Candidate Pseudolabel Learning: Enhancing Vision-Language Models by Prompt Tuning with Unlabeled Data

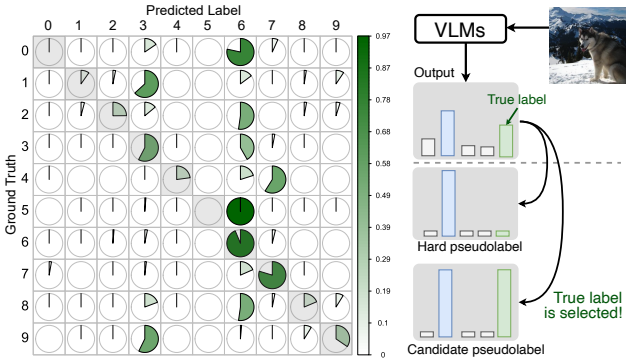
Jiahan Zhang [*1] **Qi Wei** [*2] **Feng Liu** [3] **Lei Feng** [1]

Abstract

Fine-tuning vision-language models (VLMs) with abundant unlabeled data recently has attracted increasing attention. Existing methods that resort to the pseudolabeling strategy would suffer from heavily incorrect hard pseudolabels when VLMs exhibit low zero-shot performance in downstream tasks. To alleviate this issue, we propose a **C**andidate **P**seudolabel **L**earning method, termed **CPL**, to fine-tune VLMs with suitable candidate pseudolabels of unlabeled data in downstream tasks. The core of our method lies in the generation strategy of candidate pseudolabels, which progressively generates refined candidate pseudolabels by both intra- and inter-instance label selection, based on a confidence score matrix for all unlabeled data. This strategy can result in better performance in true label inclusion and class-balanced instance selection. In this way, we can directly apply existing loss functions to learn with generated candidate psueudolabels. Extensive experiments on nine benchmark datasets with three learning paradigms demonstrate the effectiveness of our method. Our code can be found here.

1. Introduction

Recent studies in large pre-trained vision-language models (VLMs) (Radford et al., 2021; Li et al., 2022; Yuan et al., 2021) have demonstrated promising zero-shot performance. Nonetheless, previous research (Zhou et al., 2022a;b; Zhang et al., 2024) indicated that substantial labeled data is still necessary to further improve the performance of VLMs for adaptation on various downstream tasks. This requirement for adaptation would cause considerable labeling costs, as labeled data is hard to obtain.



(a) (b)

Figure 1: (a) Confusion matrix between true labels and hard pseudolabels of dataset EuroSAT, where incorrect and imbalanced pseudolabels are always generated. (b) An example illustration of a set of candidate pseudolabels, which consists of classes with the top-2 highest confidence scores.

In response to this challenge, recent studies (Menghini et al., 2023; Huang et al., 2022; Lai et al., 2023; Tanwisuth et al., 2023; Shu et al., 2022) have shifted their focus towards scenarios with abundant unlabeled data, aiming to exploit the inherent zero-shot ability of VLMs. These studies can reduce the dependency on labeled data for adapting VLMs to downstream tasks. Besides, previous studies also showed that fine-tuning VLMs with pseudolabels generated by the zero-shot ability of VLMs is an effective approach for exploiting unlabeled data (Huang et al., 2022; Menghini et al., 2023; Mirza et al., 2023). However, the performance of existing methods heavily relies on the accuracy of the generated hard pseudolabels. When VLMs exhibit diminished zero-shot abilities in certain downstream tasks, the performance of these methods would significantly deteriorate.

To illustrate this issue, we conducted a pilot experiment to empirically demonstrate the deficiency of hard pseudolabels. Specifically, we leveraged CLIP (Radford et al., 2021) to generate hard pseudolabels on the EuroSAT dataset (Helber et al., 2019) and subsequently calculated the confusion matrix between the ground-truth labels and predicted class labels. The results, as shown in Figure 1(a), reveal that a

[*]Equal contribution [1]Singapore University of Technology and Design, Singapore [2]Nanyang Technological University, Singapore [3]University of Melbourne, Australia. Correspondence to: Lei Feng <feng_lei@sutd.edu.sg>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

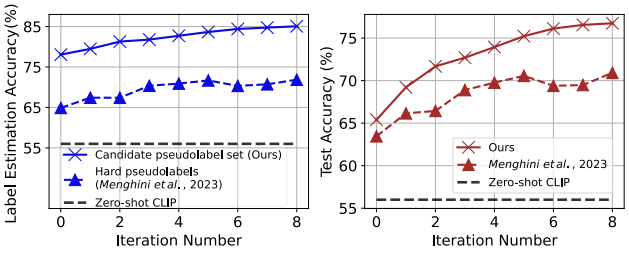


Figure 2: Our candidate pseudolabel learning (CPL) significantly surpasses hard pseudolabel learning (Menghini et al., 2023) on the RESISC45 dataset in terms of label estimation accuracy (label estimation accuracy is defined as the rate at which the true label is included in the pseudolabels), leading to improved performance on test accuracy.

substantial number of samples from other categories are incorrectly predicted as class 6, while fewer samples are classified into classes 0 and 5. Fine-tuning VLMs on such a training set with a significant number of incorrect and imbalanced pseudolabels inevitably leads to inferior performance.

Motivated by the idea of multiple annotations in crowdsourcing (Hossain & Kauranen, 2015; Li et al., 2023), we aim to provide *a set of pseudolabels* that can potentially be the true label (dubbed *candidate pseudolabels* in this paper), instead of merely considering a single hard pseudolabel. To form the set of candidate pseudolabels for each instance, we select the classes with the top-k highest prediction confidences, which would probably contain the true label even though the (top-1) prediction is incorrect (see Figure 1(b)). We also demonstrate the advantages of using candidate pseudolabels in Figure 2, where the blue lines represent the true label estimation accuracy of candidate pseudolabel learning and hard pseudolabel learning. On the other side, the red lines represent the trend of test accuracy for both methods throughout the iterations. From Figure 2, we can find that with training proceeds, the true label is given by the candidate pseudolabel with a probability of almost 85%, while it is around 70% for hard pseudolabeling. In addition, we can also find that the test accuracy of the model can be continually improved as the iteration increases.

In this paper, we propose a Candidate Pseudolabel Learning (CPL) method to fine-tune VLMs with suitable candidate pseudolabels of unlabeled data in downstream tasks. The core of our CPL framework lies in the generation strategy of candidate pseudolabels, which needs to progressively generate refined candidate pseudolabels during the fine-tuning process. Specifically, we construct a confidence score matrix encompassing all unlabeled data. Based on this matrix, we take into account two aspects, including intra- and inter-instance label selection. The simultaneous consideration of the two aspects can ensure that the true label is selected in the set of candidate pseudolabels to a large extent. Meantime, it can effectively mitigate the overwhelming influence that dominant classes may exert on the generation of pseudolabels, thereby ensuring a balanced and accurate representation of classes.

Based on this novel pseudolabel structure, we transform the multiclass classification problem into the problem of learning with multiple candidate labels (Luo & Orabona, 2010; Cour et al., 2011). In this way, we can fine-tune VLMs with candidate pseudolabels using popular loss functions for learning with multiple candidate labels (Feng et al., 2020; Wen et al., 2021; Zhang et al., 2021b). In our CPL method, the fine-tuning of the model and the update of candidate pseudolabels are conducted iteratively, mutually benefiting each other. We conduct extensive experiments across three learning paradigms (unsupervised learning, semi-supervised learning, and transductive zero-shot learning) and two prompt-tuning paradigms (textual and visual). Experimental results demonstrate that our proposed method consistently achieves state-of-the-art performance.

2. Related Work

2.1. Vison-Language Models

Recently, Vision Language Models, such as CLIP (Radford et al., 2021), ALBEF (Li et al., 2021), BLIP (Li et al., 2022), and Flamingo (Alayrac et al., 2022), pre-trained on large-scale image-text data, have achieved significant success (Zhang et al., 2024). These models are capable of zero-shot image classification. Besides, the performance of VLMs can be further enhanced by fine-tuning with annotated data from downstream datasets. For instance, CoOp (Zhou et al., 2022b) learns prompt vectors by minimizing prediction errors using the cross-entropy loss. Tip-Adapter (Zhang et al., 2021c) employs additional adapter modules for parameter-efficient fine-tuning on downstream datasets. In this paper, we primarily focus on the performance of CLIP, a representative VLM, in downstream tasks.

2.2. Prompt Tuning

Prompt tuning is a technique that can enhance the performance of large pre-trained models in specific downstream tasks through efficient parameter fine-tuning. The common types of prompt tuning include text-based (Zhou et al., 2022b;a; Ge et al., 2023) and visual prompt tuning (Bahng et al., 2022; Jia et al., 2022). Text-based prompt tuning (Zhou et al., 2022b) employs continuous optimization strategies to optimize a set of continuous vectors, thereby eliminating the need for manually designed discrete prompt texts. visual prompt tuning (Jia et al., 2022) offers an efficient alternative to complete fine-tuning of transformer models by introducing a minimal number of trainable parameters in the visual input. In classification tasks, prompt tuning

necessitates training on a small number of labeled examples for each class. In this paper, we mainly explore candidate pseudolabels in visual prompt tuning (Jia et al., 2022) and text prompting tuning (Zhou et al., 2022b) to enhance the performance of VLMs when unlabeled data is available.

2.3. Learning from Unlabeled Data

In real-world applications, we often have access to a substantial amount of unlabeled data for downstream tasks. This motivates us to devise effective methods for utilizing such data. In semi-supervised learning (Sohn et al., 2020; Xu et al., 2021; Wei et al., 2024; Zhang et al., 2021a), pseudolabeling is a widely studied and adopted technique. Grounded in the principle of entropy minimization (Grandvalet & Bengio, 2004), it typically selects the most reliable samples from unlabeled data based on the confidence for inclusion in training. However, this pseudolabeling strategy has been found challenging to apply directly to the zero-shot predictions of VLMs as it struggles to effectively estimate the most accurate samples from unlabeled data (Huang et al., 2022). In previous research on pseudolabeling for VLMs, Huang et al. (2022) initially proposed generating more reliable offline pseudolabels by selecting multiple examples with the highest confidence for each category. On the other hand, Menghini et al. (2023) proposed updating pseudolabels iteratively while still selecting the most reliable samples for each category. In this paper, we propose a novel candidate pseudolabel generation strategy that aims to improve the label estimation accuracy of pseudolabels, thereby enhancing the performance of VLMs when adapting to downstream tasks with unlabeled data.

3. Methodology

Problem Definition. In this paper, our objective is to fine-tune VLMs using downstream unlabeled data drawn from a d-dimensional feature space represented as $\mathcal{X} \subseteq \mathbb{R}^d$. The corresponding label space for all the downstream data is denoted as $\mathcal{Y} = \{1, ..., C\}$, indicating that we are considering a C-class classification problem. Specifically, our focus is on exploiting the inherent zero-shot capability of VLMs to generate pseudolabels for unlabeled data. We consider three commonly encountered learning paradigms associated with abundant unlabeled data, including Semi-supervised Learning (SSL) (Sohn et al., 2020; Cascante-Bonilla et al., 2021), Transductive Zero-shot Learning (TRZSL) (Wan et al., 2019; Gao et al., 2020), and Unsupervised Learning (UL) (Noroozi & Favaro, 2016; Schmarje et al., 2021).

Motivation. For pseudolabel generation, previous methods (Huang et al., 2022; Menghini et al., 2023) rely on confidence ranking to select the most confident samples for each class. However, the utilization of hard pseudolabels may inadvertently amplify the effects of lower prediction accuracy for certain categories. In this paper, we draw inspiration from the concept of the *multiple annotations* in crowdsourcing, constructing a set of potential true labels for model learning. Intuitively, the advantages of candidate pseudolabels over hard pseudolabels can result in *more precise label estimation*, implying that the candidate pseudolabels can better encapsulate the true label within its candidate set. This is illustrated and contrasted in Figure 2.

Overview. The overall workflow of our method can be divided into three steps: ❶ The generation of candidate pseudolabels for unlabeled training data (see Sec. 3.1). Considering both intra- and inter-instance perspectives, we selectively construct the training set with candidate pseudolabels from the unlabeled dataset D_{UL}, while keeping balanced quantities among varying classes. ❷ Based on the candidate pseudolabel, we transform the conventional multi-classification problem into a problem akin to learning in a set of candidate labels (see Sec. 3.2), which has been extensively studied in the field of partial-label learning (Cour et al., 2011). Therefore, any loss function designed for partial-label learning can be employed to update the model's parameters. ❸ We iterate the preceding steps until the model's parameters are optimized for downstream tasks. This iterative process facilitates the progressive refinement and optimization of candidate pseudolabels.

3.1. Scheme for Generating Candidate Pseudolabels

Notations. Given an unlabeled example, the set of candidate pseudolabels is denoted by S. For the unlabeled set $D_{\mathrm{UL}} = \{(\boldsymbol{x}_i)\}_{i=1}^N$ composed of N instances, the pair of the instance and the corresponding set of candidate pseudolabels for D_{UL} is represented by $\{(\boldsymbol{x}_i, S_i)\}_{i=1}^N$. Suppose a VLM with learnable parameters θ is represented by $\boldsymbol{f}_\theta$. Given an instance $\boldsymbol{x}$ in D_{UL}, the output of the model can be denoted by $\boldsymbol{f}_\theta(\boldsymbol{x})$. Then, we can obtain a vector of confidence scores for all classes via the Softmax function $\boldsymbol{g}(\cdot)$, i.e., $\boldsymbol{p} = \boldsymbol{g}(\boldsymbol{f}_\theta(\boldsymbol{x})) = (p_1, p_2, \ldots, p_C)^\top$. For convenience, we represent p_{ic} as the confidence score of the c-th class for the i-th instance. By constructing a confidence score matrix $(\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N)^\top$ for all unlabeled data (as shown in Figure 3), we propose to generate candidate pseudolabels by simultanesouly considering two aspects, including both the intra- and inter-instance label selection, respectively.

Intra-instance Label Selection. The concept of intra-instance label selection originates from the idea of selecting the top-K confident labels as the most probable label candidates for each instance. However, we further consider that it may not be reasonable to select an equal number of top-K confident labels as the candidate set for each instance, because of the varying levels of identification difficulty associated with each instance.

In response to this issue, we propose an adaptive strategy

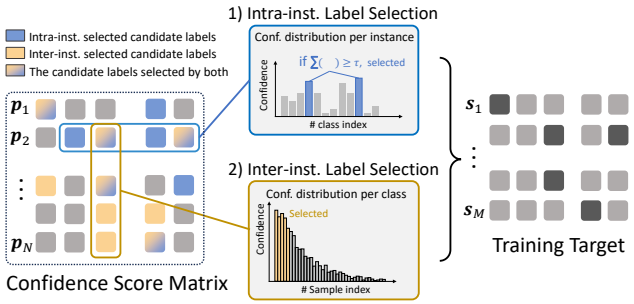


Figure 3: Illustration of the training target generation process in our CPL method. At the beginning of each training iteration, we first construct a confidence score matrix composed of confidence score vector $\boldsymbol{p}$ for each unlabeled instance. Then, candidate pseudolabels, derived from both intra- and inter-level selection, are extracted to formulate the training target $\boldsymbol{s}$ for the subsequent model training.

where different values of K are set for instances with different identification difficulty. Specifically, for each unlabeled instance $\boldsymbol{x}_i$, potential labels are sequentially incorporated into the candidate set S_i^{intra}. This is done in *descending order* based on the corresponding confidence scores until the cumulative confidence score just surpasses a threshold τ. Formally, for an unlabeled instance $\boldsymbol{x}_i$, its selected candidate set S_i^{intra} is represented as

$$S_i^{\text{intra}} = \text{MinSize}(\{c \mid \sum_{c=1}^{C} p_{ic} \geqslant \tau\}), \tag{1}$$

where $\text{MinSize}(\cdot)$ means a function that returns the set with the minimal size and we assume that we select the elements $(p_{i1}, \ldots, p_{iC})$ from the largest to the smallest. The hyper-parameter τ serves as a threshold to ensure that the candidate set S_i^{intra} for each instance $\boldsymbol{x}_i$ encompasses a nearly equivalent level of confidence scores, thereby guaranteeing a comparable level of label estimation accuracy for the candidate pseudolabels of each instance.

As for the determination of the threshold τ, considering the model's average prediction confidence increases with the training process, it is natural to think that the threshold could also be adaptively updated with the model's training. Therefore, we obtain τ from the prediction confidence among D_{UL} and update at the beginning of each training iteration. Specifically, for an unlabeled instance $\boldsymbol{x}_i$, the prediction confidence is represented as $\hat{p}_i = \max_{c\in[C]}(p_{ic})$, which is the maximum value among the confidence scores of instance $\boldsymbol{x}_i$. Subsequently, we can obtain a list of the prediction confidence denoted as $\hat{\boldsymbol{p}} = (\hat{p}_1, \hat{p}_2, ..., \hat{p}_N) \in [0,1]^N$. By sorting $\hat{\boldsymbol{p}}$ in ascending order, τ is determined when the ratio α is given:

$$\tau = \text{Quantile}(\text{Sort}(\hat{\boldsymbol{p}}), \alpha). \tag{2}$$

Here, $\text{Quantile}(\cdot, \alpha)$ is a function that returns the value at the given quantile of the vector and $\text{Sort}(\cdot)$ is a function that sorts the vector in ascending order. α is a hyper-parameter that denotes the specified quantile. It is noteworthy that, based on the formula in Equation (2), setting $\alpha = 0\%$ would result in an extreme case where all the candidates for pseudolabels contain only the most confident label. This is equivalent to the hard pseudolabeling, further demonstrating the flexibility of our candidate set generation strategy.

Inter-instance Label Selection. Since the predictions from CLIP exhibits class imbalance performance across various categories (see Figure 1(a)), the generated single hard pseudolabels are typically class-imbalanced. To balance the ratio of each category in the candidate pseudolabel set and mitigate the overwhelming influence of dominant classes in pseudolabel generation, we also employ an inter-instance label selection strategy to further refine the candidate pseudolabels.

Specifically, for each class $c \in [C]$, we employ a vector $\boldsymbol{q}_c = (p_{1c}, p_{2c}, \ldots, p_{Nc})$ to denote the confidence scores across all N instances (i.e., the column vector of the confidence score matrix illustrated in Figure 3). Then, we sort vector $\boldsymbol{q}_c$ in ascending order and construct the candidate pseudolabel set S_i^{inter} of an instance $\boldsymbol{x}_i$. Class c is included in S_i^{inter}, when p_{ic} exhibits relatively higher confidence levels within the vector $\boldsymbol{q}_c$. Given a selection ratio β, for an unlabeled instance $\boldsymbol{x}_i$, its candidate pseudolabel set S_i^{inter} is constructed by

$$S_i^{\text{inter}} = \{c \mid p_{ic} > \text{Quantile}(\text{Sort}(\boldsymbol{q}_c), \beta)\}_{c=1}^{C}. \tag{3}$$

Eventually, for each unlabeled instance $\boldsymbol{x}_i$, its final candidate pseudolabel set S_i is obtained by the intersection of the two candidate pseudolabel sets S_i^{intra} in Eq. (1) and S_i^{inter} in Eq. (3). This can be formally expressed as

$$S_i = S_i^{\text{intra}} \cap S_i^{\text{inter}}. \tag{4}$$

This refined strategy for candidate pseudolabel generation ensures a more balanced and accurate representation of classes, thereby enhancing the model's ability to learn from a diverse and equitable distribution of pseudolabels.

3.2. Learning with Candidate Pseudolabels

With the generated candidate pseudolabels for unlabeled data, we need to construct a suitable training objective to learn from such supervision information. Fortunately, many loss functions have been designed for learning with candidate pseudolabels (a.k.a. partial-label learning) (Feng et al., 2020; Wen et al., 2021; Zhang et al., 2021b), which can be directly used even without any modifications.

At the start of each training iteration, we construct the training set, denoted as D_{T}, from the unlabeled set D_{UL}. This set contains M instance-candidate pairs $\{(\boldsymbol{x}_i, S_i)\}_{i=1}^{M}$. In simple terms, we filter out instances with no candidate labels in D_{UL} and add the remaining instances, along with

their candidate labels, into D_{T}. The construction of D_{T} can be represented as

$$D_{\mathrm{T}} = \{(\boldsymbol{x}_i, S_i) \mid |S_i| > 0\}_{i=1}^{N}. \tag{5}$$

In practical training, the training target of candidate pseudolabels for each unlabeled instance $\boldsymbol{x}_i$ can be re-expressed as a vector $\boldsymbol{s}_i$ with the size of C, where $s_{ic} = 1$ if $c \in S$ and $s_{ic} = 0$ if $c \notin S$, for $c \in [C]$. Subsequently, the instances from D_{T} and the corresponding training targets are utilized for training the model.

Depending on the availability of labeled data in downstream tasks, the training objective of our method can be divided into two forms:

- In the context of semi-supervised learning and transductive zero-shot learning, a small labeled set $D_{\mathrm{L}} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{O}$ is provided. Consequently, at each iteration, we have two sets: labeled set $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{b_1}$ with a batch size of b_1 and $\{(\boldsymbol{x}_i, \boldsymbol{s}_i)\}_{i=1}^{b_2}$ with a batch size of b_2. The training objective is

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{\mathrm{L}} + \lambda \mathcal{L}_{\mathrm{UL}} \\ &= \frac{1}{b_1} \sum\nolimits_{i=1}^{b_1} L_{\mathrm{ce}}(\boldsymbol{x}_i, \boldsymbol{y}_i) + \lambda \frac{1}{b_2} \sum\nolimits_{i=1}^{b_2} L_{\mathrm{p}}(\boldsymbol{x}_i, \boldsymbol{s}_i),\end{aligned}$$

  where L_{ce} represents the cross-entropy loss function used for learning with the labeled set, and L_{p} denotes the loss function used for learning with the candidate pseudolabel set. During training, we typically set b_2 to a pre-fixed value. For b_1, we set it as $(|D_{\mathrm{L}}|/|D_{\mathrm{T}}|) \times b_2$, ensuring that D_{L} and D_{T} have similar iteration counts throughout the training process.
- In the unsupervised learning setting, we only have access to the unlabeled data D_{UL}. Given each mini-batch of training data $\{(\boldsymbol{x}_i, \boldsymbol{s}_i)\}_{i=1}^{b_2}$, the training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{UL}} = \frac{1}{b_2} \sum\nolimits_{i=1}^{b_2} L_{\mathrm{p}}(\boldsymbol{x}_i, \boldsymbol{s}_i).$$

Update of Candiate Pseudolabels. Throughout the training process, the candidate pseudolabels are progressively updated after each pre-defined iteration. For each iteration, we regenerate the candidate pseudolabels for all unlabeled data based on intra- and inter-instance label selection, learning with the newly generated candidate pseudolabels The detailed iterative process can be found in Appendix A.

4. Experiments

To evaluate the effectiveness of our proposed candidate pseudolabel learning (CPL) method, we implement experiments in several dimensions. ❶ **Learning paradigm variety**: we conduct experiments on three learning paradigms including semi-supervised learning, unsupervised learning, and transductive zero-shot learning. ❷ **Prompt tuning variety**: all methods are tested with a textual prompt as well as a visual prompt as CLIP's learnable parameters and tuning strategy. ❸ **Task variety**: we evaluate the effectiveness of CPL on nine classification tasks.

4.1. Experimental Setting

Datasets. We conduct an extensive evaluation of our method on nine classification datasets from diverse domains, including FGVC-Aircraft (Maji et al., 2013), EuroSAT (Helber et al., 2019), CUB (Wah et al., 2011), Flowers102 (Nilsback & Zisserman, 2008), RESISC45 (Cheng et al., 2017), DTD (Cimpoi et al., 2014), CALTECH-101 (Fei-Fei et al., 2004), UCF-101 (Soomro et al., 2012), and CIFAR-100 (Krizhevsky et al., 2009).

Learning Paradigms. To comprehensively evaluate the performance of our method, we consider three common scenarios involving the use of unlabeled data: Unsupervised Learning (UL), Semi-Supervised Learning (SSL), and Transductive Zero-Shot Learning (TRZSL) tasks. The details of each paradigm and how labeled data issues are handled can be found in Appendix C.2.

Hyper-parameter Configuration. Unless otherwise specified, our experiments utilize ViT-B/32 as the visual backbone, and the prefix size is set at 16 for both textual and visual prompt learning. Also, we adopt the Classifier-Consistent (CC) (Feng et al., 2020) as the default loss function for learning with candidate labels in our method. The default prompt "`a photo of a [CLASS]`" is employed to obtain initial predictions from CLIP on all unlabeled instances. We adopt SGD as the optimizer and conduct training for 50 epochs. The learning rate is set at 0.0001 for two warm-up cycles, after which it is adjusted to 0.02 and decays following the cosine annealing rule. For SSL and TRZSL, we just set λ to 1. Regarding candidate pseudolabel update, we designate the iteration number for the CUB dataset as $T = 5$ and $T = 10$ for all other datasets. Further technical specifics can be found in Appendix C.3.

4.2. Comparison with Previous Methods

Experimental Design and Baselines. We carry out experiments on two fine-tuning scenarios: tuning with few-shot unlabeled data and tuning with full unlabeled data. In the former scenario, we select q samples per class from all unlabeled data, making it more appropriate for rapid fine-tuning on the downstream dataset. In our experiments, we set $q = 16$ for all methods to ensure fair comparisons. By contrast, the latter scenario fine-tunes CLIP on the entire unlabeled data to achieve superior performance. We compare our CPL with two existing methods, namely, Few Pseudolabels (FPL) (Menghini et al., 2023) and Grow and Refine

Table 1: Comparison results of top-1 test accuracy (%) on six benchmarks when applying **Textual prompts** as tuning strategy. Note that "✓" and "✗" denote whether full unlabeled data are utilized for fine-tuning or not, respectively.

Methods		Flowers102			RESISC45			DTD		
		SSL	UL	TRZSL	SSL	UL	TRZSL	SSL	UL	TRZSL
Zero-shot CLIP		$63.67_{0.00}$		$63.40_{0.00}$	$54.48_{0.00}$		$54.46_{0.00}$	$43.24_{0.00}$		$43.45_{0.00}$
FPL	✗	$75.96_{0.74}$	$65.67_{0.23}$	$80.97_{0.00}$	$68.13_{0.55}$	$63.07_{0.38}$	$72.11_{0.00}$	$37.10_{5.45}$	$44.96_{0.55}$	$46.30_{0.03}$
CPL (Ours)	✗	$\mathbf{77.36}_{0.24}$	$\mathbf{70.01}_{0.21}$	$\mathbf{84.60}_{0.10}$	$\mathbf{71.73}_{0.57}$	$\mathbf{68.47}_{0.34}$	$\mathbf{72.16}_{0.26}$	$\mathbf{54.63}_{0.79}$	$\mathbf{48.92}_{0.17}$	$\mathbf{59.79}_{1.32}$
GRIP	✓	$83.60_{0.48}$	$69.84_{1.06}$	$86.26_{0.00}$	$74.11_{0.68}$	$70.55_{0.88}$	$81.07_{0.00}$	$56.07_{0.85}$	$46.09_{1.06}$	$65.30_{0.01}$
CPL (Ours)	✓	$\mathbf{89.66}_{0.36}$	$\mathbf{72.90}_{0.78}$	$\mathbf{87.35}_{0.76}$	$\mathbf{80.98}_{0.11}$	$\mathbf{77.39}_{0.44}$	$\mathbf{85.85}_{0.49}$	$\mathbf{61.21}_{0.56}$	$\mathbf{51.91}_{0.71}$	$\mathbf{68.00}_{0.34}$

Methods		CUB			EuroSAT			FGVCAircraft		
		SSL	UL	TRZSL	SSL	UL	TRZSL	SSL	UL	TRZSL
Zero-shot CLIP		$51.82_{0.00}$		$51.57_{0.00}$	$32.88_{0.00}$		$30.54_{0.00}$	$17.58_{0.00}$		$17.86_{0.00}$
FPL	✗	$55.29_{0.59}$	$53.04_{0.53}$	$55.44_{0.20}$	$62.05_{1.64}$	$48.96_{1.49}$	$53.70_{26.87}$	$20.02_{0.77}$	$16.62_{0.67}$	$17.55_{0.37}$
CPL (Ours)	✗	$\mathbf{56.37}_{0.45}$	$\mathbf{54.18}_{0.05}$	$\mathbf{64.01}_{0.17}$	$\mathbf{64.84}_{2.15}$	$\mathbf{51.45}_{1.97}$	$\mathbf{54.03}_{2.27}$	$\mathbf{22.37}_{0.66}$	$\mathbf{18.90}_{0.20}$	$\mathbf{28.47}_{0.43}$
GRIP	✓	$56.65_{0.33}$	$51.42_{0.21}$	$59.48_{0.38}$	$58.66_{2.64}$	$57.21_{1.77}$	$92.33_{0.69}$	$16.98_{0.82}$	$15.22_{0.71}$	$26.08_{0.25}$
CPL (Ours)	✓	$\mathbf{58.53}_{0.24}$	$\mathbf{53.47}_{0.36}$	$\mathbf{66.20}_{0.50}$	$\mathbf{77.51}_{0.80}$	$\mathbf{67.26}_{0.47}$	$\mathbf{93.78}_{0.12}$	$\mathbf{22.48}_{0.63}$	$\mathbf{18.35}_{0.27}$	$\mathbf{30.86}_{0.70}$

Table 2: Comparison results of top-1 test accuracy (%) on six benchmarks when applying **Visual prompts** as tuning strategy. Note that "✓" and "✗" denote whether full unlabeled data are utilized for fine-tuning or not, respectively.

Methods		Flowers102			RESISC45			DTD		
		SSL	UL	TRZSL	SSL	UL	TRZSL	SSL	UL	TRZSL
Zero-shot CLIP		$63.67_{0.00}$		$63.40_{0.00}$	$54.48_{0.00}$		$54.46_{0.00}$	$43.24_{0.00}$		$43.45_{0.00}$
FPL	✗	$67.03_{0.65}$	$65.50_{0.41}$	$71.94_{0.00}$	$65.14_{0.25}$	$62.24_{0.22}$	$67.85_{0.00}$	$47.60_{1.09}$	$47.69_{0.48}$	$52.43_{0.00}$
CPL (Ours)	✗	$\mathbf{70.58}_{0.13}$	$\mathbf{68.94}_{0.16}$	$\mathbf{78.13}_{0.31}$	$\mathbf{68.85}_{0.13}$	$\mathbf{67.97}_{0.17}$	$\mathbf{72.18}_{0.27}$	$\mathbf{52.64}_{0.68}$	$\mathbf{50.37}_{0.46}$	$\mathbf{55.90}_{0.69}$
GRIP	✓	$67.95_{1.2}$	$63.09_{0.56}$	$77.18_{0.00}$	$71.22_{0.77}$	$68.43_{0.61}$	$82.19_{0.00}$	$54.57_{4.86}$	$50.51_{0.99}$	$62.78_{0.00}$
CPL (Ours)	✓	$\mathbf{73.52}_{0.37}$	$\mathbf{67.25}_{0.41}$	$\mathbf{80.14}_{0.73}$	$\mathbf{78.46}_{0.74}$	$\mathbf{72.97}_{0.58}$	$\mathbf{86.67}_{0.33}$	$\mathbf{58.74}_{0.81}$	$\mathbf{53.42}_{0.56}$	$\mathbf{65.31}_{0.78}$

Methods		CUB			EuroSAT			FGVCAircraft		
		SSL	UL	TRZSL	SSL	UL	TRZSL	SSL	UL	TRZSL
Zero-shot CLIP		$51.82_{0.00}$		$51.57_{0.00}$	$32.88_{0.00}$		$30.54_{0.00}$	$17.58_{0.00}$		$17.86_{0.00}$
FPL	✗	$52.86_{0.42}$	$53.17_{0.06}$	$54.17_{0.16}$	$52.47_{2.53}$	$48.79_{3.69}$	$68.68_{14.74}$	$20.14_{0.26}$	$18.28_{0.33}$	$16.28_{0.45}$
CPL (Ours)	✗	$\mathbf{53.37}_{0.55}$	$\mathbf{53.28}_{0.31}$	$\mathbf{56.43}_{0.21}$	$\mathbf{66.37}_{2.10}$	$\mathbf{52.83}_{2.10}$	$\mathbf{74.02}_{1.34}$	$\mathbf{21.52}_{0.68}$	$\mathbf{20.10}_{0.51}$	$\mathbf{26.73}_{0.08}$
GRIP	✓	$\mathbf{53.83}_{0.11}$	$\mathbf{52.91}_{0.26}$	$54.92_{0.17}$	$63.48_{3.09}$	$63.68_{3.42}$	$96.97_{0.77}$	$19.43_{0.50}$	$17.51_{0.61}$	$26.42_{0.30}$
CPL (Ours)	✓	$49.50_{0.42}$	$52.11_{0.24}$	$\mathbf{56.37}_{0.06}$	$\mathbf{72.03}_{1.24}$	$\mathbf{68.93}_{1.15}$	$\mathbf{98.31}_{0.18}$	$\mathbf{20.51}_{0.68}$	$\mathbf{18.26}_{0.38}$	$\mathbf{30.26}_{0.46}$

Iteratively Pseudolabels (GRIP) (Menghini et al., 2023), across six classification tasks under these two scenarios. The performance of each method is reported by calculating the test set accuracy, averaged over three runs. For TRZSL, we report the harmonic mean of the accuracies of seen and unseen classes.

Results about textual prompt tuning and visual prompt tuning are shown in Table 1 and 2, respectively. We observed that: ❶ *Our method consistently outperforms existing hard pseudolabel methods.* Our proposed CPL framework consistently surpasses FPL and GRIP, across a variety of tasks and datasets under both scenarios of tuning with few-shot unlabeled data and tuning with full unlabeled data. This underscores the efficacy of our approach in enhancing the performance of CLIP in downstream tasks. ❷ *Our method exhibits less dependency on the accuracy of zero-shot CLIP and utilizes unlabeled data more effectively.* Our method excels in settings where the zero-shot capability of CLIP is relatively low and the labeled data is scarce. As shown in Table 1, our method performs well on the EuroSAT dataset, where the initial performance of CLIP is subpar. When tuning with the full unlabeled data, our method improves the top-1 test accuracy by 18.85% and 10.05% in the SSL and UL paradigms, compared with GRIP. Similarly, on the DTD dataset, our method improves the top-1 test accuracy by 5.14% and 5.82% in the SSL and UL paradigms, compared with GRIP. These results demonstrate that our method has a lower requirement for the initial zero-shot capability, rendering it more robust across various learning paradigms and tasks.

Table 3: Comparison results of top-1 test accuracy (%) on unsupervised learning when applying **parameter-efficient tuning**. The best and second-best performances are highlighted via **bold** and underline, respectively.

	Flowers-102	UCF-101	CIFAR-100	EuroSAT	DTD	CALTECH-101
CLIP	66.6	61.0	64.2	45.1	42.9	90.5
CLIP-PR	57.7	57.9	63.2	44.2	40.1	84.8
UPL	71.5	63.9	65.8	62.2	48.0	90.6
LaFTer	71.0	68.2	74.6	73.9	46.1	93.3
LaFTer + Ours	**76.7**	**71.0**	**77.3**	**82.2**	**56.3**	**93.4**

Table 4: Comparison results of top-1 test accuracy (%) on DTD with textual prompts tuning. The performance of CPL with five different loss functions is evaluated on three tasks.

Methods		SSL	UL	TRZSL
Zero-shot CLIP		$43.24_{0.00}$		$43.45_{0.00}$
FPL	✗	$37.10_{5.45}$	$44.96_{0.55}$	$46.30_{0.03}$
CPL $_{\text{Soft CE}}$	✗	$51.83_{0.62}$	$47.02_{0.37}$	$59.69_{0.59}$
CPL $_{\text{CC}}$	✗	$54.63_{0.79}$	$48.92_{0.17}$	$59.79_{1.32}$
CPL $_{\text{RC}}$	✗	$54.98_{0.49}$	$49.96_{0.15}$	$59.42_{0.44}$
CPL $_{\text{CAV}}$	✗	$55.50_{0.29}$	$48.69_{0.66}$	$59.44_{0.13}$
CPL $_{\text{LW}}$	✗	$55.21_{0.74}$	$49.82_{0.91}$	$59.24_{0.72}$
GRIP	✓	$56.07_{0.85}$	$46.09_{1.06}$	$65.30_{0.01}$
CPL $_{\text{Soft CE}}$	✓	$60.83_{0.66}$	$49.13_{0.10}$	$66.26_{0.77}$
CPL $_{\text{CC}}$	✓	$61.21_{0.56}$	$51.91_{0.71}$	$68.00_{0.34}$
CPL $_{\text{RC}}$	✓	$60.21_{0.46}$	$51.58_{0.11}$	$67.95_{0.31}$
CPL $_{\text{CAV}}$	✓	$61.06_{0.50}$	$49.31_{0.19}$	$67.76_{0.53}$
CPL $_{\text{LW}}$	✓	$60.20_{0.69}$	$52.23_{0.84}$	$68.29_{0.99}$

4.3. More Analyses

Versatility. Our proposed CPL can be universally applied to existing label-free CLIP fine-tuning scenarios, thereby enhancing their performance. To demonstrate this, we substitute the corresponding pseudolabel module in the existing state-of-the-art method with the candidate pseudolabels generation and update module from CPL. We report the performance of the existing four methods (LaFTer (Mirza et al., 2023), UPL (Huang et al., 2022), and CLIP-PR (Kahana et al., 2022)) under a parameter-efficient tuning strategy while incorporating our proposal into LaFTer.

The results are presented in Table 3. The combiner (LaFTer + Ours) consistently performs best across all six benchmarks. On the DTD dataset, an improvement of 10.2% in top-1 test accuracy is observed compared with the second-best method, LaFTer. Significant improvements across these datasets demonstrate the versatility of our proposed candidate pseudolabels. Consequently, our method is not only effective in its own right but can also enhance the performance of existing CLIP fine-tuning methods when integrated.

Training Loss. Our method is not confined to a particular loss function. Due to this flexibility, it can achieve competitive performance with various loss functions. In addition to the Classifier-Consistent (CC) (Feng et al., 2020), we have also explored four other loss functions capable of handling learning with multiple candidate labels. These include RC (Feng et al., 2020), CAV (Zhang et al., 2021b), LW (Wen et al., 2021), and a soft target cross-entropy. In our proposed soft target cross-entropy (Soft CE), normalized confidence scores from the model prediction are used as the soft targets for the next iteration. We defer the detailed discussion of this soft target scheme to Appendix B.

The comparison results are detailed in Table 4. We observe that all explored loss functions exhibit competitive performance, significantly outperforming the hard pseudolabel methods, FPL and GRIP. This suggests that our CPL is not overly sensitive to the choice of loss function, further highlighting its flexibility. Moreover, we find that all candidate pseudolabel schemes consistently outperform Soft CE, incorporating more prediction information from the prior model. This indicates that the strategy of treating all candidate labels equally can more effectively mitigate the influence of prior category bias, thereby enhancing the performance of CLIP in downstream tasks.

Imbalance Dataset. Considering the universality of the class-imbalanced training set underlying unlabeled data, we conduct relevant experiments to evaluate the performance of our proposed method in the class imbalance setting. Specifically, we keep the convention from Zhou et al. (Zhou et al., 2020) and manually construct an imbalanced CIFAR-100 via an imbalanced ratio δ. We set $\delta = 50$ and $\delta = 100$ (a larger value of δ denotes more imbalanced) to compared the performance of CPL and LaFTer on the imbalanced dataset. The comparison results are shown in Table 5. While class imbalance does exert some influence on the performance of CPL, our results indicate that CPL still outperforms LaFTer, which employs hard pseudolabels, under these imbalanced conditions.

We further explore the impact on CPL's performance when the inter-instance label selection is not applied on imbalanced CIFAR100. As shown in Table 5, the performance of CPL without the inter-instance label selection on imbalanced CIFAR100 is slightly inferior to that of CPL with the inter-instance label selection, especially when the balance ratio is 50. This result further substantiates the necessity of the inter-instance label selection strategy and intra-instance label selection strategy.

Table 5: Comparison of top-1 test accuracy (%) on CIFAR100 dataset with both balanced and imbalanced distribution. Note that "w/o inter" denotes the variant of CPL without the inter-instance label selection.

Methods	Balanced	Imbalanced δ=100	Imbalanced δ=50
LaFTer	74.64	65.63	66.59
CPL (w/o inter)	76.07	66.68	67.85
CPL	**77.32**	**67.70**	**69.65**

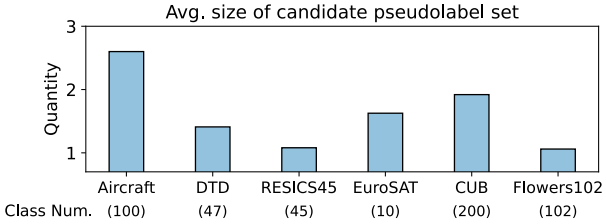


Figure 4: Visualization of the average set size of candidate pseudolabels among all unlabeled data on six datasets under the UL setting of textual prompt tuning.

Set Size of Candidate Pseudolabels. While increasing the size of the candidate set enhances the likelihood of encompassing the true label, it simultaneously amplifies the ambiguity of the training targets, thereby enhancing the difficulty for the model to learn from the candidate labels. We visualize the average size of the generated candidate pseudolabels among all unlabeled data before the last iteration and present the result in Figure 4. It can be observed that the average size of candidate pseudolabels on the majority of the six datasets is close to 1. This implicitly indicates that a large number of unlabeled data have only one candidate label. Consequently, the candidate pseudolabels in our method would not introduce a high degree of ambiguity or high entropy optimization objectives.

Different Proportion of Unlabeled Data. When a large amount of unlabeled downstream data is available, efficiently leveraging it under resource constraints becomes crucial. Typically, utilizing more unlabeled data can yield improved performance but at the cost of increased computational requirements and training time. The question then arises: is it more beneficial to invest additional resources to use all the unlabeled data, or is it better to train with a small amount of high-quality data? To explore this, we compared the performance improvement of CPL on different datasets when a certain proportion of unlabeled data or a small amount of well-labeled data is used for training.

In Figure 5, a common trend is observed across all datasets: as the proportion of data used for training increases, the performance improvement of CPL gradually diminishes, eventually reaching a saturation point. Simultaneously, we

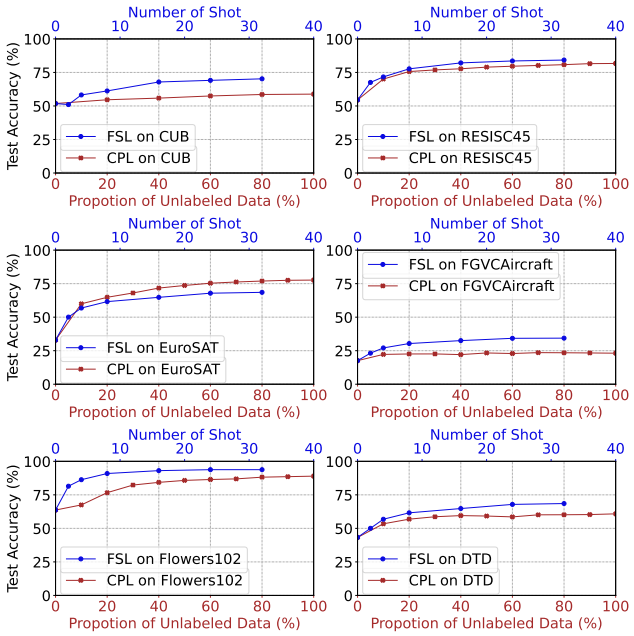


Figure 5: Visualization of the performance improvement of CLIP with CPL (each class only has two labeled data) and fully supervised few-shot learning when textual prompt tuning is applied. The x-axis in blue represents the number of labeled instances, while the x-axis in red represents the proportion of the unlabeled dataset. Both lines originate from the zero-shot performance of CLIP.

also find that few-shot learning usually achieves a higher performance improvement than methods primarily relying on unlabeled data, even though the quantity of used unlabeled data is larger.

This is particularly significant on fine-grained datasets with a large number of classes, such as CUB and FGVC-Aircraft. In addition, the onset of performance saturation when using unlabeled data also happens earlier on these datasets. These observations suggest that when a downstream dataset is challenging, using more well-labeled data for training may be a more efficient way to improve the performance of CLIP under resource constraints.

4.4. Ablation Studies

Sensitivity Analysis. There are mainly two hyperparameters in our CPL method, containing α in Eq. (2) and β in Eq. (3). We conduct a sensitivity study to explore the impact of each hyperparameter on CPL's performance. In Figure 6, we plot the performance of one hyperparameter while keeping the other constant. Generally, the larger the value of α and the smaller the value of β, the more candidate pseudolabels CPL tends to generate during the pseudolabel generation process. In this figure, we find that maintaining the value of α between 0.45 and 0.75 and β between 0.95 and 0.97

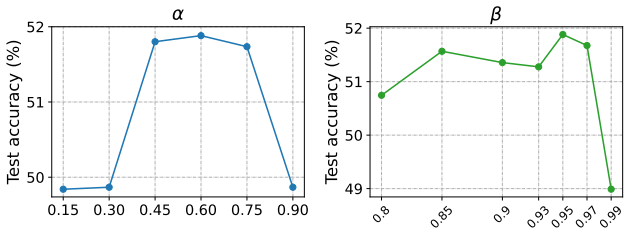


Figure 6: Hyperparameters evaluation on DTD dataset under the UL setting. We illustrate the performance of our method while keeping $\beta = 0.95$ (left) and $\alpha = 0.60$ (right) constant.

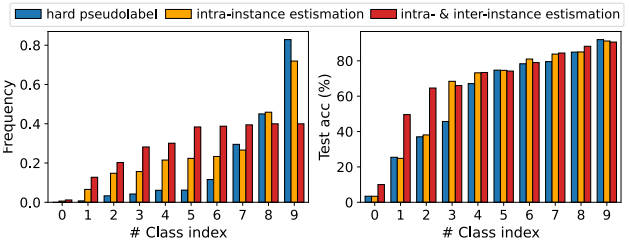


Figure 7: *left*: The frequency distribution of the generated pseudolabels with different strategies. *right*: The influence of each strategy on the class-level test accuracy for EuroSAT.

yields better performance. However, when β exceeds 0.99, the performance declines as CPL tends to generate fewer candidate pseudolabels. Consequently, a large number of unlabeled samples end up with an empty set of candidate labels, reducing the utilization of unlabeled samples. More analyses about these two hyperparameters via grid search are shown in Appendix D.

Effect of Each Selection Strategy. The core of our proposed CPL is the generation of candidate pseudolabels, which includes two label selection strategies. We evaluate the effectiveness of each strategy using two metrics: frequency of different classes in pseudolabels and class-level test accuracy. Firstly, compared to hard pseudolabels, the intra-instance label selection strategy results in a more class-balanced training target (as shown on the left of Figure 7), leading to improved test accuracy. Furthermore, incorporating an additional inter-instance label selection enables CPL to achieve a more balanced distribution among different classes, particularly evident in classes 5-9 as shown on the left of Figure 7. The right side of Figure 7 illustrates the impact of each strategy on the class-level test accuracy. The results indicate that both strategies contribute to the improvement in test accuracy. These results underscore two key points: ❶ both intra-instance and inter-instance label selection strategies are crucial for generating high-quality candidate pseudolabels and ❷ the inter-instance label selection can further enhance class balance and test accuracy.

Different Image Encoders. We further conduct experiments to evaluate the effect of different image encoders. The comparison results on Flowers102, RESISC45, and DTD using ViT-L/14 are presented in Table 6. Our method consistently surpasses the previous methods, demonstrating the effectiveness of our approach in enhancing the performance of CLIP in downstream tasks when larger image encoders are employed. This suggests that our method is capable of better leveraging unlabeled data to improve the performance of CLIP, even when the model size is increased.

Table 6: Comparison results of a different image encoder (ViT-L/14) when applying textual prompt tuning.

	Methods		SSL	UL	TRZSL
DTD	Zero-shot CLIP		$52.45_{0.00}$		$51.61_{0.00}$
	FPL	✗	$60.61_{1.56}$	$52.99_{0.43}$	$60.77_{0.54}$
	CPL	✗	$\mathbf{62.78}_{0.17}$	$\mathbf{57.23}_{0.19}$	$\mathbf{62.52}_{1.38}$
	GRIP	✓	$60.91_{0.00}$	$54.40_{0.00}$	$64.92_{0.00}$
	CPL	✓	$\mathbf{69.82}_{0.32}$	$\mathbf{57.20}_{0.45}$	$\mathbf{71.97}_{0.46}$
RESISC45	Zero-shot CLIP		$62.67_{0.00}$		$62.13_{0.00}$
	FPL	✗	$79.01_{0.55}$	$70.85_{0.66}$	$77.69_{0.83}$
	CPL	✗	$\mathbf{80.38}_{0.37}$	$\mathbf{76.01}_{0.19}$	$\mathbf{79.97}_{0.77}$
	GRIP	✓	$81.53_{0.00}$	$76.86_{0.00}$	$86.88_{0.00}$
	CPL	✓	$\mathbf{87.75}_{0.29}$	$\mathbf{80.88}_{0.86}$	$\mathbf{89.73}_{1.73}$
Flowers102	Zero-shot CLIP		$73.98_{0.00}$		$73.05_{0.00}$
	FPL	✗	$\mathbf{89.07}_{0.94}$	$77.81_{0.30}$	$91.84_{0.73}$
	CPL	✗	$88.37_{0.39}$	$\mathbf{82.98}_{0.14}$	$\mathbf{96.65}_{0.08}$
	GRIP	✓	$94.21_{0.00}$	$82.33_{0.00}$	$96.18_{0.00}$
	CPL	✓	$\mathbf{96.80}_{0.63}$	$\mathbf{83.94}_{0.69}$	$\mathbf{97.34}_{0.74}$

5. Limitations

It is important to note that the performance of our method is dependent on the quality of the generated candidate pseudolabels. As such, it inherits the inherent limitation of pseudolabeling methods - the true label may not be included in the generated candidate pseudolabels. In future work, we aim to enhance the quality of candidate pseudolabels by refining the generation strategy and devising a method to better handle this situation.

6. Conclusion

In this paper, we have proposed a novel candidate pseudolabel learning (CPL) method to fine-tune vision-language models (VLMs), with abundant unlabeled data. The key to our method lies in the strategy of generating suitable candidate pseudolabels, which contains both intra- and inter-instance label selection. In this way, our generated candidate pseudolabels can offer two key advantages over conventional hard pseudolabels: enhanced accuracy in true label estimation and balanced representation across classes. Our extensive experiments, conducted across nine benchmark datasets and three learning paradigms validated the effectiveness of our method. This is particularly evident when the zero-shot capabilities of VLMs are not reliably applicable to downstream tasks.

Acknowledgements

Lei Feng is supported by the National Natural Science Foundation of China (Grant No. 62106028) and the Chongqing Overseas Chinese Entrepreneurship and Innovation Support Program. Feng Liu is supported by the Australian Research Council with grant numbers DP230101540 and DE240101089, and the NSF&CSIRO Responsible AI program with grant number 2303037.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: A visual language model for few-shot learning. In *NeurIPS*, volume 35, pp. 23716–23736, 2022.

Bahng, H., Jahanian, A., Sankaranarayanan, S., and Isola, P. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022.

Cascante-Bonilla, P., Tan, F., Qi, Y., and Ordonez, V. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *AAAI*, volume 35, pp. 6912–6920, 2021.

Cheng, G., Han, J., and Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *CVPR*, pp. 3606–3613, 2014.

Cour, T., Sapp, B., and Taskar, B. Learning from partial labels. *The Journal of Machine Learning Research*, 12: 1501–1536, 2011.

Fei-Fei, L., Fergus, R., and Perona, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR Workshop*, pp. 178–178. IEEE, 2004.

Feng, L., Lv, J., Han, B., Xu, M., Niu, G., Geng, X., An, B., and Sugiyama, M. Provably consistent partial-label learning. In *NeurIPS*, volume 33, pp. 10948–10960, 2020.

Gao, R., Hou, X., Qin, J., Chen, J., Liu, L., Zhu, F., Zhang, Z., and Shao, L. Zero-vae-gan: Generating unseen features for generalized and transductive zero-shot learning. *IEEE Transactions on Image Processing*, 29:3665–3680, 2020.

Ge, C., Huang, R., Xie, M., Lai, Z., Song, S., Li, S., and Huang, G. Domain adaptation via prompt learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. In *NeurIPS*, volume 17. MIT Press, 2004.

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

Hossain, M. and Kauranen, I. Crowdsourcing: A comprehensive literature review. *Strategic Outsourcing: An International Journal*, 8(1):2–22, 2015.

Huang, T., Chu, J., and Wei, F. Unsupervised prompt learning for vision-language models. *arXiv preprint arXiv:2204.03649*, 2022.

Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., and Lim, S.-N. Visual prompt tuning. In *ECCV*, pp. 709–727. Springer, 2022.

Kahana, J., Cohen, N., and Hoshen, Y. Improving zero-shot models with label distribution priors. *arXiv preprint arXiv:2212.00784*, 2022.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lai, Z., Vesdapunt, N., Zhou, N., Wu, J., Huynh, C. P., Li, X., Fu, K. K., and Chuah, C.-N. Padclip: Pseudo-labeling with adaptive debiasing in clip for unsupervised domain adaptation. In *ICCV*, pp. 16155–16165, 2023.

Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., and Hoi, S. C. H. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, volume 34, pp. 9694–9705, 2021.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, pp. 12888–12900. PMLR, 2022.

Li, J., Sun, H., and Li, J. Beyond confusion matrix: Learning from multiple annotators with awareness of instance features. *Machine Learning*, 112(3):1053–1075, 2023.

Luo, J. and Orabona, F. Learning from candidate labeling sets. In *NeurIPS*, volume 23, 2010.

Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Menghini, C., Delworth, A., and Bach, S. H. Enhancing clip with clip: Exploring pseudolabeling for limited-label prompt tuning. In *NeurIPS*, 2023.

Mirza, M. J., Karlinsky, L., Lin, W., Kozinski, M., Possegger, H., Feris, R., and Bischof, H. Lafter: Label-free tuning of zero-shot classifier using language and unlabeled image collections. *arXiv preprint arXiv:2305.18287*, 2023.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *ICVGIP*, pp. 722–729, 2008.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pp. 69–84. Springer, 2016.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *ICML*, pp. 8748–8763. PMLR, 2021.

Schmarje, L., Santarossa, M., Schröder, S.-M., and Koch, R. A survey on semi-, self-and unsupervised learning for image classification. *IEEE Access*, 9:82146–82168, 2021.

Shu, M., Nie, W., Huang, D.-A., Yu, Z., Goldstein, T., Anandkumar, A., and Xiao, C. Test-time prompt tuning for zero-shot generalization in vision-language models. In *NeurIPS*, volume 35, pp. 14274–14289, December 2022.

Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., Cubuk, E. D., Kurakin, A., and Li, C.-L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, volume 33, pp. 596–608. Curran Associates, Inc., 2020.

Soomro, K., Zamir, A. R., and Shah, M. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Tanwisuth, K., Zhang, S., Zheng, H., He, P., and Zhou, M. POUF: Prompt-oriented unsupervised fine-tuning for large pre-trained models. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 33816–33832. PMLR, 23–29 Jul 2023.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Wan, Z., Chen, D., Li, Y., Yan, X., Zhang, J., Yu, Y., and Liao, J. Transductive zero-shot learning with visual structure constraint. In *NeurIPS*, volume 32, 2019.

Wei, Q., Feng, L., Sun, H., Wang, R., He, R., and Yin, Y. Learning sample-aware threshold for semi-supervised learning. *Machine Learning*, 2024.

Wen, H., Cui, J., Hang, H., Liu, J., Wang, Y., and Lin, Z. Leveraged weighted loss for partial label learning. In *ICML*, 2021.

Xu, Y., Shang, L., Ye, J., Qian, Q., Li, Y.-F., Sun, B., Li, H., and Jin, R. Dash: Semi-supervised learning with dynamic thresholding. In *ICML*, pp. 11525–11536. PMLR, 2021.

Yuan, L., Chen, D., Chen, Y.-L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

Zhang, B., Wang, Y., Hou, W., WU, HAO., Wang, J., Okumura, M., and Shinozaki, T. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *NeurIPS*, volume 34, pp. 18408–18419. Curran Associates, Inc., 2021a.

Zhang, F., Feng, L., Han, B., Liu, T., Niu, G., Qin, T., and Sugiyama, M. Exploiting class activation value for partial-label learning. In *ICLR*, October 2021b.

Zhang, J., Huang, J., Jin, S., and Lu, S. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Zhang, R., Fang, R., Zhang, W., Gao, P., Li, K., Dai, J., Qiao, Y., and Li, H. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021c.

Zhou, B., Cui, Q., Wei, X.-S., and Chen, Z.-M. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *CVPR*, pp. 9719–9728, 2020.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Conditional prompt learning for vision-language models. In *CVPR*, pp. 16816–16825, 2022a.

Zhou, K., Yang, J., Loy, C. C., and Liu, Z. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022b.

Appendix for *"Candidate Pseudolabel Learning: Enhancing Vision-Language Models by Prompt Tuning with Unlabeled Data"*

In the appendix of this paper, we provide further details:

- Elaboration on the iterative process for updating candidate pseudolabels (Appendix A).
- Exploration of an alternative approach for formulating a soft target from the candidate label set (Appendix B).
- Additional information about the datasets and hyperparameters used in our method (Appendix C).
- Presentation of experimental results derived from varying the primary hyperparameters and examination of the influence of the other hyperparameters (Appendix D).

A. Details about Training Iterations

In this section, we provide a detailed description of our iterative process for updating candidate pseudolabels, as outlined in Algorithm 1. Specifically, in each iteration, after generating candidate pseudolabels for each instance, we filter out instances where the candidate set is empty, forming D_{temp}. Subsequently, for each class, we select the K_t instances with the highest confidence scores, add these instances to the training set D_{T}, and simultaneously remove these instances from D_{UL} to avoid repeated selection. Through this step, we ensure that the number of instances for each class does not exceed K_t and output the training set D_{T} for the current iteration.

Simultaneously, to utilize more training data, after the iteration is completed, we increase K_t by Δ so that more unlabeled data can be utilized in the next iteration. This iteration process is then repeated until we reach the maximum number of iterations T. Normally, The increment for each category's quantity per iteration is set to $\Delta = \frac{|D_{\text{UL}}|}{T}$. This process is akin to curriculum learning, as it starts training with simple and reliable instances and gradually increases the difficulty. The emphasis is on updating the candidate pseudolabels corresponding to unlabeled data before each iteration and reinitializing the learnable parameters of the model.

The overall iteration process in the CPL and prior work (Menghini et al., 2023) both adhere to the above-mentioned iterative update strategy, ensuring fair comparison. At the same time, we have made some simple modifications to make it more suitable for our candidate pseudolabels.

Specifically, to select a designated number of instances for class c (class-wise Top-K_t selection), the approach adopted in previous work (Menghini et al., 2023) utilized the confidence scores of all potential labels for the ranking process. In contrast, we confine our ranking and selection to only the labels included in candidate label set S. In other words, we only involve all confidence scores of the instances for which $c \in S$ in the ranking process. This makes the improved top-K selection method more appropriate for candidate pseudolabels, as we should prioritize the categories in the candidate label set during the selection process and exclude the influence of non-candidate categories.

Table 7: Comparison results of top-1 test accuracy (%) on UL task with textual prompts tuning. The main difference here is using a soft target or not.

Methods	EuroSAT	DTD	Flowers102
CPL $_{\text{CC}}$	67.26	51.91	72.90
CPL $_{\text{RC}}$	66.38	51.58	71.91
CPL $_{\text{CAV}}$	66.91	49.31	72.06
CPL $_{\text{LW}}$	67.15	52.23	72.33
CPL $_{\text{Soft CE}}$	56.85	49.53	71.87
CPL $_{\text{Soft CC}}$	65.58	50.98	72.58
CPL $_{\text{Soft RC}}$	65.89	50.83	71.89
CPL $_{\text{Soft CAV}}$	64.90	48.78	71.72
CPL $_{\text{Soft LW}}$	65.74	51.04	72.15

Algorithm 1 Top-K Selection Process in Each Iteration

Input: Total iteration number T, an unlabeled set $D_{\mathrm{UL}} := \{\boldsymbol{x}_i\}_{i=1}^{N}$, the number of instances K_t that should be selected in iteration t for all classes. The increment number between two iterations is Δ.
Output: The training set D_{T} with candidate psuedolabels

for $t \in 1, ..., T$ **do**
  Initialize the training set $D_{\mathrm{T}} := \varnothing$
  Generate candidate pseudolabels S for each unlabeled instance $\boldsymbol{x}_i$ according to two label selection strategies (with Eq. (1) (3)).
  Refine the set of instance-candidate pairs $D_{\mathrm{temp}} := \{(\boldsymbol{x}_i, S_i)\}_{i=1}^{M}$ by filtering the sample where $S_i = \varnothing$
  for $c \in [C]$ **do**
    $Q := |D_{\mathrm{temp}}|$ ▷ Q is the number of instances in current D_{temp}
    $\mathcal{V}_c := \{p_{ic} | c \in S_i\}_{i=1}^{Q}$ ▷ Collect the corresponding confidence scores when c is contained in the set S_i
    $D_{\mathrm{T}} \leftarrow D_{\mathrm{T}} \bigcup \{(\boldsymbol{x}_i, S_i) | p_{ic} \in \text{Top-}K_t(\mathcal{V}_c)\}_{i=1}^{Q}$ ▷ Select top-K_t instances according to the candidate labels' confidence scores
    if sample $(\boldsymbol{x}_i, S_i)$ is selected **then**
      $D_{\mathrm{temp}} \leftarrow D_{\mathrm{temp}} \backslash (\boldsymbol{x}_i, S_i)$
    end if
  end for
  $K_{t+1} := K_t + \Delta$
  Transform each candidate set S in D_{T} into training target $\boldsymbol{s}$.
  Return the training set D_{T}.
end for

B. Discussion on Soft Pseudolabels

In this section, we explore an alternative approach to utilizing the candidate label set S as the learning target, specifically by formulating a soft target derived from S. This approach aims to provide a more comprehensive evaluation of our method.

Unlike candidate pseudolabels, which treat all candidate labels equally, soft pseudolabels utilize normalized confidence scores from the previous model's prediction as the soft target for the subsequent iteration. Specifically, the confidence scores from the preceding model's output for each category $c \in S$ are used as the criteria for defining the soft target.

Let $\hat{\boldsymbol{y}} = (y_1, y_2, y_3, \cdots, y_C)$ represents the soft target for unlabeled instance $\boldsymbol{x}$. For a category $c \in [C]$, the corresponding value y_c in soft target can be calculated as follows:

$$y_c = \begin{cases} \frac{P(y=c|\boldsymbol{x})}{\sum_{k \in S} P(y=k|\boldsymbol{x})} = \frac{p_c}{\sum_{k \in S} p_k}, & \text{if } c \in S \\ 0, & \text{otherwise} \end{cases}, \tag{6}$$

where S is the candidate pseudolabel set for instance $\boldsymbol{x}$. We compared the performance between this soft target generation method and CPL which treats all candidate labels equally, as shown in Table 7.

In Table 7, "Soft CE" refers to the approach where the soft target is obtained using Equation (6) before each iteration, and training is conducted using the cross-entropy loss. Similarly, "Soft CC" refers to the method where the soft target is obtained in the same manner, but training is executed using the CC objective function. All other experimental settings are the same as those in Section 4.1.

From the table, we observe that methods not utilizing the soft target exhibit marginally superior performance on the three datasets. This suggests that directly using candidate pseudolabels as the training target can more effectively enhance the performance of CLIP when fine-tuning on downstream tasks, especially when the zero-shot capabilities of CLIP are relatively weak on these datasets. We posit that this may be attributed to the fact that the utilization of a soft target, which contains more information, might inadvertently reinforce the model's previous erroneous predictions and class bias, thus exacerbating the impact of confirmation bias and hindering subsequent learning process. Therefore, in the main text, we implement our CPL by treating all candidate labels equally.

C. Experimental Details

C.1. Comparison Methods

In this section, we provide a detailed introduction to the methods included in our experiments, which are divided into two categories. The first category encompasses strategies for fine-tuning CLIP under limited data conditions, including:

- **Few-pseudolabels (FPL)** (Menghini et al., 2023): This approach generates offline pseudolabels by selecting the top-K

samples with the highest confidence for each class from CLIP zero-shot predictions, performed only once.

- **Grow and Refine Iteratively Pseudolabels (GRIP)** (Menghini et al., 2023): GRIP maintains class balance by selecting the top-K samples at each iteration for each class, with K progressively increasing after each iteration. The key distinction between GRIP and FPL lies in GRIP's iterative pseudolabel updates and the incremental increase of samples for each class at each iteration.
- **Unsupervised Prompt Learning (UPL)** (Huang et al., 2022): Align with FPL, UPL employs the most confident samples for each class and generates offline pseudolabels to learn text prompts through the CLIP text encoder.
- **CLIP-PR** (Kahana et al., 2022): This method optimizes an adapter atop the CLIP vision encoder. It uses label distribution priors from the training set of downstream datasets and generates offline pseudolabels only once.
- **LaFTer** (Mirza et al., 2023): This method utilizes an unlabeled image collection and a set of text descriptions generated by a Large Language Model (LLM) to fine-tune CLIP with online pseudolabels. Notably, it also generates hard pseudolabels and employs a consistency regularization strategy (Sohn et al., 2020) to learn from unlabeled data.

The second category pertains to the loss functions in partial-label learning:

- **Classifier-Consistent (CC) & Risk-Consistent (RC)** (Feng et al., 2020): These two methods are designed for partial-label learning. They develop two novel methods that are guaranteed to be provably consistent when dealing with learning from a candidate set of labels.
- **Class Activation Value (CAV)** (Zhang et al., 2021b) : This method introduces the class activation value as a versatile tool to select the true label. It identifies the class with the maximum CAV for model training.
- **Leveraged Weighted (LW) Loss** (Wen et al., 2021): The leveraged weighted loss function introduces a leverage parameter to balance the losses on partial labels and non-partial ones.

C.2. Task Introduction

We provide a detailed explanation of experimental settings for three learning paradigms, in line with (Menghini et al., 2023).

- 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 **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 **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 unseen classes. It is noteworthy that in TRZSL, we report the harmonic mean of the accuracies of seen and unseen classes.

Table 8: Detailed settings for experiments in Section 4

	Flowers102	RESISC45	DTD	CUB	EuroSAT	FGVCAircraft
Statistic data						
Class number	102	45	47	200	10	100
Training set size	2040	6300	3760	5594	27000	6667
Testing set size	6149	25200	1880	5794	5000	3333
Training procedure						
Network	ViT-B / 32					
Batch size	64					
Epoch	50 where first two epochs are set for warmup					
Optimizer	SGD					
Momentum	0.9					
Learning rate (LR)	0.02					
Weight decay	5e-2					
LR scheduler	CosineAnnealingLR					
Hyperparameters						
α in intra-instance label selection	0.60	0.90	0.75	0.75	0.75	0.90
β in inter-instance label selection	0.99	0.97	0.95	0.99	0.80	0.97

C.3. Datasets and Hyperparameters

In this section, we provide additional visualization and details regarding the datasets and hyperparameters used in CPL.

Setup. We provide the statistical data for six datasets and the complete experimental setup in Table 8.

Additional Dataset Visualizations. In addition to the pilot experiment in Figure 1(a) in the main text, which reveals the low label estimation accuracy and class bias issues of hard pseudolabels on EuroSAT, we also visualize the confusion matrix on the DTD dataset in a similar manner, as shown in Figure 8. This visualization reveals similar issues on the DTD dataset.

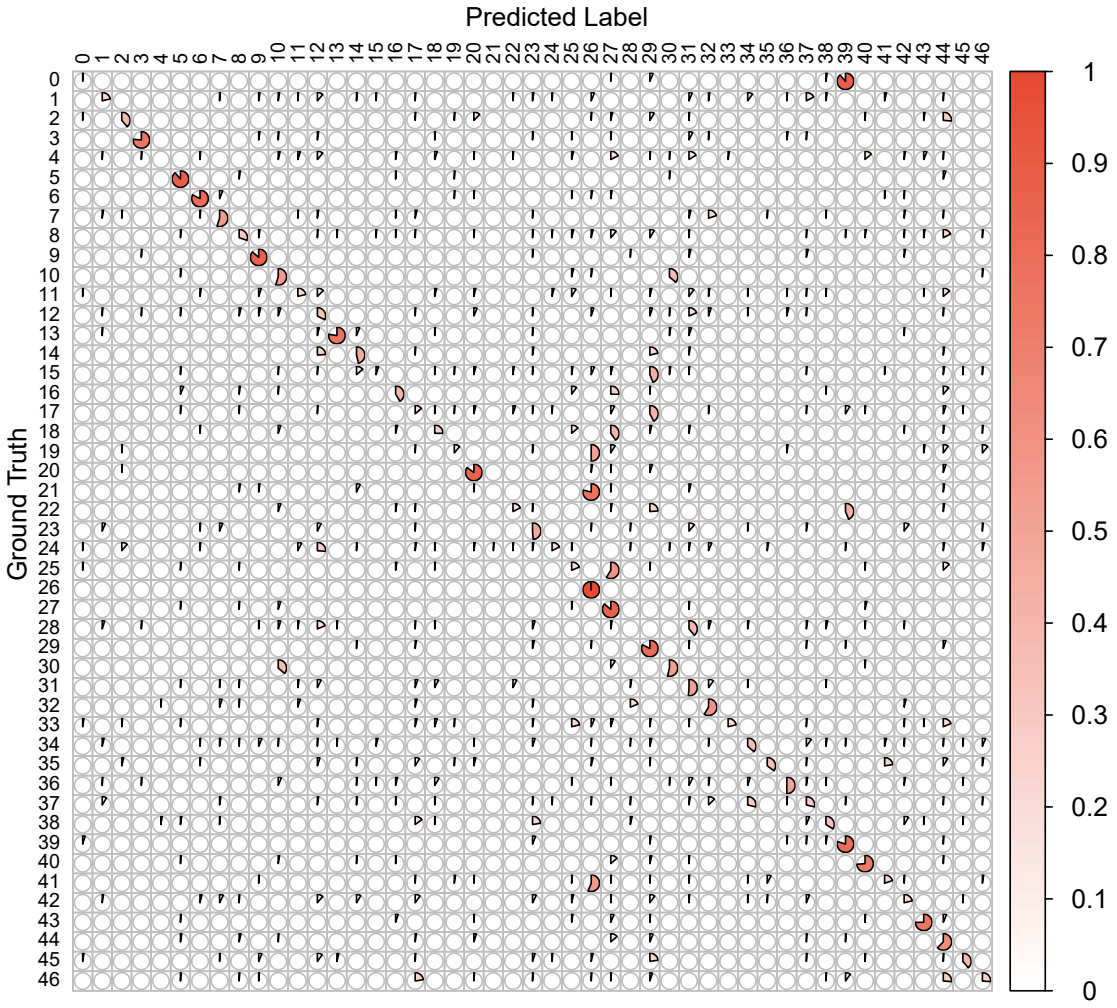


Figure 8: Confusion matrix illustrating the correlation between true labels and predicted hard pseudolabels on DTD.

D. Additional Experimental Results

In this section, we present the experimental results derived from our method when varying the two primary hyperparameters (α, β) and hyperparameter λ which controls the balance between labeled and unlabeled data for SSL and TRZSL tasks.

Ablation Studies for the Trade-off Coefficient λ. For the SSL and TRZSL tasks, we conduct ablation studies on the hyperparameter λ, as shown in Table 9 and Table 10. We observe that, except for a few datasets (e.g., Flowers102), most datasets do not exhibit excessive dependency on the hyperparameter λ. Therefore, for SSL and TRZSL, we consistently set λ to 1 in the main text to avoid the impact of over-parameterization.

Grid Search for Hyperparameter Selection. We conduct ablation studies to examine the influence of hyperparameters more comprehensively. We set $\alpha \in \{0.15, 0.30, 0.45, 0.60, 0.75, 0.90\}$ and $\beta \in \{0.80, 0.90, 0.93, 0.95, 0.97, 0.99\}$ for SSL and UL, and $\beta \in \{0.60, 0.70, 0.75, 0.85, 0.90, 0.95\}$ for TRZSL.The results of these experiments for the three tasks on

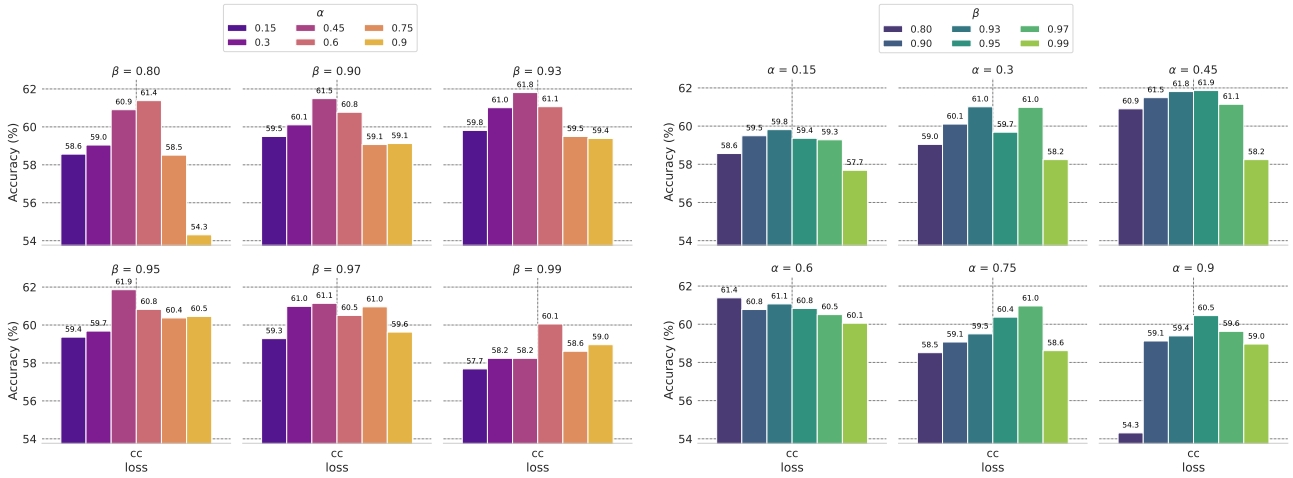
Table 9: Comparison of top-1 test accuracy (%) on **SSL** tasks with textual prompt tuning, illustrating the influence of λ. Here, we use all the unlabeled data for fine-tuning CLIP.

Methods	Flowers102	RESISC45	DTD	CUB	EuroSAT	FGVCAircraft
Zero-shot CLIP	63.67	54.48	43.24	51.82	32.80	17.58
CPL $_{\lambda=0.50}$	90.09	80.97	58.96	58.11	76.71	22.76
CPL $_{\lambda=0.75}$	90.28	81.76	60.13	58.46	76.96	22.07
CPL $_{\lambda=1.00}$	89.66	80.98	61.21	58.53	77.51	22.48
CPL $_{\lambda=1.25}$	89.68	81.37	59.31	58.42	77.60	22.86
CPL $_{\lambda=1.50}$	88.77	81.64	61.28	58.28	77.33	22.65

Table 10: Comparison of top-1 test accuracy (%) on **TRZSL** tasks with textual prompt tuning, illustrating the influence of λ. Here, we use all the unlabeled data for fine-tuning CLIP.

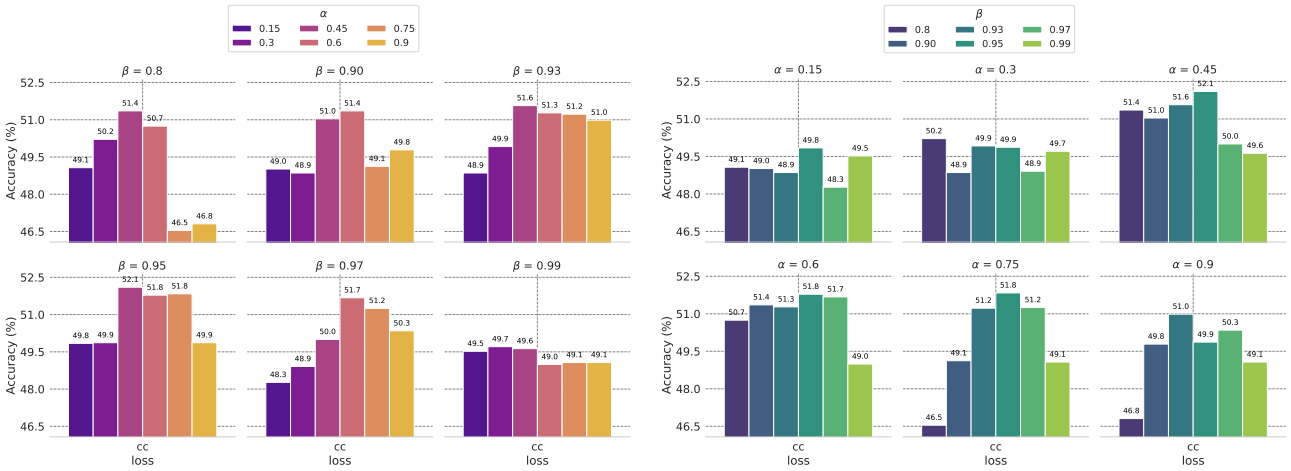
Methods	Flowers102	RESISC45	DTD	CUB	EuroSAT	FGVCAircraft
Zero-shot CLIP	63.40	54.46	43.45	51.57	30.54	17.86
CPL $_{\lambda=0.50}$	90.01	85.95	67.31	65.28	93.37	32.60
CPL $_{\lambda=0.75}$	86.91	84.91	67.77	65.25	93.11	31.47
CPL $_{\lambda=1.00}$	87.35	85.85	68.00	63.94	93.78	30.26
CPL $_{\lambda=1.25}$	88.98	86.10	67.82	64.84	94.01	30.42
CPL $_{\lambda=1.50}$	87.13	85.96	68.11	64.81	93.56	30.05

the DTD dataset are depicted in Figures 9, 10, and 11. The results indicate that our method is robust to changes in these hyperparameters in a range (especially in TRZSL and SSL), and can achieve competitive performance across a wide range of settings.



(a) SSL on DTD (b) SSL on DTD

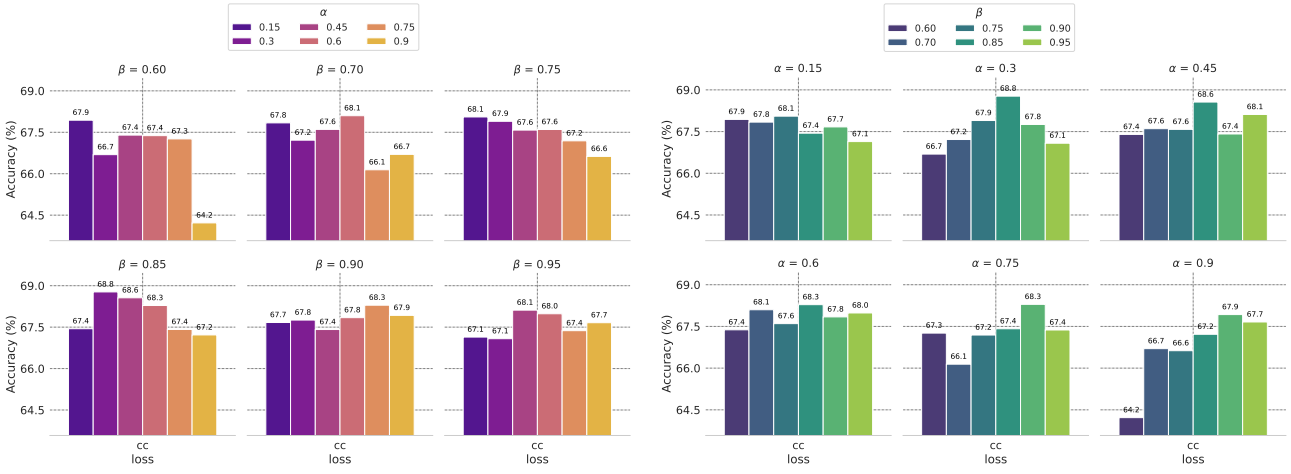
Figure 9: (a) Illustrates the impact of the parameter α under various settings of β. (b) Illustrates the impact of the parameter β under various settings of α.



(a) UL on DTD

(b) UL on DTD

Figure 10: (a) Illustrates the impact of the parameter α under various settings of β. (b) Illustrates the impact of the parameter β under various settings of α.



(a) TRZSL on DTD

(b) TRZSL on DTD

Figure 11: (a) Illustrates the impact of the parameter α under various settings of β. (b) Illustrates the impact of the parameter β under various settings of α.