-label **S**elf-**B**alancing (PPSB) framework, which measures the class-wise prediction bias and calculates quantity upper bounds for certain classes to construct relatively balanced pseudo-labeled dataset, while introducing fewer incorrect pseudo-tail labels. In this way, the class-wise prediction bias is progressively corrected and the pseudo-label dataset scale increases autonomously during training. Furthermore, the confidence scores generated by VLMs are not entirely reliable. Therefore, we design a neighborhood consistency filtering mechanism and a visual classifier to improve the pseudo-label accuracy. We theoretically prove that our method can significantly reduce the generalization error under certain conditions. Extensive experiments across six benchmarks under three learning paradigms demonstrate that our method outperforms state-of-the-art methods by average **3.84**% in accuracy. The code is available at https://github.com/siqin-seu/PPSB-NeurIPS-26.

1 Introduction

Vision-Language Models (VLMs) [1, 2, 3, 4] are pre-trained on large-scale image-text pairs, which align images with corresponding textual descriptions and bridge the gap between visual and textual modalities. Recent studies [5, 6] have shown that VLMs still require labeled data to improve performance on downstream tasks. However, obtaining substantial labeled data is impractical and expensive [7, 8], so how to fine-tune VLMs with unlabeled data has attracted increasing attention.

Recent studies [9, 10] rely on the zero-shot capability of VLMs to generate pseudo-labels for unlabeled data. Nonetheless, VLMs tend to generate long-tailed pseudo-labels that are inconsistent with the true label distribution (See Fig.1). The bias mainly arises from two factors: (i) During pre-training, certain classes appear more frequently in the training data, leading VLMs to exhibit a stronger prediction bias towards these classes [11]. (ii) VLMs exploit class names to build a textual classifier for zero-shot classification, but some class names fail to accurately represent the underlying semantics of corresponding classes, leading to their under-prediction [10]. For instance, in the FGVC-Aircraft dataset, certain class names (e.g., "747-300") do not precisely convey the semantic meaning of the corresponding class ("Boeing 747-300"). To model class-wise imbalance in

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

pseudo-labels, we divide all classes into pseudo-head, pseudo-middle, and pseudo-tail classes (Fig. 1). The prefix "pseudo-" is adopted because these classes are divided by pseudo-label quantity.

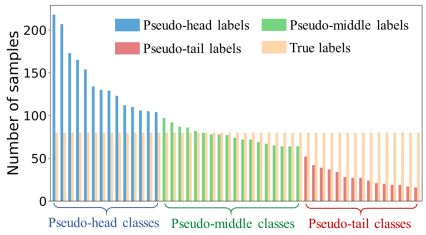


Figure 1: Long-tailed pseudo-label distribution on the dataset DTD.

Fine-tuning VLMs on imbalanced pseudo-labels inevitably leads to inferior performance [7]. Existing methods address this issue by constructing balanced pseudo-labeled dataset. Typically, FPL [9], and GRIP [9] filter the top-k confident pseudo-labeled samples for each class to construct pseudo-labeled dataset. CPL [7], AiR [10], and CAP [8] are built on the basis of GRIP or FPL. Among these methods, CPL generates partial labels [12] instead of hard pseudo-labels. CAP additionally incorporates a subset of high-confidence samples from the remaining unlabeled data and alleviates bias via confusion-aware calibrated margin. AiR exploits diffusion-generated images to build an auxiliary classifier and mitigate the bias of VLMs. However, this balanced strategy aligns the pseudo-label quantity of all classes, resulting in more incorrect pseudo-tail labels (See Fig. 2) and consequently impeding model training. Furthermore, existing methods merely rely on unreliable confidence scores [11] to filter pseudo-labels, not fully exploiting the representation information provided by samples.

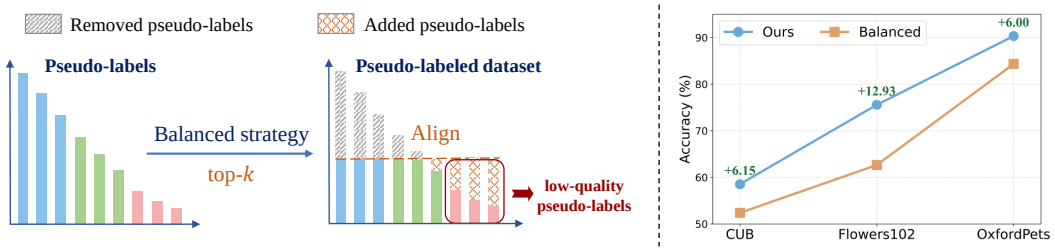


Figure 2: *left:* Balanced strategy aligns the pseudo-label quantity of all classes, resulting in more incorrect pseudo-tail labels. *right:* Pseudo-tail label accuracy across three datasets. Existing balanced strategy reduces the accuracy.

To address these challenges, we propose a **P**rogressive **P**seudo-label **S**elf-**B**alancing (PPSB) framework. As illustrated in Fig. 3, PPSB avoids generating extra pseudo-tail labels. Instead, it leverages the division of long-tailed pseudo-labels to measure the class-wise prediction bias and calculate an upper bound $\mathcal{U}$ to control the pseudo-head and pseudo-middle label quantity, thereby constructing a relatively balanced pseudo-labeled dataset. In the early stage of training, the model suffers from severe bias, resulting in a small $\mathcal{U}$ and constructing a small-scale pseudo-labeled dataset. As training on the relatively balanced dataset, upper bound $\mathcal{U}$ increases with the class-wise prediction bias gradually corrected, and all samples are progressively incorporated into the pseudo-labeled dataset.

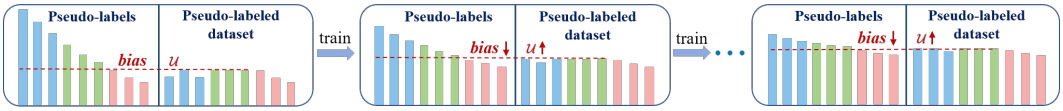


Figure 3: Overview of the progressive pseudo-label self-balancing framework.

To fully exploit the representation information of samples, we design the image-text neighborhood consistency filtering mechanism. Specially, it constructs an image-text representation matrix and captures its neighborhood information to filter high-quality pseudo-labels. Compared with the image representation matrix, our image-text representation matrix provides more discriminative neighborhood information (See Fig. 4). Furthermore, we introduce visual prototypes to build a visual classifier, and combine it with the class-wise prediction biased VLMs to generate more reliable pseudo-labels. Unlike AiR [10], we compute visual prototypes with reliable pseudo-labeled samples, without depending on generative models [13]. Finally, we theoretically prove that our method can significantly reduce the generalization error under certain conditions.

In summary, the main contributions of this work are as follows.

- We propose a **P**rogressive **P**seudo-label **S**elf-**B**alancing (PPSB) framework to avoid generating incorrect pseudo-tail labels and control the relative balance of the pseudo-labeled dataset. As training proceeds, PPSB gradually corrects the class-wise prediction bias and expands the pseudo-labeled dataset autonomously.
- We present an image-text neighborhood consistency filtering mechanism to select reliable pseudo-labels rather than merely rely on confidence scores. Assisted by filtered pseudo-labeled samples, a visual classifier is built to generate more reliable pseudo-labels.
- We provide a theoretical analysis of the generalization error for unsupervised VLMs adaptation, and prove that our method significantly reduces the generalization error under certain conditions.
- Comprehensive experiments on six commonly used benchmarks (DTD, FGVC-Aircraft, CUB, Flowers102, OxfordPets, and StanfordCars) under three learning paradigms validate that our method achieves new state-of-the-art performances with **3.84**% average improvement.

2 Related Work

2.1 Vision-Language Models

Vision-Language Models (VLMs) [1, 2, 3, 4] are pre-trained on large-scale image-text pairs with contrastive learning methods, which bridge the gap between visual and textual modalities. VLMs, such as CLIP [1], consist of an image encoder and a text encoder. A set of text prompts (e.g., "*a photo of [class]*") are encoded by the text encoder to form textual classifier for zero-shot classification. However, their performance degrades when facing domains that are rarely covered in the pre-training datasets. In recent years, numerous works have further improved the performance of VLMs on downstream tasks by fine-tuning them with annotated data [14, 15, 16]. In this paper, we mainly focus on fine-tuning CLIP [1] with unlabeled data to reduce the dependence on labeled data.

2.2 Prompt Tuning

Prompt tuning has emerged as a parameter-efficient alternative to full fine-tuning for adapting large-scale pre-trained models, such as CLIP [1] and GPT [17], to specific downstream tasks. In the context of VLMs, prompt tuning has been explored across different modalities. Textual prompt tuning, pioneered by CoOp [14], replaces manual templates with learnable context vectors in the textual branch to automate prompt engineering. Conversely, Visual Prompt Tuning [18] introduces a small budget of trainable parameters directly into the input sequence of the vision transformer's layers, providing an efficient adaptation strategy for the image encoder. In this paper, we fine-tune the prompts for class names in a manner similar to CoOp [14], enabling the CLIP model to adapt to downstream classification tasks.

2.3 Learning from pseudo-labels

In certain real-world downstream tasks, obtaining even a small portion of labeled data is expensive [7, 8, 10]. Recently, numerous studies have investigated fine-tuning downstream tasks with unlabeled data. These approaches mainly rely on the zero-shot capability of VLMs to generate pseudo-labels for unlabeled data. Existing pseudo-labeling methods construct balanced pseudo-labeled dataset to avoid the influence of imbalanced pseudo-labels generated by VLMs. FPL [9] select the top-k confident samples for each class. GRIP [9] regenerates pseudo-labels and filters top-k confident samples for classes within each iteration, where k increases with the predefined iterations. CPL [7] generates partial labels instead of hard pseudo-labels. CAP [8] additionally incorporates a subset of high-confidence samples from the remaining unlabeled data and alleviates bias via confusion-aware calibrated margin. AiR [10] exploits diffusion-generated images to build an auxiliary classifier. However, CPL, CAP, and AiR are still built on the basis of GRIP. This balanced strategy would introduce numerous incorrect pseudo-tail labels and hinder model training.

3 Methodology

Problem Definition. This paper addresses the problem of adapting VLMs to downstream tasks with only unlabeled data. Formally, given a set of unlabeled data $\mathcal{D}_{\mathrm{U}} = \{(x_i)\}_{i=1}^{N}$ containing N samples,

we focus on generating pseudo-label p_i for i-th sample within a class name space $\mathcal{Y} = \{c_j\}_{j=1}^{C}$ containing C classes. Then, we filter a portion of the pseudo-labels to construct the pseudo-labeled dataset $\mathcal{D}_L = \{(x_t, p_t)\}_{t=1}^{M}$ of size M, which is leveraged to fine-tune VLMs.

3.1 Progressive Pseudo-label Self-balancing Framework

Long-tailed pseudo-label distribution. The pseudo-labels generated by VLMs exhibit a typical long-tailed distribution (Fig. 1). Accordingly, we divide all classes into pseudo-head, pseudo-middle, and pseudo-tail classes based on the pseudo-label quantity of each class. The specific division method is as follows:

$$\begin{aligned} \mathcal{Y}_h &= \left\{ c_j \mid \tau_h \frac{N}{C} < N_{c_j} \right\}, \\ \mathcal{Y}_m &= \left\{ c_j \mid \tau_t \frac{N}{C} < N_{c_j} \leqslant \tau_h \frac{N}{C} \right\}, \\ \mathcal{Y}_t &= \left\{ c_j \mid N_{c_j} \leqslant \tau_t \frac{N}{C} \right\}, \end{aligned} \tag{1}$$

where N_{c_j} denotes the number of pseudo-labels for class c_j. $\mathcal{Y}_h$, $\mathcal{Y}_m$ and $\mathcal{Y}_t$ represent the class sets of pseudo-head, pseudo-middle and pseudo-tail classes, respectively. τ_h and τ_t are fixed thresholds for dividing $\mathcal{Y}_h$, $\mathcal{Y}_m$ and $\mathcal{Y}_t$. Sensitivity analysis of the two thresholds is shown in the Appendix F.1.

Pseudo-label self-balancing. If the model exhibits mild bias, it tends to predict a larger number of pseudo-middle and pseudo-tail labels. Therefore, the pseudo-middle and pseudo-tail label quantity is utilized to measure the class-wise prediction bias degree. We consider setting an upper bound $\mathcal{U}$ on the number of pseudo-head and pseudo-middle labels to control the relative balance of the pseudo-labeled dataset, which can be defined as:

$$\mathcal{U} = \left\lceil \frac{\sum_{c \in \mathcal{Y}_m \cup \mathcal{Y}_t} N_c}{|\mathcal{Y}_m \cup \mathcal{Y}_t|} \right\rceil. \tag{2}$$

$\mathcal{U}$ is the average number of pseudo-middle and pseudo-tail labels. A smaller $\mathcal{U}$ indicates more severe class-wise prediction bias.

For all pseudo-head labels, we sort them in descending order of confidence scores, and retain maximum $\mathcal{U}$ pseudo-labels for each class via image-text neighborhood consistency filtering mechanism (See ξ 3.2). For each pseudo-middle class, we observe that the precision of its pseudo-labels is relatively high (See Appendix F.2). Therefore, we directly retain up to $\mathcal{U}$ pseudo-labels in descending order of confidence scores. Since the number of pseudo-tail labels is small, we add all of them to the pseudo-labeled dataset.

Progressive process. *How does $\mathcal{U}$ control the pseudo-labeled dataset scale?* At the initial training stage, the small pseudo-tail label quantity results in a small $\mathcal{U}$, thus constructing a small scale and relative balanced the pseudo-labeled dataset. As the model is trained on the relative balanced labeled dataset, it gradually corrects the bias, and the number of pseudo-tail labels increases steadily. Meanwhile, $\mathcal{U}$ gradually increases as the model trains, and the pseudo-labeled dataset also expands continuously. Overall, we require no hyperparameters like iteration times to adjust the scale of the pseudo-labeled dataset. In contrast, the pseudo-labeled dataset expands autonomously based on the class-wise prediction bias degree.

3.2 Image-text Neighborhood Consistency Filtering Mechanism

Image-text representation matrix construction. Existing pseudo-labeling methods rely on confidence scores to filter high-quality pseudo-labels. However, the pseudo-label confidence scores generated by VLMs are not completely reliable. This raises the question:

How to filter reliable pseudo-labels beyond merely relying on confidence scores?

In this section, we attempt to select more reliable pseudo-labels exploiting image-text neighborhood information. Denote the image encoder and text encoder of VLMs as $\mathcal{I}(\cdot)$ and $\mathcal{T}(\cdot)$, respectively. We extract image feature vector $\mathbf{x}_i \in \mathbb{R}^d$ of each $x_i \in \mathcal{D}_U$ through $\mathcal{I}(\cdot)$ to form the image representation matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times d}$ and extract text feature vector $\mathbf{c}_j \in \mathbb{R}^d$ of

each $c_j \in \mathcal{Y}$ through $\mathcal{T}(\cdot)$ to form the text representation matrix $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_C]^\top \in \mathbb{R}^{C \times d}$. We try to construct $\mathbf{W} \in \mathbb{R}^{N \times C}$, an image-text representation matrix, to obtain more discriminative representation for each sample. The optimization problem for solving $\mathbf{W}$ is given as follows:

$$\min_{\mathbf{W}} \|\mathbf{X} - \mathbf{W}\mathbf{C}\|_F^2 + \gamma \|\mathbf{W}\|_F^2, \tag{3}$$

where γ is a regularization coefficient that prevents $\mathbf{W}$ from falling into a trivial solution. Problem 3 is a standard ridge regression problem, whose closed-form solution is given by:

$$\mathbf{W} = \mathbf{X}\mathbf{C}^\top \left(\mathbf{C}\mathbf{C}^\top + \gamma \mathbf{E}_C\right)^{-1}, \tag{4}$$

where $\mathbf{E}_C \in \mathbb{R}^{C \times C}$ is an identity matrix. Each row of $\mathbf{W}$ represents the image-text feature vector of sample x_i, and is essentially the set of coefficients for a linear combination of text feature vectors guided by $\mathbf{x}_i$.

Neighborhood consistency filtering. For sample x_i, we extract its representation $\mathbf{w}_i$ from $\mathbf{W}$, and calculate the cosine similarity $cos(\mathbf{w}_i, \mathbf{w}_j)$ between x_i and each sample x_j $(i \neq j)$ from $\mathcal{D}_U$. Next, search for k samples $(x_1, \cdots, x_k)$ with the maximum cosine similarity to x_i. If the following equality holds:

$$\sum_{t=1}^{k} \mathbb{I}[p_t = p_i] = k, \tag{5}$$

then the pseudo-label p_i of x_i is considered reliable and added to pseudo-labeled dataset. $\mathbb{I}[\cdot]$ is an indicator function that takes 1 if the equality in it holds, and 0 otherwise.

Why we exploit image-text representation matrix $\boldsymbol{W}$ instead of image representation matrix $\boldsymbol{X}$? There are two main reasons:

- VLMs are trained on image-text pairs, and their zero-shot classification capability mainly stem from the alignment between visual and textual modality. Therefore, the cross-modal representation matrix $\mathbf{W}$ can better capture the differences among different samples. As shown in the Fig. 4, $\mathbf{W}$ achieves higher clustering accuracy than $\mathbf{X}$.
- During model training, we adopt a prompt tuning method similar to CoOp to fine-tune the prompts for class names, thereby yielding the $\mathbf{C}$ with more accurate semantic representations. Subsequently, we can solve $\mathbf{W}$ with higher clustering accuracy (See Fig. 4) via Eq. 4 during training.

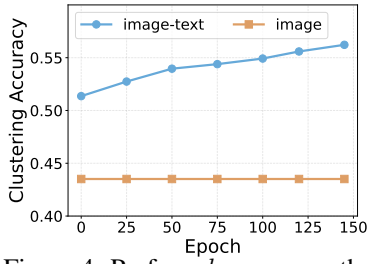


Figure 4: Perform k-means on the image representation matrix $\mathbf{X}$ and the image-text representation matrix $\mathbf{W}$ of the DTD dataset. $\mathbf{W}$ achieves higher accuracy.

3.3 Pseudo-labels Regeneration Assisted by Visual Prototypes

Original VLMs compute the class probability distribution $\hat{y}_i$ and pseudo-label p_i of sample $x_i \in \mathcal{D}_U$ with the textual classifier $\mathbf{C}$ as follows:

$$\hat{y}_i = \sigma(\mathbf{C}\mathbf{x}_i), \quad p_i = \arg\max_{c_j \in \mathcal{Y}} \left(\hat{y}_i^{c_j}\right), \tag{6}$$

where $\sigma(\cdot)$ denotes the softmax function and $\hat{y}_i^{c_j}$ denotes the probability of class c_j. For each class $c_j \in \mathcal{Y}$, we compute its visual prototype $\mathbf{I}_{c_j} \in \mathbb{R}^d$ with reliable pseudo-labeled samples in $\mathcal{D}_L$:

$$\mathbf{I}_{c_j} = \frac{1}{N_{c_j}} \sum_{t=1}^{M} \mathbb{I}[p_t = c_j] \mathbf{x}_t, \tag{7}$$

where $N_{c_j} = \sum_{t=1}^{M} \mathbb{I}[p_t = c_j]$. Then, a visual classifier $\mathbf{I} = [\mathbf{I}_1, \mathbf{I}_2, \cdots, \mathbf{I}_C]^\top \in \mathbb{R}^{C \times d}$ is built by concatenating all visual prototypes together. With the aid of visual classifier $\mathbf{I}$, we compute the class probability distribution of sample $x_i \in \mathcal{D}_U$:

$$\hat{y}_i = f(\mathbf{x}_i) = \sigma(\alpha \mathbf{C}\mathbf{x}_i + \mathbf{I}\mathbf{x}_i), \tag{8}$$

where α is a trade-off parameter. By fusing the logits from the visual and textual classifiers, we alleviate the bias of textual classifier with visual prototypes to regenerate more reliable pseudo-labels via Eq. 6. It should be noted that after regenerating pseudo-labels, we need to adjust the relative balance of the pseudo-labeled dataset.

3.4 Model Training and Prediction

Following CAP [8], we utilize LLMs to generate class descriptions to replace the original class names, thereby alleviating the concept mismatch problem. During model training, the image encoder and text encoder of VLMs are frozen, and only the prompts for the class descriptions are fine-tuned. We train the model with the cross-entropy loss function:

$$\mathcal{L}_{cro} = \mathbb{E}_{x_t \in \mathcal{D}_L} \left[-\sum_{c=1}^{C} \mathbb{I}\left[p_t = c\right] \log\left(y_t^c\right) \right]. \tag{9}$$

The detailed training process is shown in Appendix A. After model training, for a test sample x with image embedding $\mathbf{x}$, we calculate its logits via Eq. 8 and predict its class via Eq. 6.

3.5 Theoretical Analysis

We prove that our method can significantly reduce the generalization error under some conditions. Let $\ell_{avg} = \frac{1}{M_s}\left[\sum_{t=1}^{M_s} \ell_{cro}(f(x_t), p_t)\right]$ be the training loss on the M_s pseudo-labeled samples at the s-th training step, where p_t denotes the pseudo-label. Denote the pseudo-labeling error rate at the s-th training step as ϵ_s. Define the model empirical risk R at the s-th training step as:

$$R_s = R_{s-1} - G_s + I_{\epsilon_s} + I_{M_s}, \tag{10}$$

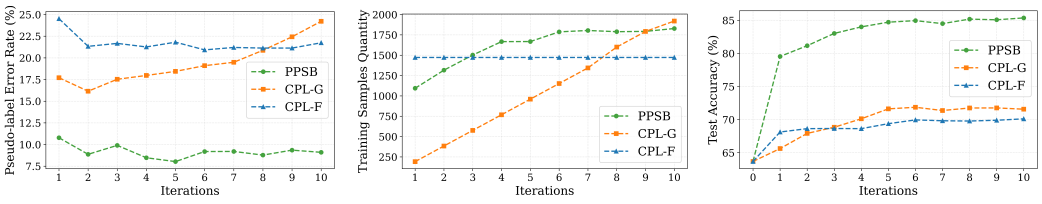
where $G_s \geq 0$ quantifies the empirical risk reduction achieved, and I_{ϵ_s} and I_{M_s} represent the impact of incorrect pseudo-labels and number of training samples, respectively. Let the function space $\mathcal{H}_c$ for the class $c \in \{1, \ldots, C\}$ be $\{h : x \longmapsto f_c(x) | f \in \mathcal{F}\}$, where $f_c(x)$ denotes the predicted probability of the c-th class and $\mathcal{F}$ denotes the function set of model f. Let $\mathcal{R}_{M_s}(\mathcal{H}_c)$ be the expected Rademacher complexity [19] of $\mathcal{H}_c$ with M_s training samples. The theorem is as follows:

Theorem 1 (Generalization Error)**.** *Suppose that the loss function $\ell_{cro}(f(x_t), c)$ is ρ-Lipschitz with respect to $f(x_t)$ for all $c \in \{1, \ldots, C\}$ and upper-bounded by B. Given the pseudo-labeling error rate $0 < \epsilon < 1$, for any $\delta > 0$, with probability at least $1 - \delta$, we have:*

$$R_S \leq R_0 - \sum_{s=1}^{S} G_s + B \sum_{s=1}^{S} \epsilon_s + 4\sqrt{2}\rho \sum_{s=1}^{S} \sum_{c=1}^{C} \mathcal{R}_{M_s}(\mathcal{H}_c) + 2B \sum_{s=1}^{S} \sqrt{\frac{\log \frac{2}{\delta}}{2M_s}}, \tag{11}$$

where S denotes the total number of training steps, R_S and R_0 represent the model empirical risk at the final training step and initial training step (with zero-shot pseudo-labels), respectively, and ϵ_s represents the pseudo-labeling error rate at the s-th training step.

The proof of Theorem 1 is provided in Appendix B. Based on Theorem 1, model empirical risk R_S mainly depends on three key factors, i.e., **the cumulative model empirical risk reduction $\sum_{s=1}^{S} G_s$, the cumulative pseudo-labeling error rate $\sum_{s=1}^{S} \epsilon_s$, and training samples quantity M_s at step s.** As shown in Figs. 5(a) and 5(b), our PPSB outperforms other methods, achieving both: (i) a consistently lower pseudo-labeling error rate ϵ_s (due to the combined contributions of the pseudo-label self-balancing framework, the neighborhood consistency filtering mechanism, and the visual classifier), and (ii) a growing number of training samples M_s gradually approaches the full dataset (due to the assistance of the progressive process). Moreover, training on a relatively balanced pseudo-labeled dataset with lower pseudo-label error rate yields higher cumulative model empirical risk reduction and ultimately reduces the generalization error significantly.



(a) Pseudo-labeling error rate. (b) Training samples quantity. (c) Test accuracy.

Figure 5: Comparison of pseudo-labeling error rate (a), training samples quantity (b), and test accuracy (c) among CPL-G [9], CPL-F [9], and PPSB on the dataset Flw. PPSB achieves lower pseudo-labeling error rate and comparable training samples quantity, obtaining higher test accuracy.

4 Experiments

To demonstrate the effectiveness of the proposed method, we conduct experiments under three different learning paradigms across six datasets and compare with eight SOTA methods. In addition, we conduct ablation studies and analytical experiments to explain why our method is effective.

4.1 Experiments Settings

Datasets and learning paradigms. We conducted experiments on six image classification datasets from distinct domains, including DTD (DTD.) [20], FGVC-Aircraft (FA.) [21], CUB (CUB.) [22], Flowers102 (Flw.) [23], OxfordPets (Pets.) [24], and StanfordCars (Cars.) [25]. To comprehensively evaluate our method, we consider three learning paradigms, including unsupervised learning (UL), semi-supervised learning (SSL) [26, 27, 28, 29], and transductive zero-shot learning (TRZSL) [30, 31]. More details about datasets and learning paradigms can be find in Appendix C.

Baselines. We compare our method with eight SOTA methods, namely, CLIP [1], FPL [9], GRIP [9], CPL-F [7], CPL-G [7], UEO [32], AiR [10], and CAP [8]. Appendix D shows more details about comparison methods. To ensure a fair comparison, we select the textual prompting tuning for each method except CAP which uses multi-modal prompts as MaPLe [33].

Implementation details. In our experiments, the image encoder of CLIP adopts ViT-B/32 and the prefix size is set to 16 for textual prompt learning. We set to fixed $\tau_h = 1.1$, $\tau_t = 0.9$, $\gamma = 1$, $k = 2$, and $\alpha = 1$. Qwen-plus is employed to enhance the descriptions of class names. We adopt SGD as the optimizer and conduct training for maximum 150 epochs. The learning rate is adjusted to 0.01 and decays following the cosine annealing rule. More implementation details can be find in Appendix E.

4.2 Main Results

Table 1 compares the performance of different methods under three learning paradigms. We report the average test accuracy over three random seeds 1, 2, 3. It can be observed that the proposed method consistently outperforms existing methods under the UL setting. Notably, our approach achieves significant improvements on Flowers102, FGVCAircraft, and CUB datasets, exceeding other baselines by **9.65**%, **4.09**%, and **4.26**%, respectively. This demonstrates that our method can effectively leverage unlabeled data to achieve superior performance. Under the SSL and TRZSL settings, PPSB outperforms other best method by average **2.62**% and **3.85**%, respectively. This suggests the capability of our method to leverage labeled data with unlabeled data, further validating its versatility regarding different paradigms. We also provide the results of different methods fine-tuned with VPT [18] in Appendix F.3.

Table 1: Comparison results of test accuracy (%) across six datasets under three learning paradigms. Values are expressed as mean_{std}. The highest accuracies are bold.

	Flowers102			FGVCAircraft			CUB			
Methods	SSL	UL	TRZSL	SSL	UL	TRZSL	SSL	UL	TRZSL	Avg.
CLIP [1]	$63.67_{0.00}$		$63.40_{0.00}$	$17.58_{0.00}$		$17.86_{0.00}$	$51.82_{0.00}$		$51.57_{0.00}$	$44.32_{0.00}$
FPL [9]	$75.57_{0.30}$	$68.22_{0.14}$	$86.25_{0.23}$	$21.03_{0.80}$	$17.90_{0.19}$	$28.15_{0.32}$	$55.83_{0.19}$	$53.43_{0.11}$	$63.12_{0.45}$	$52.17_{0.23}$
GRIP [9]	$88.42_{0.58}$	$67.43_{0.70}$	$85.49_{1.41}$	$22.95_{0.45}$	$16.78_{0.40}$	$30.08_{0.82}$	$57.52_{0.57}$	$51.50_{1.07}$	$65.07_{0.26}$	$53.92_{0.43}$
CPL-F [7]	$78.27_{0.03}$	$70.14_{0.32}$	$86.20_{0.45}$	$22.93_{0.48}$	$19.14_{0.13}$	$28.02_{0.31}$	$56.31_{0.24}$	$54.15_{0.13}$	$64.42_{0.21}$	$53.29_{0.23}$
CPL-G [7]	$90.48_{0.30}$	$72.14_{0.32}$	$89.65_{1.96}$	$23.38_{0.11}$	$17.88_{0.25}$	$30.74_{0.41}$	$57.54_{0.33}$	$53.60_{0.23}$	$66.09_{0.45}$	$55.72_{0.34}$
UEO [32]	$61.61_{1.75}$	$62.11_{1.06}$	$57.87_{3.50}$	$20.12_{0.26}$	$16.83_{1.16}$	$19.27_{0.69}$	$52.65_{0.30}$	$51.38_{0.84}$	$51.03_{1.06}$	$43.65_{0.36}$
CAP [8]	$87.72_{1.25}$	$76.78_{0.68}$	$87.49_{0.82}$	$21.32_{0.48}$	$17.26_{0.27}$	$27.52_{0.33}$	$57.73_{0.70}$	$55.54_{0.12}$	$60.68_{0.22}$	$54.67_{0.39}$
AiR [10]	$89.31_{0.11}$	$67.00_{0.63}$	$86.60_{1.65}$	$22.26_{0.23}$	$16.32_{0.20}$	$29.71_{0.36}$	$57.47_{0.16}$	$48.83_{0.54}$	$66.11_{0.13}$	$53.73_{0.30}$
PPSB (Ours)	$\mathbf{92.81}_{0.14}$	$\mathbf{86.43}_{0.81}$	$\mathbf{94.56}_{0.08}$	$\mathbf{25.45}_{0.63}$	$\mathbf{23.23}_{0.30}$	$\mathbf{34.21}_{0.57}$	$\mathbf{61.05}_{0.87}$	$\mathbf{59.80}_{0.29}$	$\mathbf{69.02}_{0.55}$	$\mathbf{60.73}_{0.48}$

	DTD			OxfordPets			StanfordCars			
Methods	SSL	UL	TRZSL	SSL	UL	TRZSL	SSL	UL	TRZSL	Avg.
CLIP [1]	$43.24_{0.00}$		$43.45_{0.00}$	$84.43_{0.00}$		$83.10_{0.00}$	$58.79_{0.00}$		$58.17_{0.00}$	$61.86_{0.00}$
FPL [9]	$52.09_{0.33}$	$47.00_{0.42}$	$57.27_{0.33}$	$87.92_{0.52}$	$88.52_{0.62}$	$87.83_{1.02}$	$64.20_{0.39}$	$60.32_{0.30}$	$70.13_{0.18}$	$68.36_{0.34}$
GRIP [9]	$59.71_{0.84}$	$47.96_{0.89}$	$66.88_{0.40}$	$87.18_{0.33}$	$87.70_{0.47}$	$89.91_{0.87}$	$64.47_{0.89}$	$57.33_{0.31}$	$73.36_{0.58}$	$70.50_{0.56}$
CPL-F [7]	$55.14_{0.73}$	$48.82_{0.67}$	$59.55_{0.80}$	$87.24_{0.73}$	$86.97_{0.12}$	$87.97_{1.32}$	$64.79_{0.02}$	$60.81_{0.21}$	$71.25_{0.20}$	$69.17_{0.49}$
CPL-G [7]	$60.90_{0.37}$	$50.99_{1.29}$	$67.62_{0.92}$	$85.93_{0.44}$	$83.56_{0.28}$	$83.17_{0.83}$	$65.78_{0.19}$	$60.33_{0.95}$	$74.15_{0.80}$	$70.27_{0.45}$
UEO [32]	$45.85_{0.41}$	$44.02_{0.81}$	$45.08_{1.32}$	$87.01_{1.22}$	$86.23_{1.25}$	$77.65_{1.63}$	$57.60_{0.60}$	$59.25_{0.12}$	$58.31_{0.83}$	$62.33_{0.49}$
CAP [8]	$61.40_{0.51}$	$56.04_{0.47}$	$69.92_{0.77}$	$\mathbf{91.37}_{0.19}$	$91.11_{0.20}$	$90.09_{0.19}$	$66.16_{0.25}$	$61.31_{0.36}$	$67.50_{0.50}$	$72.77_{0.42}$
AiR [10]	$58.98_{0.28}$	$48.35_{0.62}$	$66.39_{1.56}$	$87.95_{0.61}$	$88.46_{0.15}$	$89.46_{0.12}$	$64.37_{0.25}$	$56.85_{0.94}$	$73.21_{0.30}$	$70.45_{0.36}$
PPSB (Ours)	$\mathbf{63.36}_{0.70}$	$\mathbf{57.26}_{0.41}$	$\mathbf{71.24}_{0.42}$	$91.06_{0.31}$	$\mathbf{91.26}_{0.03}$	$\mathbf{90.98}_{0.30}$	$\mathbf{67.65}_{0.15}$	$\mathbf{62.23}_{0.36}$	$\mathbf{74.56}_{0.19}$	$\mathbf{74.40}_{0.29}$

4.3 Ablation Studies

Ablation of key components in our method. Our method consists of three components: the progressive pseudo-label self-balancing framework (PPSB-F), neighborhood consistency filtering mechanism (denoted as NCF), and visual classifier (denoted as VC). We conduct experiments on the DTD., Flw., and CUB. under the UL setting. As shown in Table 2, PPSB-F brings a significant gain of 11.25% on average across three datasets. Based on the baseline PPSB-F, incorporating the NCF and VC separately results in gains of 1.36% and 1.74%, respectively. When the two components are combined, our method achieves a remarkable performance improvement of 14.76% in accuracy, reinforcing the effectiveness and robustness of our design.

Table 2: Comparison results of test accuracy (%) w/ and w/o key components.

PPSB-F	NCF	VC	DTD.	CUB.	Flw.	Avg.
✗	✗	✗	44.99	43.49	70.51	52.99
✓	✗	✗	54.30	54.57	83.85	64.24
✓	✓	✗	56.43	56.31	84.07	65.60
✓	✗	✓	55.55	57.97	84.42	65.98
✓	✓	✓	**56.88**	**60.03**	**86.36**	**67.75**

Ablation of image encoder. To evaluate the generalizability of our method on different visual backbones, we conduct experiments with a larger image encoder ViT-14/L under the UL setting. As shown in Table 3, our method consistently outperforms other baselines by average **5.85**% in accuracy. The results illustrate that our method exhibits strong robustness and adaptability across various visual backbones, and can fully leverage the representation of diverse vision models.

Table 3: Comparison results of test accuracy (%) with image encoder ViT-14/L.

Method	Flw.	FA.	CUB.	DTD.	Pets.	Cars.
CLIP [1]	73.99	30.51	62.65	52.39	92.91	76.44
FPL [9]	79.67	31.17	67.19	55.95	94.71	79.82
GRIP [9]	76.46	28.86	69.24	55.58	95.53	79.08
CPL-F [7]	80.64	31.95	68.22	58.56	94.30	80.36
CPL-G [7]	80.20	32.43	71.52	57.55	95.31	81.02
UEO [32]	76.82	34.20	61.40	57.02	94.79	76.35
CAP [8]	87.80	33.48	70.76	61.59	95.17	70.95
AiR [10]	76.77	28.29	65.41	53.98	95.06	78.70
PPSB (Ours)	**95.10**	**40.08**	**75.62**	**64.57**	**96.07**	**83.43**

Table 4: Comparison results of test accuracy (%) on the foundation model OpenCLIP.

Method	Flw.	FA.	CUB.	DTD.	Pets.	Cars.
CLIP [1]	63.58	17.55	51.72	43.13	84.43	58.79
FPL [9]	68.98	19.38	61.16	51.86	88.60	82.91
GRIP [9]	69.27	18.72	59.71	52.23	87.92	82.32
CPL-F [7]	71.88	19.41	62.58	53.19	89.17	83.91
CPL-G [7]	72.58	20.07	61.49	54.78	89.56	84.69
UEO [32]	66.54	21.45	56.90	51.22	88.90	82.02
CAP [8]	74.14	18.45	59.02	52.02	**91.11**	66.34
AiR [10]	70.15	17.70	59.21	51.64	88.38	82.19
PPSB (Ours)	**87.57**	**26.67**	**66.34**	**59.99**	89.97	**86.17**

Ablation of hyperparameters. The hyperparameters of our method are fixed across all the datasets and experiments. We conduct a sensitivity study to explore the impact of hyperparameters (e.g. α and k) on the datasets Pets. and FA. in Appendix F.1. It can be observed that our method is insensitive to these hyperparameters, demonstrating the robustness of our method.

4.4 More Analysis

Progressive pseudo-label self-balancing framework improves pseudo-tail labels. We compare the pseudo-tail label accuracy with several methods (including GRIP, CPL-G, and AiR) on the final iteration (set $\tau_t = 1.0$). Fig. 6 shows that PPSB improves pseudo-tail label accuracy across all the datasets, which demonstrates that it effectively avoids introducing incorrect pseudo-tail labels.

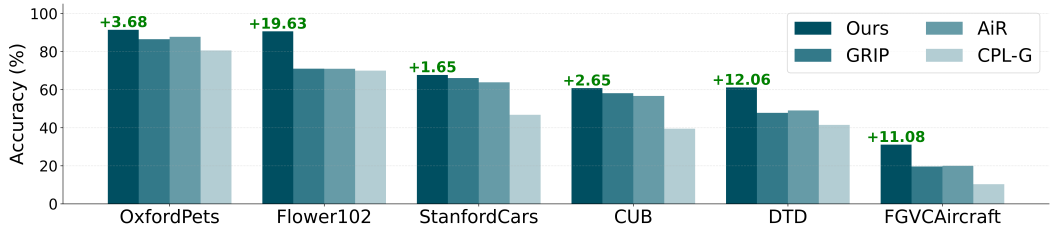
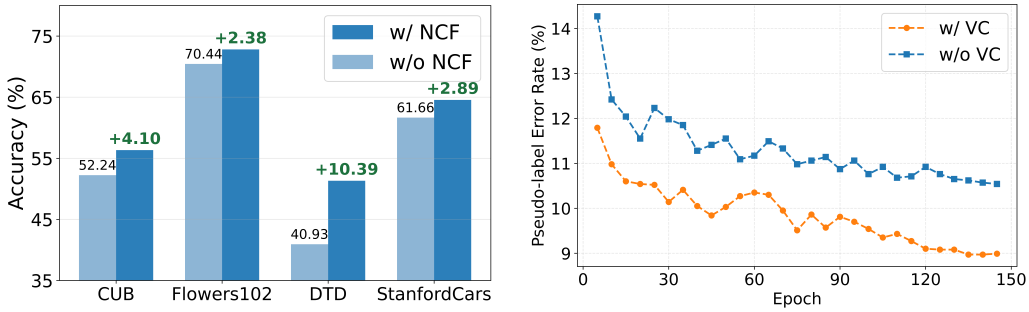


Figure 6: Comparison results of pseudo-tail label accuracy.

Neighborhood consistency filtering mechanism select more reliable pseudo-head labels. We compare pseudo-head label accuracy w/ and w/o neighborhood consistency filtering mechanism (NCF) across four datasets. As shown in Fig. 7(a), w/ NCF achieves a significant improvement of **10.39**% on the DTD and an average improvement of **4.94**% across four datasets.

Visual classifier consistently reduces the pseudo-label error rate. We conduct experiments on the dataset OxfordPets to compare the pseudo-label error rate w/ and w/o Visual Classifier (VC). As

shown in Fig. 7(b), w/ VC consistently exhibits lower pseudo-label error rate than w/o VC, which demonstrates that w/ VC can effectively generate more reliable pseudo-labels.



(a) Pseudo-head label accuracy w/ and w/o NCF.

(b) Pseudo-label error rate w/ and w/o VC.

Figure 7: (a) w/ NCF improve pseudo-head labels. (b) w/ VC reduce the pseudo-label error rate.

Evaluations on Imbalanced dataset. We explore the performance of different methods on the imbalanced datasets with varying imbalance ratios (10, 20). Experiments are conducted on four datasets under the UL setting. Existing methods perform quantity alignment for pseudo-labels of all classes. Instead, by controlling the relative balance of pseudo-labeled dataset, our method avoids introducing incorrect pseudo-tail labels. As shown in Table 5, PPSB consistently surpasses other baselines, which demonstrates the robustness of our method on imbalanced datasets.

Table 5: Comparison results of test accuracy (%) on the imbalanced datasets.

	Imbalance Ratio 10				Imbalance Ratio 20				
Methods	FA.	Pets.	Cars.	DTD.	FA.	Pets.	Cars.	DTD.	Avg.
FPL [9]	18.36	85.99	43.75	44.73	16.68	82.28	34.59	42.87	46.16
GRIP [9]	12.75	73.39	43.86	41.70	12.48	68.49	39.58	36.96	41.15
CPL-F [7]	17.22	86.64	58.36	47.55	16.74	85.71	56.77	44.89	51.74
CPL-G [7]	15.06	80.81	48.42	45.74	12.84	72.66	49.45	42.55	45.94
UEO [32]	14.64	81.65	57.40	39.41	14.67	80.43	57.07	37.28	47.83
CAP [8]	18.12	87.21	58.26	48.40	16.59	82.03	56.08	42.07	51.10
AiR [10]	15.09	74.32	44.06	39.20	11.94	64.67	38.36	38.56	40.78
PPSB (Ours)	**22.50**	**87.78**	**58.83**	**48.93**	**21.90**	**87.65**	**58.23**	**46.96**	**54.10**

Evaluations on different foundation models. To verify the effectiveness of our method on different foundation models under the UL setting, we conduct experiments on the OpenCLIP (LAION2B) with image encoder ViT-32/B across six datasets. Table 4 shows that PPSB exceeds other best baselines by average **5.59**%, which illustrates that our method can adapt to different foundation models.

Other experiments. We conduct more experiments (e.g. evolution of the pseudo-labels predictions, evolution of the pseudo-labeled dataset, statistical significance test, clustering performance of the image-text representation matrix, comparison of soft and hard pseudo-labels and evaluation with stronger vision-language models) in Appendix F, which further verifies the effectiveness of PPSB.

5 Conclusion

In this work, we delved into the issue of long-tailed pseudo-labels in unsupervised vision-language models adaptation. We proposed a progressive pseudo-label self-balancing framework that models long-tailed pseudo-label distribution and progressively corrects bias while introducing fewer pseudo-tail labels. By dynamically controlling the pseudo-head and pseudo-middle label quantity, PPSB constructs a relatively balanced pseudo-labeled dataset that expands autonomously during training. Furthermore, we introduced an image-text neighborhood consistency filtering mechanism and a visual classifier to improve pseudo-label accuracy. We also theoretically proved that our method reduces generalization error significantly under certain conditions. Extensive experiments across multiple benchmarks and learning paradigms demonstrate that our approach consistently outperforms existing methods by average **3.84**%, validating its effectiveness and robustness.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants U24A20322 and 62576094. This research work was also supported by the Big Data Computing Center of Southeast University.

References

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021.

[2] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900, 2022.

[3] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4818–4829, 2024.

[4] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems*, pages 9694–9705, 2021.

[5] Amit Peleg, Naman Deep Singh, and Matthias Hein. Advancing compositional awareness in CLIP with efficient fine-tuning. In *Advances in Neural Information Processing Systems*, 2025.

[6] Yung-Sung Chuang, Yang Li, Dong Wang, Ching-Feng Yeh, Kehan Lyu, Ramya Raghavendra, James Glass, Lifei Huang, Jason Weston, Luke Zettlemoyer, Xinlei Chen, Zhuang Liu, Saining Xie, Wen-tau Yih, Shang-Wen Li, and Hu Xu. Meta CLIP 2: A worldwide scaling recipe. In *Advances in Neural Information Processing Systems*, 2025.

[7] Jiahan Zhang, Qi Wei, Feng Liu, and Lei Feng. Candidate pseudolabel learning: Enhancing vision-language models by prompt tuning with unlabeled data. In *International Conference on Machine Learning*, pages 60004–60020, 2024.

[8] Yuchen Wang, Xuefeng Bai, Xiucheng Li, Weili Guan, Liqiang Nie, and Xinyang Chen. Handling imbalanced pseudolabels for vision-language models with concept alignment and confusion-aware calibrated margin. In *International Conference on Machine Learning*, pages 62357–62372, 2025.

[9] Cristina Menghini, Andrew Delworth, and Stephen Bach. Enhancing clip with clip: Exploring pseudolabeling for limited-label prompt tuning. In *Advances in Neural Information Processing Systems*, pages 60984–61007, 2023.

[10] Hairui Ren, Fan Tang, He Zhao, Zixuan Wang, Dandan Guo, and Yi Chang. Beyond words: Augmenting discriminative richness via diffusions in unsupervised prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25135–25144, 2025.

[11] Tony Huang, Jack Chu, and Fangyun Wei. Unsupervised prompt learning for vision-language models. *arXiv preprint arXiv:2204.03649*, 2022.

[12] Jie Luo and Francesco Orabona. Learning from candidate labeling sets. In *Advances in Neural Information Processing Systems*, 2010.

[13] 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 10674–10685, 2022.

[14] 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.

[15] 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 (CVPR)*, pages 16795–16804, 2022.

[16] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5625–5644, 2024.

[17] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pages 1877–1901, 2020.

[18] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge J. Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727, 2022.

[19] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, 2018.

[20] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3606–3613, 2014.

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

[22] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, California Institute of Technology, 2011.

[23] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, pages 722–729, 2008.

[24] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3498–3505, 2012.

[25] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 554–561, 2013.

[26] Yaxin Hou and Yuheng Jia. A square peg in a square hole: Meta-expert for long-tailed semi-supervised learning. In *International Conference on Machine Learning*, pages 23960–23975, 2025.

[27] Yaxin Hou, Bo Han, Yuheng Jia, Hui LIU, and Junhui Hou. Keep it on a leash: Controllable pseudo-label generation towards realistic long-tailed semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2025.

[28] Yaxin Hou, Jun Ma, Hanyang Li, Bo Han, Jie Yu, and Yuheng Jia. Beyond distribution estimation: Simplex anchored structural inference towards universal semi-supervised learning. *arXiv preprint arXiv:2605.07557*, 2026.

[29] Bo Han, Zhuoming Li, Xiaoyu Wang, Yaxin Hou, Hui Liu, Junhui Hou, and Yuheng Jia. Dicap: Distribution-calibrated pseudo-labeling for semi-supervised multi-label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 21540–21548, 2026.

[30] Ziyu Wan, Dongdong Chen, Yan Li, Xingguang Yan, Junge Zhang, Yizhou Yu, and Jing Liao. Transductive zero-shot learning with visual structure constraint. In *Advances in Neural Information Processing Systems*, pages 9972–9982, 2019.

[31] Rui Gao, Xingsong Hou, Jie Qin, Jiaxin Chen, Li Liu, Fan Zhu, Zhao Zhang, and Ling Shao. Zero-vae-gan: Generating unseen features for generalized and transductive zero-shot learning. *IEEE Transactions on Image Processing*, 29:3665–3680, 2020.

[32] Jian Liang, Lijun Sheng, Zhengbo Wang, Ran He, and Tieniu Tan. Realistic unsupervised CLIP fine-tuning with universal entropy optimization. In *International Conference on Machine Learning*, pages 29667–29681, 2024.

[33] Muhammad Uzair Khattak, Hanoona Abdul Rasheed, Muhammad Maaz, Salman H. Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19113–19122, 2023.

[34] Andreas Maurer. A vector-contraction inequality for rademacher complexities. In *Algorithmic Learning Theory (ALT)*, volume 9925, pages 3–17, 2016.

Table of Contents for the Appendix

- A: Training Process of PPSB 14
- B: Proof of Theorem 1 14
- C: Dataset and Learning Paradigm Details 17
- C.1: Dataset Details 17
- C.2: Learning Paradigm Details 17
- D: Comparison Methods 18
- E: Implementation Details 18
- F: More Experiment Results and Analysis 18
- F.1: Sensitivity Study of the Hyperparameters 18
- F.2: Precision of Pseudo-labels 19
- F.3: Comparison of Methods Fine-tuned with VPT 19
- F.4: Evolution of Pseudo-label Predictions 20
- F.5: Evolution of Pseudo-labeled Dataset 20
- F.6: Statistical Significance Test 20
- F.7: Clustering Performance of the Image-text Representation Matrix 21
- F.8: Comparison of Soft and Hard Pseudo-labels 21
- F.9: Evaluation with Stronger Vision-Language Models 22
- G: Limitations 22
- H: Compute Resources 22
- I: Broader Impacts 22

A Training Process of PPSB

Overview. The overall workflow of our method PPSB can be divided into three steps:

(i) Leverage VLMs to generate pseudo-labels, and divide all classes into pseudo-head, pseudo-middle, and pseudo-tail classes. (ξ 3.1).

(ii) Calculate an upper bound for pseudo-head and pseudo-middle label quantity (ξ 3.1). Neighborhood consistency filtering mechanism is adopt to filter reliable pseudo-labels (ξ 3.2).

(iii) Update the visual classifier with filtered reliable pseudo-labeled samples, which assists in regenerating more reliable pseudo-labels, and re-execute Steps (i) and (ii) to control the relative balance of the pseudo-labeled dataset (ξ 3.3).

Steps (i-iii) are repeated until the model converge. Algorithm 1 shows the training process of PPSB.

Algorithm 1 Training process of PPSB

Input: Unlabeled datasets $\mathcal{D}_U$.
Output: Prompts $\mathbf{P}$ for class descriptions.
Initialize the parameters of $\mathbf{P}$ randomly.
for epoch $= 0, 1, \ldots$ **do**
  if epoch$\%5 = 0$ **then**
    Generate the pseudo-labels of $\mathcal{D}_U$.
    Step 1: Divide all classes into head, middle, and tail classes via Eq. 1.
      Calculate the upper bound $\mathcal{U}$ via Eq. 2.
    Step 2: Filter reliable pseudo-labels according to section 3.1.
    Update the visual classifier $\mathbf{I}$ via Eq. 7 and regenerate the pseudo-labels of $\mathcal{D}_U$.
    Repeat Steps 1 and 2 to obtain relatively balanced pseudo-labeled dataset $\mathcal{D}_L$.
  end if
  Calculate $\mathcal{L}_{cro}$ via Eq. 9.
  Update parameters $\mathbf{P}$ via gradient descent.
end for

B Proof of Theorem 1

Theorem 1. *Suppose that the loss function $\ell_{cro}(f(x_t), c)$ is ρ-Lipschitz with respect to $f(x_t)$ for all $c \in \{1, \ldots, C\}$ and upper-bounded by B. Given the pseudo-labeling error rate $0 < \epsilon < 1$, for any $\delta > 0$, with probability at least $1 - \delta$, we have:*

$$R_S \leq R_0 - \sum_{s=1}^{S} G_s + B \sum_{s=1}^{S} \epsilon_s + 4\sqrt{2}\rho \sum_{s=1}^{S} \sum_{c=1}^{C} \mathcal{R}_{M_s}(\mathcal{H}_c) + 2B \sum_{s=1}^{S} \sqrt{\frac{\log \frac{2}{\delta}}{2M_s}}, \tag{12}$$

where S denotes the total number of training steps, R_S and R_0 represent the model empirical risk at the final training step and initial training step (with pseudo-labels generated by zero-shot capability of VLMs), respectively, and ϵ_s represents the pseudo-labeling error rate at the s-th training step.

Proof. We first restate the definition of the model empirical risk R at the s-th training step, i.e.,

$$R_s = R_{s-1} - G_s + I_{\epsilon_s} + I_{M_s}, \tag{13}$$

where R_s denotes the model empirical risk at the s-th training step, $G_s \geq 0$ quantifies the empirical risk reduction achieved, and I_{ϵ_s} and I_{M_s} represent the impact of incorrect pseudo-labels and number of training samples, respectively. We derive I_{ϵ_s} and I_{M_s} below, omitting the subscript s for brevity.

Denote the true risk on the training dataset with respect to the classification model $f(x; \theta)$ as:

$$R(f) = \mathbb{E}_{(x,y)} \left[\ell_{cro}(f(x), c) \right]. \tag{14}$$

We aim to learn a good classification model by minimizing the empirical risk $\widehat{R}(f) = \frac{1}{M} \sum_{t=1}^{M} \ell_{cro}(f(x_t), c_t)$ on the labeled dataset $\mathcal{D}_L$.

Next, we derive the uniform deviation bound I_M between the true risk $R(f)$ and the empirical risk $\widehat{R}(f)$ using the following lemma.

Lemma 1. *Suppose that the loss function $\ell_{cro}(f(x), c)$ is ρ-Lipschitz with respect to $f(x)$ for all $c \in \{1, \ldots, C\}$ and upper-bounded by B. For any $\delta > 0$, with probability at least $1 - \delta$, we have:*

$$|R(f) - \widehat{R}(f)| \leq 2\sqrt{2}\rho \sum_{c=1}^{C} \mathcal{R}_M(\mathcal{H}_c) + B\sqrt{\frac{\log \frac{2}{\delta}}{2M}}, \tag{15}$$

where the function space $\mathcal{H}_c$ for the label $c \in \{1, \ldots, C\}$ is $\{h : x \longmapsto f_c(x) | f \in \mathcal{F}\}$.

Proof. In order to prove this lemma, we define the Rademacher complexity of the composition of loss function ℓ_{cro} and model $f \in \mathcal{F}$ with M training samples as follows:

$$\begin{aligned} &\mathcal{R}_M(\ell_{cro} \circ \mathcal{F}) \\ =&\mathbb{E}_{(x,c,\mu)}\left[\sup_{f\in\mathcal{F}} \sum_{t=1}^{M} \mu_t\Big(\ell_{cro}\big(f(x_t), c_t\big)\Big)\right] \\ \leq&\sqrt{2}\rho \sum_{c=1}^{C} \mathcal{R}_M(\mathcal{H}_c), \end{aligned} \tag{16}$$

where $\circ$ denotes the function composition operator, $\mathbb{E}_{(x,c,\mu)}$ denotes the expectation over x, c, and μ, μ denotes the Rademacher variable, $\sup_{f\in\mathcal{F}}$ denotes the supremum (or least upper bound) over the function set $\mathcal{F}$ of model f. The second line holds because of the Rademacher vector contraction inequality [34].

Then, we proceed with the proof by showing that the one direction $\sup_{f\in\mathcal{F}} R(f) - \widehat{R}(f)$ is bounded with probability at least $1-\delta/2$, and the other direction $\sup_{f\in\mathcal{F}} \widehat{R}(f) - R(f)$ can be proved similarly. Note that replacing a sample (x_t, c_t) leads to a change of $\sup_{f\in\mathcal{F}} R(f) - \widehat{R}(f)$ at most $\frac{B}{M}$ due to the fact that ℓ_{cro} is bounded by B. According to the McDiarmid's inequality [19], for any $\delta > 0$, with probability at least $1 - \delta/2$, we have:

$$\sup_{f\in\mathcal{F}} R(f) - \widehat{R}(f) \leq \mathbb{E}\left[\sup_{f\in\mathcal{F}} R(f) - \widehat{R}(f)\right] + B\sqrt{\frac{\log \frac{2}{\delta}}{2M}}. \tag{17}$$

According to the result in [19] that shows $\mathbb{E}\left[\sup_{f\in\mathcal{F}} R(f) - \widehat{R}(f)\right] \leq 2\mathcal{R}_M(\mathcal{F})$, and further considering the other direction $\sup_{f\in\mathcal{F}} \widehat{R}(f) - R(f)$, with probability at least $1 - \delta$, we have:

$$\sup_{f\in\mathcal{F}} |R(f) - \widehat{R}(f)| \leq 2\sqrt{2}\rho \sum_{c=1}^{C} \mathcal{R}_M(\mathcal{H}_c) + B\sqrt{\frac{\log \frac{2}{\delta}}{2M}}, \tag{18}$$

which completes the proof.

□

In unsupervised learning, we cannot minimize the empirical risk $\widehat{R}(f)$ directly, since the true labels are inaccessible for $\mathcal{D}_L$. Therefore, we need to train the model with $\widehat{R}'(f) = \frac{1}{M}\sum_{t=1}^{M} \ell_{cro}(f(x_t), p_t) \geq \widehat{R}(f)$, where p_t denotes the pseudo-label of unlabeled sample x_t. Let $\hat{f} = \operatorname{argmin}_{f\in\mathcal{F}} \widehat{R}(f)$ be the empirical risk minimizer, and $f^* = \operatorname{argmin}_{f\in\mathcal{F}} R(f)$ be the true risk minimizer. Let $\ell_{all} = \frac{1}{M}\sum_{t=1}^{M} \ell_{cro}(f(x_t), p_t)$ be the training loss on M pseudo-labeled samples with pseudo-labeling error rate ϵ for our method.

Then, we can bound the difference I_{ϵ_s} between $\widehat{R}(f)$ and $\widehat{R}'(f)$ as follows.

Lemma 2. *Suppose that the loss function $\ell_{cro}(f(x), c)$ is ρ-Lipschitz with respect to $f(x)$ for all $c \in \{1, \ldots, C\}$ and upper-bounded by B. Given the pseudo-labeling error rate $0 < \epsilon < 1$, we have:*

$$0 \leq |\widehat{R}'(f) - \widehat{R}(f)| \leq B\epsilon. \tag{19}$$

Proof. Let's first expand $\widehat{R}'(f)$:

$$\begin{aligned}\widehat{R}'(f) &= \frac{1}{M}\sum_{t=1}^{M}\ell_{cro}(f(x_t),p_t)\\ &= \frac{1}{M}\sum_{t=1}^{M}\left[(1-\epsilon)\ell_{cro}(f(x_t),c_t)+\frac{\epsilon}{C-1}\sum_{c=1}^{C}\mathbb{I}(p_t\neq c_t)\ell_{cro}(f(x_t),p_t)\right].\end{aligned} \tag{20}$$

Next, we show its upper bound:

$$\begin{aligned}\widehat{R}'(f) &\leq \widehat{R}(f)+\frac{1}{M}\sum_{t=1}^{M}\left[\frac{\epsilon}{C-1}\sum_{c=1}^{C}\mathbb{I}(p_t\neq c_t)\ell_{cro}(f(x_t),c_t)\right]\\ &\leq \widehat{R}(f)+\frac{1}{M}\sum_{t=1}^{M}\left[\frac{\epsilon}{C-1}\sum_{c=1}^{C}\mathbb{I}(\hat{y}_j\neq y_j)B\right]\\ &\leq \widehat{R}(f)+B\epsilon,\end{aligned} \tag{21}$$

and when $\epsilon\to 0$, it reaches the lower bound $\widehat{R}(f)$, which concludes the proof.

□

Based on the above lemmas, for any $\delta>0$, with probability at least $1-\delta$, at the s-th training step, we have:

$$\begin{aligned}R_s(\hat{f}) &\leq \widehat{R}_s(\hat{f})+2\sqrt{2}\rho\sum_{c=1}^{C}\mathcal{R}_{M_s}(\mathcal{H}_c)+B\sqrt{\frac{\log\frac{2}{\delta}}{2M_s}}\\ &\leq \widehat{R}'_s(\hat{f})+2\sqrt{2}\rho\sum_{c=1}^{C}\mathcal{R}_{M_s}(\mathcal{H}_c)+B\sqrt{\frac{\log\frac{2}{\delta}}{2M_s}}\\ &\leq \widehat{R}'_s(f)+2\sqrt{2}\rho\sum_{c=1}^{C}\mathcal{R}_{M_s}(\mathcal{H}_c)+B\sqrt{\frac{\log\frac{2}{\delta}}{2M_s}}\\ &\leq \widehat{R}_s(f)+B\epsilon+2\sqrt{2}\rho\sum_{c=1}^{C}\mathcal{R}_{M_s}(\mathcal{H}_c)+B\sqrt{\frac{\log\frac{2}{\delta}}{2M_s}}\\ &\leq R_s(f)+B\epsilon+4\sqrt{2}\rho\sum_{c=1}^{C}\mathcal{R}_{M_s}(\mathcal{H}_c)+2B\sqrt{\frac{\log\frac{2}{\delta}}{2M_s}},\end{aligned} \tag{22}$$

where the first and last lines are based on Lemma 1, and the fourth lines are due to Lemma 2. The third line is by the definition of $\hat{f}$.

Then, we can derive $I_{\epsilon_s}+I_{M_s}$ as follows:

$$\begin{aligned}I_{\epsilon_s}+I_{M_s} &= |\widehat{R}'_s(f)-\widehat{R}_s(f)|+|R_s(f)-\widehat{R}_s(f)|\\ &= \widehat{R}'_s(f)-\widehat{R}_s(f)-R_s(f)+\widehat{R}_s(f)\\ &= \widehat{R}'_s(f)-R_s(f)\\ &\leq R_s(\hat{f})-R_s(f)\\ &\leq B\epsilon_t+4\sqrt{2}\rho\sum_{c=1}^{C}\mathcal{R}_{M_s}(\mathcal{H}_c)+2B\sqrt{\frac{\log\frac{2}{\delta}}{2M_s}},\end{aligned} \tag{23}$$

Substituting Eq. (23) into Eq. (13) yields:

$$\begin{cases} R_s \leq R_{s-1} - G_s + B\epsilon_s + 4\sqrt{2}\rho \sum_{c=1}^{C} \mathcal{R}_{M_s}(\mathcal{H}_c) + 2B\sqrt{\frac{\log\frac{2}{\delta}}{2M_s}}, \\ R_{s-1} \leq R_{s-2} - G_{s-1} + B\epsilon_{s-1} + 4\sqrt{2}\rho \sum_{c=1}^{C} \mathcal{R}_{M_{s-1}}(\mathcal{H}_c) + 2B\sqrt{\frac{\log\frac{2}{\delta}}{2M_{s-1}}}, \\ \quad \vdots \\ R_1 \leq R_0 - G_1 + B\epsilon_1 + 4\sqrt{2}\rho \sum_{c=1}^{C} \mathcal{R}_{M_1}(\mathcal{H}_c) + 2B\sqrt{\frac{\log\frac{2}{\delta}}{2M_1}}. \end{cases} \tag{24}$$

Finally, combining the sub-equations in Eq. (24) through summation yields:

$$R_S \leq R_0 - \sum_{s=1}^{S} G_s + B\sum_{s=1}^{S} \epsilon_s + 4\sqrt{2}\rho \sum_{s=1}^{S}\sum_{c=1}^{C} \mathcal{R}_{M_s}(\mathcal{H}_c) + 2B\sum_{s=1}^{S}\sqrt{\frac{\log\frac{2}{\delta}}{2M_s}}. \tag{25}$$

At this point, we have proven Theorem 1.

□

C Dataset and Learning Paradigm Details

C.1 Dataset Details

Experiments are conducted on six image classification datasets from distinct domains, including DTD [20], FGVC-Aircraft [21], CUB [22], Flowers102 [23], OxfordPets [24], and StanfordCars [25]. We present dataset statistics in Table 6. All datasets adopt the default training-test splits.

Table 6: Dataset statistics.

	DTD	FGVC-Aircraft	CUB	Flowers102	OxfordPets	StanfordCars
Class number	47	100	200	102	37	196
Training set size	3760	6667	5594	2040	3680	8144
Testing set size	1880	3333	5794	6149	3669	8041
Balanced	✓	✓	✗	✗	✗	✗

C.2 Learning Paradigm Details

In main results (ξ 4.2), we consider three learning paradigms, including unsupervised learning (UL), semi-supervised learning (SSL), and transductive zero-shot learning (TRZSL). We provide a detailed explanation of experimental settings for three learning paradigms, in line with [9].

- For Unsupervised Learning (UL), we only have access to unlabeled data. In this scenario, we initially rely on the zero-shot predictions of CLIP to obtain all pseudolabels without any manual annotation.
- For Semi-Supervised Learning (SSL), access to labeled data is limited. We assess the impact of pseudolabels in scenarios with a few labeled data and abundant unlabeled data, using two labeled samples per class.
- For Transductive Zero-Shot Learning (TRZSL), labeled data for certain target classes (seen classes) are provided in the downstream dataset. We set the ratio of seen to unseen classes at 62-38, with all pseudolabels generated from unseenclasses. It is noteworthy that in TRZSL, we report the harmonic mean of the accuracies of seen and unseen classes.

D Comparison Methods

We compare our method with eight SOTA methods, including CLIP [1], Few Pseudo-Labels (FPL [9]), Grow and Reffne Iteratively Pseudo-labels (GRIP [9]), Candidate Pseudo-Labeling with FPL (CPL-F [7]), Candidate Pseudo-Labeling with GRIP (CPL-G [7]), Universal Entropy Optimization (UEO [32]), Augmenting Discriminative Richness (AiR [10]), and Concept-Adaptive Pseudo-labeling (CAP [8]). The details of each method are presented as follows:

- **CLIP :** CLIP exhibits zero-shot capability for classification.
- **FPL :** FPL filters top-k confident pseudo-labels of each class to construct balanced pseudo-label dataset for model training.
- **GRIP :** GRIP pre-defines the iteration times. Within each iteration, it regenerates pseudo-labels and filters top-k confident pseudo-labels of each class to construct balanced pseudo-label dataset, where k increases linearly as iterations proceed.
- **CPL:** CPL uses intra- and inter-instance selected candidate pseudo-labels instead of hard pseudo-labels, effectively reducing label noise and class imbalance for better adaptation. **CPL-F** and **CPL-G** are implemented on the basis of FPL and GRIP, respectively.
- **UEO:** UEO is a universal entropy optimization method for realistic unsupervised CLIP fine-tuning. It uses sample confidence to minimize entropy for in-distribution samples and maximize marginal entropy for OOD samples.
- **AiR:** AiR utilizes LoRA-fine-tuned diffusion models to generate class-representative synthetic images and builds an auxiliary classifier fused with text classifier for high-quality pseudo-labels.
- **CAP:** CAP proposes concept alignment to fix semantic-visual misalignment, and a confusion-aware calibrated margin to improve discrimination between similar classes. It also adopts dual independent adapters to separately learn balanced pseudo-labels and high confidence unlabeled data.

E Implementation Details

We present implementation details in Table 7. Other methods follow the implementation in the original paper.

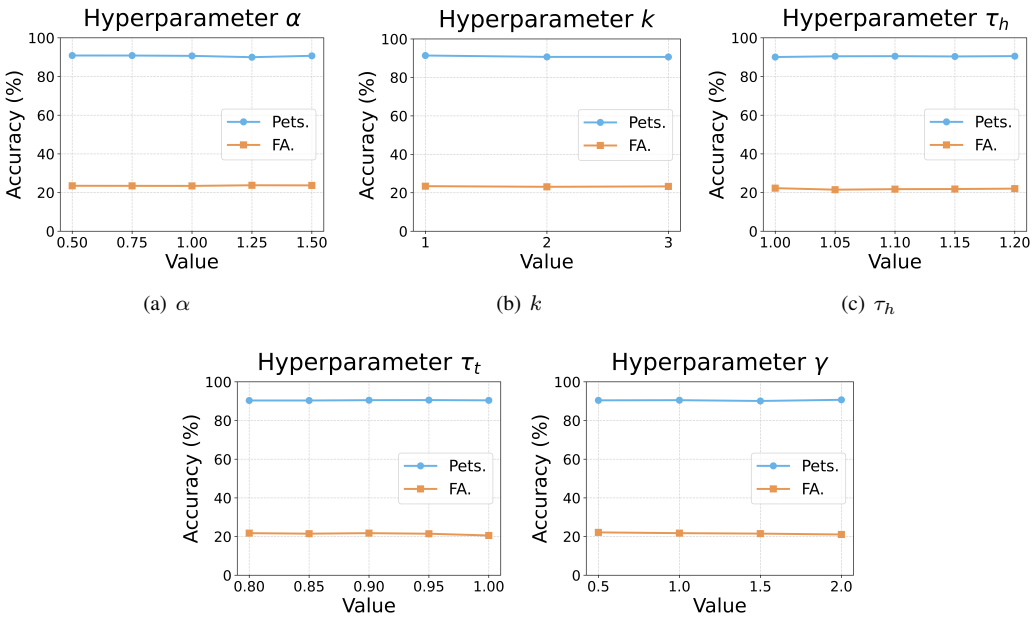
Table 7: Implementation Details

Training Setting		**Hyperparameters**	
Network	ViT-B / 32	Pseudo-head class threshold τ_h	1.1
Batch size	32	Pseudo-tail class threshold τ_t	0.9
Epoch	150	γ in Eq. 3	1
Optimizer	SGD	k in Eq. 5	2
Learning rate (LR)	0.01	α in Eq. 8	1
Warm epoch	2		
Warmup Learning rate	1e-4		
Weight decay	0.05		
LR scheduler	CosineAnnealingLR		
LLMs	Qwen-plus		

F More Experiment Results and Analysis

F.1 Sensitivity Study of the Hyperparameters

We conduct a sensitivity study to explore the impact of hyperparameters α, k, τ_h, τ_t, and γ on the datasets Pets. and FA. . Fig. 8 shows that our method is insensitive to these hyperparameters, further demonstrating the robustness of our method.



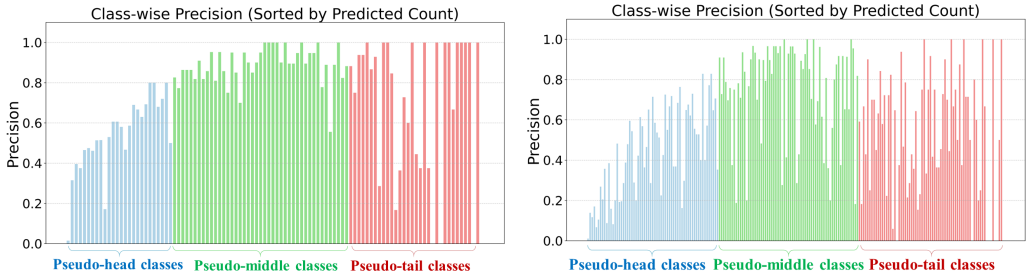
(a) α (b) k (c) τ_h

(d) τ_t (e) γ

Figure 8: Sensitivity analysis of α, k, τ_h, τ_t, and γ.

F.2 Precision of Pseudo-labels

As shown in Fig. 9, we observe that pseudo-middle classes achieve higher precision than pseudo-head classes. Therefore, the confidence scores of pseudo-middle labels are more reliable than pseudo-head labels and we filter them via confidence scores directly without neighborhood consistency filtering mechanism.



(a) Flowers102 (b) CUB

Figure 9: Precision of pseudo-labels across different classes which sorted by pseudo-label quantity.

F.3 Comparison of Methods Fine-tuned with VPT

We fine-tune all comparison methods with VPT [18] except CAP [8], due to its results with MaPLe [33] already shown in Table 1. As shown in Table 8, our method PPSB also achieves competitive performance on six datasets under three learning paradigms, exceeding SOTA baselines by average **3.70**% in accuracy. This demonstrates that PPSB can adapt to different fine-tuning methods.

Table 8: Test accuracy (%) of methods fine-tuned with VPT across six datasets under three learning paradigms. Values are expressed as mean ± standard deviation. The highest accuracies are bold.

Methods	Flowers102			FGVCAircraft			CUB			
	SSL	UL	TRZSL	SSL	UL	TRZSL	SSL	UL	TRZSL	Avg.
FPL [9]	$70.22_{0.26}$	$67.50_{0.23}$	$79.02_{0.54}$	$20.84_{0.32}$	$19.35_{0.27}$	$26.00_{0.35}$	$52.78_{0.19}$	$53.25_{0.22}$	$56.11_{0.15}$	$49.45_{0.28}$
GRIP [9]	$71.68_{0.09}$	$63.65_{0.28}$	$78.52_{0.75}$	$19.90_{0.22}$	$18.12_{0.34}$	$30.56_{0.29}$	$49.72_{0.49}$	$50.72_{0.30}$	$56.02_{0.14}$	$48.77_{0.32}$
CPL-F [7]	$70.16_{0.07}$	$69.20_{0.24}$	$78.85_{0.22}$	$21.43_{0.59}$	$19.88_{0.11}$	$26.87_{0.41}$	$53.49_{0.48}$	$53.72_{0.40}$	$56.29_{0.39}$	$49.99_{0.32}$
CPL-G [7]	$72.72_{0.45}$	$67.89_{1.48}$	$79.81_{0.43}$	$19.72_{0.24}$	$18.47_{0.71}$	$\mathbf{31.08}_{0.47}$	$49.64_{0.40}$	$51.55_{0.20}$	$57.00_{0.61}$	$49.76_{0.55}$
UEO [32]	$54.11_{2.19}$	$56.28_{2.31}$	$49.01_{4.13}$	$18.97_{0.16}$	$18.95_{0.47}$	$18.48_{0.26}$	$49.88_{0.77}$	$50.80_{0.21}$	$48.03_{0.84}$	$40.50_{1.26}$
AiR [10]	$72.02_{0.39}$	$63.96_{0.15}$	$78.25_{0.51}$	$20.70_{0.32}$	$18.00_{0.12}$	$29.62_{0.42}$	$48.83_{0.46}$	$49.73_{0.21}$	$56.27_{0.37}$	$48.60_{0.33}$
PPSB (Ours)	$\mathbf{79.30}_{1.91}$	$\mathbf{73.55}_{0.37}$	$\mathbf{82.44}_{0.05}$	$\mathbf{23.77}_{0.51}$	$\mathbf{22.61}_{0.26}$	$30.08_{0.10}$	$\mathbf{56.26}_{0.04}$	$\mathbf{57.55}_{0.16}$	$\mathbf{57.96}_{0.23}$	$\mathbf{53.72}_{0.40}$
Methods	DTD			OxfordPets			StanfordCars			
	SSL	UL	TRZSL	SSL	UL	TRZSL	SSL	UL	TRZSL	Avg.
FPL [9]	$51.63_{0.58}$	$47.62_{0.32}$	$55.06_{0.61}$	$89.14_{0.21}$	$89.12_{0.14}$	$89.38_{0.10}$	$59.54_{0.28}$	$59.92_{0.39}$	$63.66_{0.47}$	$67.23_{0.34}$
GRIP [9]	$57.73_{0.73}$	$48.91_{0.23}$	$64.12_{0.71}$	$89.65_{0.08}$	$89.94_{0.52}$	$90.53_{0.07}$	$54.96_{0.19}$	$54.67_{0.33}$	$63.77_{0.25}$	$68.25_{0.35}$
CPL-F [7]	$52.62_{0.69}$	$51.06_{0.37}$	$55.72_{0.35}$	$88.51_{0.35}$	$87.83_{0.01}$	$87.22_{0.12}$	$59.32_{0.21}$	$59.55_{0.06}$	$63.27_{0.26}$	$67.23_{0.27}$
CPL-G [7]	$55.85_{0.16}$	$52.39_{0.13}$	$64.35_{0.31}$	$89.12_{0.26}$	$89.94_{0.19}$	$89.79_{0.56}$	$54.97_{0.16}$	$55.94_{0.20}$	$63.77_{0.05}$	$68.46_{0.22}$
UEO [32]	$41.18_{0.62}$	$42.39_{0.08}$	$38.63_{1.75}$	$84.70_{0.31}$	$85.51_{0.47}$	$82.73_{0.13}$	$55.58_{0.39}$	$55.55_{0.14}$	$54.26_{0.56}$	$60.06_{0.49}$
AiR [10]	$57.30_{0.85}$	$49.41_{0.27}$	$63.71_{0.21}$	$90.13_{0.31}$	$89.72_{0.50}$	$90.69_{0.20}$	$55.59_{0.27}$	$53.71_{0.47}$	$\mathbf{64.24}_{0.76}$	$68.28_{0.43}$
PPSB (Ours)	$\mathbf{59.59}_{0.70}$	$\mathbf{56.38}_{0.63}$	$\mathbf{65.51}_{1.47}$	$\mathbf{91.42}_{0.16}$	$\mathbf{91.51}_{0.11}$	$\mathbf{90.95}_{0.31}$	$\mathbf{62.85}_{0.41}$	$\mathbf{64.26}_{0.28}$	$64.11_{0.20}$	$\mathbf{71.84}_{0.47}$

F.4 Evolution of Pseudo-label Predictions

Fig. 10 shows evolution of pseudo-label predictions on the dataset Flowers102. It can be observed that our method generates pseudo-labels gradually approach balanced distribution.

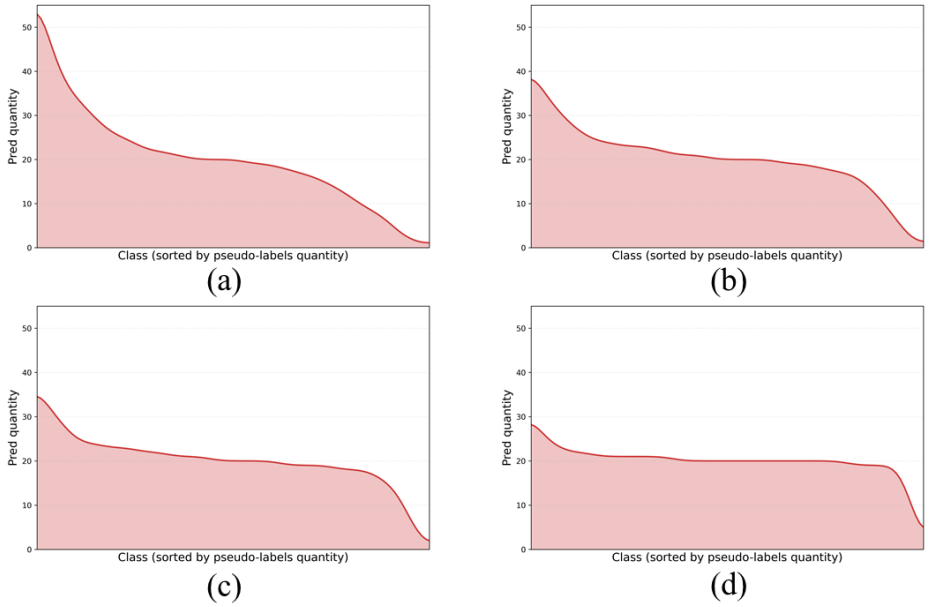


Figure 10: Evolution of Pseudo-label Predictions on the dataset Flowers102. (a)-(d) show that pseudo-label distribution approaches balanced gradually.

F.5 Evolution of Pseudo-labeled Dataset

Fig. 11 shows evolution of pseudo-label dataset on DTD. It can be observed that our method consistently construct relatively balanced pseudo-label dataset and the scale expands progressively.

F.6 Statistical Significance Test

Table 9 presents the win/tie/loss counts between our PPSB and eight baseline methods across different datasets under three settings with a pairwise t-test at a 0.05 significance level. It can be observed that

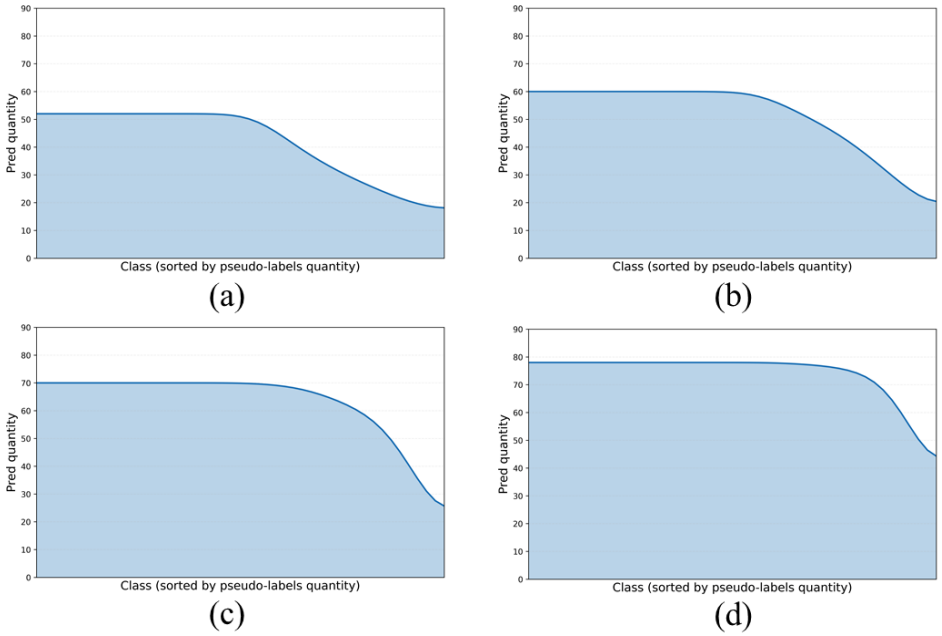
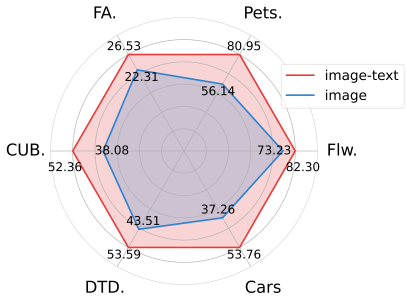


Figure 11: Evolution of Pseudo-label dataset on DTD. (a)-(d) show that pseudo-label dataset consistently remains relatively balanced and expands progressively.

Table 9: Statistical significance of performance differences assessed with pairwise t-test at a 0.05 significance level, reported as win/tie/loss counts.

Method	SSL	UL	TRZSL	Total
CLIP	6 / 0 / 0	6 / 0 / 0	6 / 0 / 0	18 / 0 / 0
FPL	6 / 0 / 0	6 / 0 / 0	6 / 0 / 0	18 / 0 / 0
GRIP	6 / 0 / 0	6 / 0 / 0	5 / 1 / 0	17 / 1 / 0
CPL-F	6 / 0 / 0	6 / 0 / 0	6 / 0 / 0	18 / 0 / 0
CPL-G	6 / 0 / 0	6 / 0 / 0	5 / 1 / 0	17 / 1 / 0
UEO	6 / 0 / 0	6 / 0 / 0	6 / 0 / 0	18 / 0 / 0
CAP	5 / 1 / 0	5 / 1 / 0	5 / 1 / 0	15 / 3 / 0
AiR	6 / 0 / 0	6 / 0 / 0	6 / 0 / 0	18 / 0 / 0
Total	47 / 1 / 0	47 / 1 / 0	45 / 3 / 0	139 / 5 / 0

Figure 12: Perform k-means on the image-text representation matrix $\mathcal{W}$ and image representation matrix $\mathcal{X}$ across six datasets. $\mathcal{W}$ consistently achieve higher clustering accuracy.



our method significantly outperforms baselines in 96.52 pp of cases (139 out of 144), demonstrating its effectiveness.

F.7 Clustering Performance of the Image-text Representation Matrix

We perform k-means on the image-text representation matrix $\mathbf{W}$ and image representation matrix $\mathbf{X}$ across six datasets. As shown in Fig. 12, $\mathbf{W}$ consistently achieves higher clustering performance than $\mathbf{X}$, which demonstrates that $\mathbf{W}$ exhibits more discriminative neighborhood information.

F.8 Comparison of Soft and Hard Pseudo-labels

Soft pseudo-labels preserve prediction uncertainty by retaining the full predicted class distribution. However, their reliability depends in part on model calibration, and non-maximum probabilities may encode erroneous correlations and propagate pseudo-label bias [7]. Hard pseudo-labels discard these probabilities and may therefore reduce the influence of unreliable class associations.

To compare the two alternatives, we replace hard pseudo-labels with soft pseudo-labels while keeping all other settings unchanged. As shown in Table 10, hard pseudo-labels achieve higher test accuracy across all six datasets, increasing the average accuracy from 58.50% to 63.37%, a gain of **4.87** percentage points. These results support the use of hard pseudo-labels as a more effective source of supervision for our method under the evaluated settings.

Table 10: Comparison of test accuracy (%) using soft and hard pseudo-labels. Results are reported as mean ± standard deviation, except for the average column. The highest accuracies are bold.

Pseudo-labels	Flw.	FA.	CUB.	DTD.	Pets.	Cars.	Avg.
Soft	76.99 ± 0.22	22.83 ± 0.30	55.20 ± 0.08	47.03 ± 0.39	88.33 ± 0.64	60.64 ± 0.08	58.50
Hard	$\mathbf{86.43} \pm 0.81$	$\mathbf{23.23} \pm 0.30$	$\mathbf{59.80} \pm 0.29$	$\mathbf{57.26} \pm 0.41$	$\mathbf{91.26} \pm 0.03$	$\mathbf{62.23} \pm 0.36$	**63.37**

F.9 Evaluation with Stronger Vision-Language Models

We use CLIP as the primary backbone to enable direct comparisons with existing unsupervised VLM adaptation methods. However, PPSB operates on image-text representations and pseudo-label distributions rather than architecture-specific components, allowing it to be applied to other stronger VLM backbones. To evaluate the generality, we conduct additional experiments using SigLIP-B/16 and PE-Core-B/16 on four datasets.

As shown in Tables 11 and 12, our method achieves the highest test accuracy on all four datasets with both backbones. On SigLIP-B/16, PPSB achieves an average accuracy of 73.81%, exceeding the strongest baseline by **1.62** percentage points. On PE-Core-B/16, its average accuracy reaches 82.82%, outperforming CAP by **4.99** percentage points. These results demonstrate the generality of PPSB across stronger vision-language models.

Table 11: Comparison results of test accuracy (%) on SigLIP-B/16.

Method	FA.	DTD.	Pets.	Cars.	Avg.
FPL [9]	37.41	65.85	92.50	88.22	71.00
GRIP [9]	27.90	65.53	92.96	82.14	67.13
CPL-F [7]	37.56	67.07	92.45	79.65	69.18
CPL-G [7]	31.44	66.17	90.92	76.14	66.17
UEO [32]	13.32	60.53	88.36	78.64	60.21
CAP [8]	39.00	68.13	93.64	87.99	72.19
AiR [10]	27.24	65.10	92.20	82.51	66.76
PPSB (Ours)	**41.55**	**70.21**	**94.23**	**89.23**	**73.81**

Table 12: Comparison results of test accuracy (%) on PE-Core-B/16.

Method	FA.	DTD.	Pets.	Cars.	Avg.
FPL [9]	56.64	66.86	94.16	92.26	77.48
GRIP [9]	49.68	68.61	93.56	89.81	75.42
CPL-F [7]	55.95	67.23	94.14	90.16	76.87
CPL-G [7]	51.78	68.35	93.81	89.31	75.81
UEO [32]	46.86	68.24	94.19	84.65	73.49
CAP [8]	53.23	72.12	94.76	91.22	77.83
AiR [10]	48.54	69.41	93.29	90.33	75.39
PPSB (Ours)	**68.34**	**73.77**	**95.33**	**93.84**	**82.82**

G Limitations

Although the proposed progressive pseudo-label self-balancing framework effectively avoids numerous incorrect pseudo-tail labels, further improving the accuracy of pseudo-tail classes remains a challenging task. In the future work, we aim to refine our framework to better address this issue.

H Compute Resources

- CPU: AMD EPYC 7642 48-Core Processor × 2
- GPU: NVIDIA GeForce RTX 4090 24G × 1
- MEM: 500G
- Maximum total computing time per dataset: training + testing ≈ 30min.

I Broader Impacts

This paper presents work whose goal is to advance the field of Machine Learning. Specifically, we divide all classes into pseudo-head, pseudo-middle, and pseudo-tail classes based on pseudo-label

quantity. Accordingly, we propose a progressive pseudo-label self-balancing framework, which measures the class-wise prediction bias and calculates quantity upper bounds for certain classes to construct relatively balanced pseudo-labeled dataset. Furthermore, we design an image-text neighborhood filtering mechanism and a visual classifier to generate high-quality pseudo-labels. We also theoretically prove that our method can significantly reduce the generalization error under certain conditions. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

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 main claims made in the abstract and introduction accurately reflect the paper's contributions and scope. Please see Abstract and Sec. 1.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: The paper discusses the limitations of the work performed by the authors. Please see Appendix G.

   Guidelines:

   - The answer [N/A] 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: For each theoretical result, the paper provides the full set of assumptions and a complete (and correct) proof. Please see Sec. 3.5 and Appendix B.

Guidelines:

- The answer [N/A] 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: The paper fully discloses 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). Please see Sec. 4.1, Appendix C, and Appendix E.

Guidelines:

- The answer [N/A] 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: [Yes]

   Justification: The paper provides open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results. Please see Sec. 4.1, Appendix C, and Appendix E.

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: The paper reports error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments. Please see Sec. F.6.

   Guidelines:

   - The answer [N/A] 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: The paper provides sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments. Please see Appendix H.

   Guidelines:

   - The answer [N/A] 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: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines.

   Guidelines:

   - The answer [N/A] 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: [Yes]

    Justification: The paper discusses both potential positive societal impacts and negative societal impacts of the work performed. Please see Appendix I.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our research does not release data or models, so the paper has no such risks.

    Guidelines:

    - The answer [N/A] 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: The creators or original owners of assets (e.g., code, data, models), used in the paper, are properly credited and are the license and terms of use explicitly mentioned and properly respected.

    Guidelines:

    - The answer [N/A] 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: [Yes]

    Justification: We publicly release the source code of PPSB at `https://github.com/siqin-seu/PPSB-NeurIPS-26`. The repository provides documentation on environment setup, dataset preparation, and instructions for reproducing the experimental results.

    - The answer [N/A] 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: [N/A]

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

    Guidelines:

    - The answer [N/A] 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: [N/A]

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

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.