Erasing the Bias: Fine-Tuning Foundation Models for Semi-Supervised Learning

Kai Gan [1] **Tong Wei** [1]

Abstract

Semi-supervised learning (SSL) has witnessed remarkable progress, resulting in the emergence of numerous method variations. However, practitioners often encounter challenges when attempting to deploy these methods due to their subpar performance. In this paper, we present a novel SSL approach named FINESSL that significantly addresses this limitation by adapting pre-trained foundation models. We identify the *aggregated biases* and *cognitive deviation* problems inherent in foundation models, and propose a simple yet effective solution by imposing balanced margin softmax and decoupled label smoothing. Through extensive experiments, we demonstrate that FINESSL sets a new state of the art for SSL on multiple benchmark datasets, reduces the training cost by over six times, and can seamlessly integrate various fine-tuning and modern SSL algorithms. The source code is available at https://github.com/Gank0078/FineSSL.

1. Introduction

Semi-supervised learning (SSL) has emerged as a prominent learning paradigm of machine learning, which aims to train models using a combination of a large amount of unlabeled data and a limited number of labeled samples. Recently, numerous SSL algorithms have been proposed to address this challenge by automatically generating pseudo-labels for unlabeled samples using the model and selecting reliable ones for subsequent training (Lee et al., 2013; Bachman et al., 2014; Laine & Aila, 2016; Sajjadi et al., 2016; Xie et al., 2020; Rizve et al., 2021; Wang et al., 2022c). Specifically, these algorithms consider model predictions with confidence higher than a specified threshold as reliable pseudo-labels, which are then used to calculate the unsupervised loss. Modern SSL methods like FixMatch (Sohn et al., 2020) and FlexMatch (Zhang et al., 2021) determine the use of pseudo-labels based on a fixed threshold of 0.95 or dynamic thresholds, employing consistency regularization to enforce consistent predictions between weak and strong augmented views of an image.

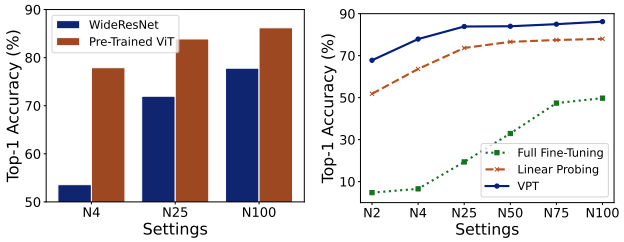


Figure 1. Left: Fine-tuning pre-trained ViT significantly outperforms training Wide ResNet starting from scratch. *Right*: VPT improves full fine-tuning and linear probing by a large margin. Experiments are conducted on CIFAR-100 using FixMatch. Throughout the paper, we denote the setting with 4 labeled samples for each class as "N4", and other settings are defined accordingly.

While current SSL algorithms have shown promising performance on various tasks, there is still a gap when it comes to deploying them in real-world applications, especially in scenarios with extremely limited labeled data. To address this issue, we propose to develop a more effective SSL approach based on foundation models, *e.g.,* CLIP (Radford et al., 2021), instead of training models from scratch. These foundation models are pre-trained on large-scale pretext datasets and possess the ability to learn generalizable representations that can be transferred to different tasks. In fact, the adaptation of foundation models has already demonstrated effectiveness in various downstream tasks, including few-shot learning (Liu et al., 2022), long-tail learning (Shi et al., 2023), and natural language understanding (Yang et al., 2022b). However, the potential of foundation models in improving SSL remains unexplored. To motivate our approach, we first find that visual prompt tuning (VPT) (Jia et al., 2022), a representative parameter-efficient fine-tuning (PEFT) method, is better suited for SSL tasks compared to commonly used full fine-tuning (FFT) and linear probing (LP). Note that PEFT keeps the pre-trained model frozen and only learns a small set of task-specific parameters for adaptation, while FFT updates the entire neural network and LP modifies only the linear classifier. As depicted in

[1]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China. Correspondence to: Tong Wei <weit@seu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

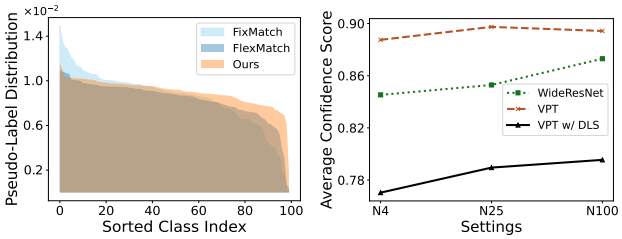


Figure 2. Left: The distribution of pseudo-labels for unlabeled data. Classes are sorted by the frequencies of pseudo-labels for each class. *Right*: The average confidence across settings with different numbers of labeled samples per class based on FixMatch. "DLS" denotes the decoupled label smoothing proposed in Section 4.2. Experiments are conducted on CIFAR-100.

Figure 1, fine-tuning pre-trained vision Transformer yields significantly better results than training Wide ResNet from scratch (Zagoruyko & Komodakis, 2016), and VPT outperforms other fine-tuning methods across all SSL settings.

Considering that PEFT helps improve the performance of pre-trained models in downstream tasks, we are intrigued by its potential as a versatile approach for combining mainstream SSL methods and leveraging the full potential of unlabeled data. However, upon investigating this question, we uncover that fine-tuning foundation models is a double-edged sword. While it improves performance over random initialization, it can also have adverse effects on the selection of reliable pseudo-labels during training. We identify two perspectives from which this issue arises.

One of the perspectives we identified is that foundation models can exhibit biases among different groups of classes due to the inherent imbalances in their pre-training datasets (Zhu et al., 2023). Consequently, they tend to generate more pseudo-labels for classes that occur more frequently, particularly when employing strategies like VPT, enabling the reservation of more pre-trained representations (Yang et al., 2022b). This confirmation bias (Arazo et al., 2020; Wang et al., 2021) becomes more pronounced as the training progresses, leading to the selection of incorrect pseudo-labels with high model predictive confidence. We refer the amplification of the foundation model's inherent imbalances during the training of SSL methods due to conformation bias as *aggregated biases*. We illustrate its impact on the distribution of pseudo-labels generated by FixMatch and FlexMatch in the left panel of Figure 2. The figure clearly shows that FixMatch exhibits a significantly imbalanced distribution of pseudo-labels, while FlexMatch partially mitigates this issue by dynamically adjusting confidence thresholds based on the learning difficulties of different classes. However, there is still significant room for improvement in addressing the challenge of aggregated biases.

The second perspective we identified is that the predictive confidence of foundation models is not reliable. More concretely, this can be manifested in holistic and individual aspects. From the holistic aspect, we observe that confidence scores cannot accurately reflect the true learning difficulties of different SSL tasks. The right panel of Figure 2 provides an intuitive demonstration of this finding. In this experiment, we compare the average predictive confidence of FixMatch using Wide ResNet and VPT across three different settings representing varying levels of learning difficulties. Overall, the Wide ResNet model tends to become more confident in its predictions as the number of labeled samples in the training data increases. However, VPT tends to be overconfident (Thulasidasan et al., 2019; Chan et al., 2020) even when each class only has four labeled samples, *i.e.,* the "N4" setting. Moreover, the model's confidence level in the "N25" setting surpasses that of the "N100" setting, leading to inaccurate estimates of learning difficulties. In addition, from the individual aspect, we find that the confidence score fails to effectively distinguish between correctly labeled pseudo-labels, incorrectly labeled pseudo-labels, and out-of-distribution (OOD) samples. As depicted in Figure 4b, the distribution of the confidence score across various samples exhibits substantial overlap, which impedes the ability to reliably select high-quality pseudo-labels. We term the phenomenon as *cognitive deviation* which makes it challenging to choose confidence thresholds for pseudo-label selection.

In this paper, we propose a novel approach called FINESSL to address the issues of aggregated biases and cognitive deviation caused by pre-trained foundation models simultaneously. To mitigate the problem of aggregated biases, FINESSL incorporates a balanced margin softmax to compensate for the learning of hard classes and encourage the model to generate more balanced pseudo-labels. The class margins are updated adaptively during the training process. Unlike previous SSL methods (Xu et al., 2021; Wang et al., 2022c; Chen et al., 2023), which manipulate confidence thresholds to achieve class-balanced pseudo-labels, FINESSL takes a different approach to rectify model preferences directly. To address the issue of cognitive deviation, FINESSL introduces the decoupled label smoothing to regularize the learning pace of an auxiliary classifier, which helps to align the model's confidence with the learning difficulties of tasks. Additionally, FINESSL employs a reweighting strategy to utilize all unlabeled data better. In doing so, it prevents the selection of inaccurate pseudo-labels by thresholding unreliable model confidence. As shown in Figure 2, FINESSL significantly improves the baselines in terms of both pseudo-label distribution and average model confidence. In summary, our key contributions are:

- We identify the aggregated biases and cognitive deviation problems caused by fine-tuning foundation models. We find that PEFT significantly outperforms FFT and LP in SSL tasks.

- We propose the balanced margin softmax to erase aggregated biases, decoupled label smoothing to reduce cognitive deviation, and a reweighting strategy to take full advantage of unlabeled data.
- We propose a general fine-tuning framework for SSL, which can smoothly incorporate various PEFT and leading SSL methods for improving performance.
- Extensive experimental analyses show that FINESSL achieves state-of-the-art performance on multiple SSL benchmarks. Moreover, FINESSL obtains significant reductions in the training cost.

2. Related Works

Semi-Supervised Learning. Modern SSL methods typically combine multiple techniques for leveraging unlabeled data, including entropy minimization (Grandvalet & Bengio, 2004), consistency regularization (Sohn et al., 2020), distribution alignment (Kim et al., 2020; Wei & Gan, 2023), and contrastive learning (Lee et al., 2022; Yang et al., 2022a). Most of them involve selecting reliable pseudo-labels during the training process. Initially, FixMatch selects pseudo-labels with confidence higher than a fixed threshold of 0.95. Recently, some works mention that a fixed threshold can be too strict for classes that are hard to learn, and propose to dynamically adjust the threshold as the training progresses. For example, FlexMatch (Zhang et al., 2021) assigns distinct thresholds to each class based on their levels of learning difficulty as well as the overall learning pace. FreeMatch (Wang et al., 2022c) integrates global and local threshold adjustments while incorporating class-fairness regularization, which encourages the model to make more diverse predictions. SoftMatch (Chen et al., 2023) achieves a quantity-quality trade-off through unified sample reweighting by employing a soft confidence threshold mechanism. In contrast, this paper encourages the model to explicitly produce class-balanced pseudo-labels rather than adjusting confidence thresholds.

SSL based on Foundation Models. A few prior efforts have been made to leverage pre-trained models to improve the performance of SSL tasks. As a representative, SimCLRv2 (Chen et al., 2020b) employs a pre-trained ResNet model (He et al., 2016) using self-supervised learning, and fine-tunes it using the labeled data. Zhou et al. (2018) delve comprehensively into the performance dynamics of SSL methods originating from pre-trained models, exploring their effectiveness across varying conditions. USB (Wang et al., 2022b) introduces pre-trained Transformers (Vaswani et al., 2017) into SSL, which reduces 80% training steps without hurting the performance. However, USB merely explores the FFT to adapt the pre-trained model, which is inferior to PEFT in SSL tasks. Recently, DebiasPL (Wang et al., 2022a) integrates large-scale multi-modal pre-trained model CLIP (Radford et al., 2021) into FixMatch by pseudo-labeling the discarded unlabeled instances with CLIP and effectively adjusts the distribution of pseudo-labels by logit adjustment (Menon et al., 2020) according to the distribution of pseudo-labels. However, how to effectively fine-tune the foundation models in SSL tasks remains underexplored.

3. Preliminaries

3.1. Background of Semi-Supervised Learning

First of all, we describe the problem setup and define notations frequently used throughout the paper. The goal of SSL is to learn a C-class classifier using a large number of unlabeled samples coupled with a few labeled ones. Let $\mathcal{D}^l = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ denote the labeled dataset of size N and $\mathcal{D}^u = \{\boldsymbol{u}_j\}_{j=1}^M$ denote the unlabeled dataset of size M. Here, $\boldsymbol{x}_i, \boldsymbol{u}_j \in \mathbb{R}^d$ are the d-dimensional feature vectors. For each labeled sample $\boldsymbol{x}_i$, we have access to its ground-truth class label $y_i \in \{0,1\}^C$. Typically, we have $N \ll M$ in SSL tasks. Then, our goal is to learn a model $f(\boldsymbol{x})$ parameterized by θ utilizing $\mathcal{D}^l$ and $\mathcal{D}^u$.

Modern SSL approaches usually have two components in their learning objectives, *i.e.,* the supervised and unsupervised losses (Sohn et al., 2020). Specifically, the standard cross-entropy $\mathcal{H}$ is utilized to optimize the supervised loss ℓ_s using labeled dataset:

$$\ell_s = \frac{1}{B}\sum_{i=1}^{B} \mathcal{H}(y_i, p(y \mid \boldsymbol{x}_i)) \tag{1}$$

where $p(y \mid \boldsymbol{x}_i) = \mathrm{Softmax}(f(\boldsymbol{x}_i;\theta))$ denotes the posterior probability of $\boldsymbol{x}_i$ being classified into class y, and B represents the batch size for labeled data.

Regarding the unsupervised loss, the majority of existing methods rely on pseudo-labeling (Lee et al., 2013; Berthelot et al., 2019; Sohn et al., 2020; Zhang et al., 2021; Wang et al., 2022c; Chen et al., 2023). To ensure the quality of generated pseudo-labels, these methods commonly employ a thresholding mechanism to choose pseudo-labels with higher confidence and impose consistency regularization on them. The widely used consistency regularizer proposed in FixMatch (Sohn et al., 2020) is defined as follows:

$$\ell_u = \frac{1}{\mu B}\sum_{j=1}^{\mu B} \mathbb{I}\left(\max\left(\boldsymbol{q}_j\right) \geq \tau\right) \mathcal{H}(\widehat{q}_j, p(y \mid \Omega(\boldsymbol{u}_j))) \tag{2}$$

where μ is a hyperparameter used to control the relative batch size ratio between labeled and unlabeled data, $\boldsymbol{q}_j = p(y \mid \omega(\boldsymbol{u}_j))$, and $\widehat{q}_j = \arg\max_k(q_{jk})$ is the pseudo-label of $\boldsymbol{u}_j$. τ is a threshold to select reliable pseudo-labels for calculating the loss and $\mathbb{I}(\cdot)$ is the indicator function. Since applying the consistency regularizer needs two different

views of images, we use $\omega(\cdot)$ and $\Omega(\cdot)$ to represent the weak and strong augmented versions of $\boldsymbol{u}_j$, respectively.

3.2. Fine-tuning Foundation Models

The prevailing trend in modern deep learning is the fine-tuning of large-scale pre-trained vision Transformers (Dosovitskiy et al., 2020; Radford et al., 2021; Jia et al., 2021) by leveraging rich semantic information to augment the performance of downstream tasks. Common fine-tuning strategies are FFT and LP, wherein the former involves adjusting the parameters of the entire pre-trained model, while the latter entails solely fine-tuning the final linear classifier of the neural networks. As depicted in Figure 1, FFT achieves inferior performance because the limited supervisory information available in SSL is not enough to guide the training of a big model. In comparison, LP performs reasonably well by freezing the pre-trained parameters except the linear classifier. However, since there exists a gap between pre-training datasets and our downstream SSL dataset, the learned representations can be suboptimal, which hinders the improvement of performance. Therefore, we propose to employ PEFT to effectively adapt the representations and the classifier. Although many PEFT methods can be utilized, we use VPT (Jia et al., 2022) as the default choice without loss of generality. The rationale of VPT is simple. The Transformers divide an input image into multiple patches of the same size and we denote $[\boldsymbol{c}_l; \boldsymbol{E}_l]$ as the input for l-th layer of ViT for simplicity. Here, $\boldsymbol{c}_l$ is the extra learnable token for classification and $\boldsymbol{E}_l$ denotes the collection of image patch embeddings. VPT prepends learnable prompts $\boldsymbol{P}_l$ at each layer to extend $[\boldsymbol{c}_l; \boldsymbol{E}_l]$ to $[\boldsymbol{c}_l; \boldsymbol{P}_l; \boldsymbol{E}_l]$. VPT has two variations: 1) VPT-shallow, where prompts are appended solely at the first layer; 2) VPT-deep, characterized by the extension of prompts to all layers. In this paper, we focus on VPT-deep for its stronger generalization capability. More analyses on different PEFT methods can be found in Section 5.3. To simplify the mathematical notation, we denote the parameters of the pre-trained foundation model as Θ and the parameters of the PEFT modules, including the final linear classifier, as θ.

4. The Proposed FINESSL Framework

In this section, we present the FINESSL framework in detail by introducing two of its core components, *i.e.,* balanced margin softmax and decoupled label smoothing.

4.1. Balanced Margin Softmax

Our empirical studies in Figure 2 highlight that employing dynamic thresholds for classes based on learning difficulties may not sufficiently mitigate aggregated biases. Instead of focusing on pseudo-label selection, we propose to explicitly intervene in the model training process. Drawing inspiration from previous works in class-imbalanced learning (Cao et al., 2019; Menon et al., 2020; Ye et al., 2023), we devise a balanced margin softmax loss with enforced dynamic class margins into the standard cross-entropy. For each training sample $(\boldsymbol{x}, y)$, we define:

$$\mathcal{H}_m(y, \boldsymbol{z}) = -\log \frac{e^{z_y}}{e^{z_y} + \sum_{k \neq y} e^{z_k + \alpha_t \Delta_y^t}} \tag{3}$$

where $k \in \{1, \ldots, C\}$, $\boldsymbol{z} = f(\boldsymbol{x}; \{\Theta, \theta\})$ denotes the output logits, and Δ_k^t is the margin for class k at t-th training iteration. α_t is a scaling hyperparameter of class margins which is adaptively updated during training. Instead of using a constant margin for all classes, the incorporation of class-specific margins helps improve inter-class discrimination and makes it more flexible to control the learning pace of classes. In Equation (3), we impose a large positive margin Δ_y^t when the model exhibits weak preference for predicting class y. This encourages the model to be more confident in its predictions in the next training iteration, gradually erasing the inherent bias of the foundation model and aggregated biases.

Next, we present a simple approach to automatically assign class margins based on model dynamics. We start by introducing the concept of learning pace for each class, which describes the model's capability of producing accurate pseudo-labels.

Definition 4.1. *Class Learning Pace.* The learning pace of the k-th class $\sigma_t(k)$ at timestamp t is defined as the number of unlabeled samples predicted as class k with confidence surpassing a constant ζ, *i.e.,*

$$\sigma_t(k) = \sum_{j=1}^{M} \mathbb{I}\left(\max\left(\boldsymbol{q}_j\right) \geq \zeta\right) \cdot \mathbb{I}\left(\arg\max\left(\boldsymbol{q}_j\right) = k\right). \tag{4}$$

The learning pace effectively reflects the model's abilities in each class, with classes having fewer predicted pseudo-labels above the threshold indicating slower learning paces, and vice versa. A similar concept of learning pace is explored in FlexMatch (Zhang et al., 2021), where confidence thresholds for classes are assigned based on a similar principle. In our implementation, we fix the threshold $\zeta = 0.7$ for simplicity, and the learning paces can be calculated by counting the number of confident pseudo-labels for each class. To normalize the learning paces across all classes, we define $\beta_t(y) = \frac{\sigma_t(y)}{\max_k \sigma_t(k)}$ and let $\Delta_y^t = 1 - \beta_t(y)$. In doing so, the model pays more attention to classes of slow learning paces.

Additionally, we use the parameter α_t in Equation (3) to control the magnitude of the margin Δ_y^t. Ideally, α_t is supposed to be 0 if the learning pace of all classes is approximately equal and the balanced margin softmax reduces to the standard cross-entropy. Conversely, a large α_t is desired to enhance the effect of the balanced margin softmax. Given a

base scaling parameter α, we propose to dynamically adjust its value at the t-th iteration based on learning paces via:

$$\alpha_t = (\max_k(\beta_t(k)) - \min_k(\beta_t(k)))\alpha. \quad (5)$$

Once obtaining α_t and Δ_y^t, we can improve the balancedness of pseudo-labels and reduce aggregated biases by minimizing Equation (3). While the learning paces are estimated based on unlabeled data, they effectively capture the learning dynamics of the model. By imposing balanced margins over the labeled data, we can further improve the model's generalization capabilities. To achieve this, we replace the standard cross-entropy with the balanced margin softmax for both the supervised loss and consistency regularization.

4.2. Decoupled Label Smoothing

While the balanced margin softmax alleviates the imbalance issue caused by pre-trained foundation models, the presence of cognitive deviation in the training process can still lead to suboptimal performance in selecting reliable pseudo-labels based on confidence thresholds. To address this, we propose a simple yet effective approach called decoupled label smoothing to overcome the mismatched model confidence and task difficulties. Label smoothing is a well-known technique that has been successfully applied across a range of tasks to regularize model training and mitigate overconfidence issues (Müller et al., 2019; Zhong et al., 2021). In a nutshell, label smoothing regularizes the model training by softening the ground-truth labels. However, directly applying label smoothing to the learning objective can negatively impact representation learning, as demonstrated in our ablation studies. To overcome this limitation, we introduce an auxiliary classifier $f_{aux}(\boldsymbol{x})$, which is a single fully connected layer appended to the feature extractor. To distinguish $f_{aux}(\boldsymbol{x})$ from $f(\boldsymbol{x})$, we denote its output logits using $f_{aux}((\boldsymbol{x}); \{\Theta, \theta'\})$, where θ' contains parameters for PEFT modules and the auxiliary classifier. This auxiliary classifier is learned by minimizing the supervised loss ℓ_s and the soft version of consistency regularization:

$$\ell_u^{ls} = \frac{1}{\mu B}\sum_{j=1}^{\mu B} \mathcal{H}_m(\widetilde{\boldsymbol{q}}_j, f_{aux}(\Omega(\boldsymbol{u}_j); \{\Theta, \theta'\})) \quad (6)$$

where $\widetilde{\boldsymbol{q}}_j$ denotes the continuous pseudo-label vector for $\boldsymbol{u}_j$, $\widetilde{q}_{jk}$ equals $(1-\lambda) + \lambda/C$ if $k = \widehat{q}_j$, otherwise λ/C. $\lambda \in (0, 1)$ is the label smoothing parameter.

It is crucial to note that the auxiliary classifier is intentionally decoupled from the main branch of neural networks. This ensures that the gradients of the auxiliary classifier's loss function do not propagate to the earlier layers during backpropagation. As a result, this prevents label smoothing from damaging the representation learning and allows it to focus solely on generating discriminative weights between correct and incorrect pseudo-labels.

Algorithm 1 The Proposed FINESSL

Input: labeled dataset $\mathcal{D}^l$ and unlabeled dataset $\mathcal{D}^u$, pre-trained foundation model, hyperparameters $\{\zeta, \alpha, \lambda\}$, number of iterations T
for $t = 1$ **to** T **do**
  Update learning pace $\sigma_t(y)$ by Equation (4)
  if $\max_y(\sigma_t(y)) > 0$ **then**
    $\beta_t(y) = \frac{\sigma_t(y)}{\max_k \sigma_t(k)}$ ▷ Normalization
    $\Delta_y^t = 1 - \beta_t(y)$ ▷ Obtain class-specific margin
  else
    $\Delta_y^t = 1$
  end if
  Compute scaling parameter α_t by Equation (5)
  Compute unlabeled sample weights $\psi(\boldsymbol{u})$
  Compute ℓ_s^m and $\widetilde{\ell}_u^m$ for the main branch
  Compute ℓ_s and ℓ_u^{ls} for the auxiliary classifier
  Compute the total loss $\mathcal{L}$ by Equation (8)
  Update θ based on $\nabla\mathcal{L}$ using SGD
end for

Next, we aim to resolve another substantial disadvantage of pseudo-label selection, which merely utilizes unlabeled data of high confidence, and low-confidence data are discarded. Obtaining discriminative predictive probabilities from the auxiliary classifier, we take full advantage of unlabeled data to optimize the model by incorporating sample reweighting into consistency regularization. Specifically, we define the weighted consistency regularizer as follows:

$$\widetilde{\ell}_u^m = \frac{1}{\mu B}\sum_{j=1}^{\mu B} \psi(\boldsymbol{u}_j)\mathcal{H}_m(\widehat{q}_j, f(\Omega(\boldsymbol{u}_j); \{\Theta, \theta\})) \quad (7)$$

where $\psi(\boldsymbol{u}_j) = \gamma \cdot \max(p_{aux}(y \mid \omega(\boldsymbol{u}_j))$ represents the importance weight for the j-th unlabeled sample, and γ is a tunable hyperparameter. The output probabilities $p_{aux}(y \mid \omega(\boldsymbol{u}_j)) = \mathrm{Softmax}(f_{aux}(\omega(\boldsymbol{u}_j); \{\Theta, \theta'\}))$. Notice that the auxiliary classifier is solely employed for generating sample weights, which facilitates the learning of the main branch. In return, better representations learned by the main branch improve the training of the auxiliary classifier. We analyze model confidence used for sample reweighting in Appendix C.3.

To sum up, the overall loss for the proposed method comprises two parts, *i.e.,* losses for the main branch and auxiliary classifier, which are formulated as follows:

$$\mathcal{L} = \underbrace{\ell_s^m + \widetilde{\ell}_u^m}_{\text{main branch}} + \underbrace{\ell_s + \ell_u^{ls}}_{\text{auxiliary classifier}} . \quad (8)$$

Here, we denote ℓ_s^m as the balanced margin softmax loss over labeled data. We present the pseudo-code of the proposed FINESSL algorithm in Algorithm 1.

Table 1. Performance comparisons on CIFAR-10, CIFAR-100, and FOOD-101 datasets: mean ± std of accuracy over 3 trials are reported. The best performance is highlighted in **bold** and the second-best performance is underlined.

	CIFAR-10			CIFAR-100			FOOD-101		
SETTINGS	N1	N2	N4	N4	N25	N100	N2	N4	N10
SUPERVISED	68.41 ±6.60	78.54 ±4.94	86.94 ±4.17	64.18 ±0.33	80.23 ±0.14	84.45 ±0.29	61.42 ±0.21	73.74 ±0.22	82.10 ±0.13
PL	48.45 ±2.99	74.33 ±10.97	92.80 ±0.38	66.17 ±0.30	81.45 ±0.30	84.65 ±0.20	58.22 ±1.18	77.57 ±0.93	84.62 ±0.12
FIXMATCH	60.27 ±1.89	87.31 ±0.10	**97.43** ±0.14	77.10 ±1.04	84.05 ±0.16	86.17 ±0.03	74.92 ±1.09	84.93 ±0.70	88.72 ±0.02
FLEXMATCH	61.71 ±15.07	92.71 ±7.31	97.41 ±0.10	79.19 ±0.23	83.77 ±0.13	86.23 ±0.25	80.96 ±1.68	88.51 ±0.13	89.24 ±0.11
FREEMATCH	60.41 ±1.78	87.04 ±0.06	97.26 ±0.14	76.19 ±0.82	83.60 ±0.16	86.34 ±0.30	74.31 ±4.56	84.69 ±1.11	88.97 ±0.24
SOFTMATCH	59.74 ±11.44	90.72 ±5.28	97.35 ±0.06	75.94 ±0.46	83.85 ±0.06	86.31 ±0.07	75.82 ±5.24	85.05 ±1.05	89.01 ±0.08
DEBIASPL	78.30 ±16.85	96.83 ±0.14	97.39 ±0.06	79.57 ±0.32	84.01 ±0.07	86.16 ±0.13	86.51 ±0.74	88.60 ±0.34	89.32 ±0.17
FINESSL	**96.15** ±0.13	**96.87** ±0.10	97.39 ±0.05	**80.44** ±0.24	**84.51** ±0.01	**86.66** ±0.18	**87.04** ±0.43	**89.28** ±0.14	**89.69** ±0.15

5. Experiments

5.1. Experiment Setup

Datasets and Evaluation Metrics. We conduct extensive experiments on five publicly available datasets to evaluate the performance of FINESSL. The datasets are CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), FOOD-101 (Bossard et al., 2014), Semi-Aves (Su et al., 2021), and ImageNet (Deng et al., 2009). For each dataset, we consider different numbers of labeled samples to cover various SSL scenarios. Following previous works (Wang et al., 2022c; Chen et al., 2023), we select $\{1, 2, 4\}$ labeled samples per class for CIFAR-10, $\{4, 25, 100\}$ labeled samples per class for CIFAR-100, and $\{2, 4, 10\}$ labeled samples per class for FOOD-101. For Semi-Aves, we consider two scenarios of unlabeled data, *i.e.,* unlabeled data consists of either in-distribution samples or mixed in-distribution and OOD samples. Finally, for ImageNet, we consider the settings where 1% and 10% training samples are labeled following SimCLR (Chen et al., 2020a). In our experiments, we compare FINESSL with existing state-of-the-art SSL methods, including Pseudo-Label (PL) (Lee et al., 2013), FixMatch (Sohn et al., 2020), FlexMatch (Zhang et al., 2021), FreeMatch (Wang et al., 2022c), SoftMatch (Chen et al., 2023), and DebiasPL (Wang et al., 2022a). All competing methods are implemented in our proposed framework. The classification performance is evaluated using the Top-1 accuracy on the test dataset.

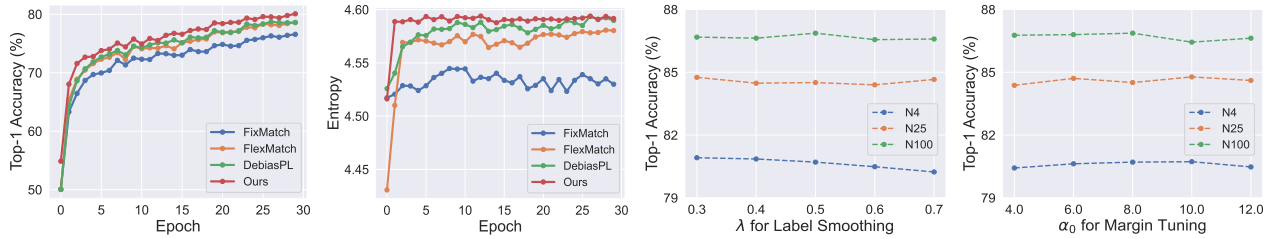
Implementation details. We use the publicly available CLIP model[1] as our backbone. Model parameters of CLIP are all frozen, and only the image encoder is used during training and inference. In our experiments, we use VPT (Jia et al., 2022) by default to fine-tune the CLIP model due to its effectiveness and efficiency. The length of learnable prompts is set to 50. We also provide the results for more PEFT methods in Section 5.3. We employ the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.03, utilizing a batch size of 32 alongside a weight decay set at 5×10^{-4}, and a momentum factor of 0.9. We fine-tune the model for 30 epochs, with each epoch comprising 500 steps. It is noteworthy that all competing methods share the same setting of those hyperparameters. All experiments are conducted in PyTorch with a single NVIDIA RTX 3090 24GB GPU. More details can be found in Appendix A.

[1] https://github.com/mlfoundations/open_clip

5.2. Main Results

Results on standard datasets. Table 1 reports the test accuracy of competing methods over CIFAR-10, CIFAR-100, and FOOD-101 datasets. Overall, FINESSL achieves superior performance, substantially outperforming the other seven methods in most datasets and settings. FINESSL presents a relatively low performance in N4 setting on CIFAR-10. This is partly because the task is simple such that several methods achieve comparably high performance (Wang et al., 2022a). Notably, FINESSL outperforms competing methods by nearly 18% in accuracy with only 1 labeled sample per class on CIFAR-10, showing its robust performance in scenarios where labels are extremely scarce. The weak performance of Pseudo-Labeling and FixMatch in low-label regimes can be attributed to the biases and overconfidence issues of the model. By dynamically adjusting the confidence threshold, FlexMatch and FreeMatch achieve better results. DebiasPL addresses the class-imbalanced pseudo-labels and achieves comparable results with FINESSL in two settings on CIFAR-10. However, in other cases, FINESSL surpasses DebiasPL by an average of 3% in accuracy, showing that FINESSL can perform much more accurately across different settings.

Results on complex datasets. To evaluate the generalization ability of our model, we further examine FINESSL on two more challenging datasets, *i.e.,* Semi-Aves and ImageNet. Table 2 and Table 3 show the results for FINESSL and five competeing methods. Semi-Aves (Su et al., 2021) is a dataset of hundreds of bird species. Data exhibits considerable class imbalance, and the unlabeled data contains



(a) (b) (c) (d)

Figure 3. (3a and 3b): The accuracy and entropy of pseudo-labels for FixMatch, FlexMatch, DebiasPL, and FINESSL on CIFAR100 with 4 labeled data per class. (3c and 3d): The sensitivity of λ and α_0 under various settings on CIFAR-100.

Table 2. Top-1 accuracy for Semi-Aves. $\mathcal{D}^u_{in}$ denotes that unlabeled data are sampled from the same classes as labeled data, and $\mathcal{D}^u_{out}$ contains OOD samples.

SETTINGS	$\mathcal{D}^u = \mathcal{D}^u_{in}$	$\mathcal{D}^u = \mathcal{D}^u_{in} \cup \mathcal{D}^u_{out}$
FIXMATCH	65.52 ±0.30	60.15 ±0.16
FLEXMATCH	64.60 ±0.25	57.96 ±0.11
FREEMATCH	64.57 ±0.23	58.01 ±0.38
SOFTMATCH	64.75 ±0.27	57.79 ±0.52
DEBIASPL	66.82 ±0.35	60.85 ±0.37
FINESSL	**67.25** ±0.11	**61.12** ±0.07

Table 3. Top-1 accuracy for ImageNet. We consider two settings, distinguished by 1% and 10% labeled data.

SETTINGS	1% LABELED	10% LABELED
FIXMATCH	73.19	78.77
FLEXMATCH	73.21	78.71
FREEMATCH	72.94	78.49
SOFTMATCH	72.92	78.56
DEBIASPL	73.58	78.72
FINESSL	**74.22**	**79.21**

images from novel classes, which makes the task particularly challenging. We consider both settings where the unlabeled data comes from the same classes with labeled data or contains new classes. FixMatch achieves much better performance than its three sophisticated counterparts, especially when OOD data is present in the unlabeled data. This is attributed to the selection of more OOD samples by decreasing the threshold during training. Our method FINESSL outperforms FixMatch and DebiasPL by 1.4% and 0.35% in accuracy on average. For the ImageNet dataset, we randomly choose 1% and 10% images as labeled data following SimCLR (Chen et al., 2020a). Overall, all five competing methods achieve similar performance. By employing FINESSL, we observe a 0.57% relative performance increase over DebiasPL. This demonstrates the superior performance of FINESSL in large-scale datasets.

5.3. A Deeper Look into FINESSL

In this subsection, we conduct qualitative analyses of FINESSL in comparison to other SSL methods. Additionally, we examine the sensitivity of hyperparameters to show the robustness of our method. The comparative evaluations with alternative PEFT strategies and the analysis of training costs are also included.

Quality of Pseudo-Labels. From Figure 3a, it is noteworthy that the pseudo-labels generated by FINESSL consistently exhibit higher accuracy during training compared to other methods, and it is obvious that the accuracy of pseudo-labels plays a pivotal role in enhancing the performance of SSL tasks (Chen et al., 2023). In addition, FINESSL shows higher entropy in the pseudo-label distribution during training, indicating that our method effectively mitigates aggregated biases and achieves a more balanced distribution of pseudo-labels.

Hyperparameter Sensitivity. Despite the commendable performance achieved by our method, the inclusion of some significant parameters prompts us to question whether the outstanding results are attributed to parameter tuning. Therefore, we are interested in exploring the robustness of these parameters and their impact on model performance. Specifically, as illustrated in Figures 3c and 3d, we observe that variations in both λ and α do not result in significant fluctuations in the performance of FINESSL across three settings on CIFAR-100, indicating that the efficacy of FINESSL stems from its thoughtful design rather than an extensive reliance on parameter tuning. More analysis for hyperparameter sensitivity can be found in Appendix C.1.

Table 4. Results for PEFT strategies on FINESSL.

	CIFAR-100			SEMI-AVES	
SETTINGS	N4	N25	N100	$\mathcal{D}^u_{in}$	$\mathcal{D}^u_{in} \cup \mathcal{D}^u_{out}$
VPT	80.44	84.51	86.66	67.25	61.12
LORA	79.71	85.56	87.29	68.30	61.13
ADAPTER	79.91	84.24	86.25	65.94	60.04
ADAPTFORMER	81.02	85.44	87.50	68.39	60.73

More PEFT Strategies. While we use VPT for most experiments, it is noteworthy that various PEFT strategies are

Table 5. The comparison for training time consumption (in seconds) on the CIFAR-100 dataset.

	PER STEP	# STEPS	TOTAL
FIXMATCH (WIDE RESNET)	0.053	1024×1024	55574.5
FIXMATCH (VPT)	0.616	30×500	9240.0
FINESSL (VPT)	0.642	30×500	9630.0

Table 6. Ablation studies. We study the impact of core components of FINESSL on CIFAR-100 and Semi-Aves datasets.

	CIFAR-100			SEMI-AVES	
SETTINGS	N4	N25	N100	$\mathcal{D}_{in}^u$	$\mathcal{D}_{in}^u \cup \mathcal{D}_{out}^u$
FINESSL	80.44	84.51	86.66	67.25	61.12
W/O BMS	76.70	83.71	86.21	65.83	59.91
W/O MARGIN IN ℓ_s^m	80.52	84.34	86.60	66.88	58.94
W/O DYNAMIC α	80.30	84.34	86.75	67.10	61.00
W/O DLS	80.29	84.36	86.36	67.02	58.58
W/O LS	80.48	84.40	86.49	67.22	59.49
W/O DETACH	79.85	84.23	86.78	67.11	61.05

applicable. In Table 4, Adaptformer (Chen et al., 2022b) exhibits an average performance surpassing VPT by up to 0.62%. Another widely adopted PEFT strategy, Lora (Hu et al., 2021), presents significant advantages in scenarios with a larger quantity of labeled data. Nevertheless, its performance is 0.73% inferior to VPT in the N4 setting of CIFAR-100, which is characterized by a limited amount of labeled data. Adapter (Houlsby et al., 2019) also performs reasonably well, yet they show an averaged 0.72% decline in comparison to VPT. While it is acknowledged that Adaptformer yields superior performance in Table 4, we choose VPT as the fine-tuning strategy in our primary results, considering its broader impact and potential applications (Han et al., 2024).

Time Consumption. Our method stands out not just for its superior and robust performance but also for its training efficiency. Table 5 shows that our proposed method FINESSL only requires about $1/6$ training time to converge compared with the consumption of training Wide ResNet from scratch. Additionally, FixMatch implemented in our framework, *i.e.,* fine-tuning the foundation model via VPT, exhibits comparable time consumption with FINESSL which increases negligible training costs by introducing an auxiliary linear classifier to the original neural networks. The efficient framework we propose can be beneficial for promoting the deployment of SSL in real-world applications.

5.4. Ablation Analysis

To better understand FINESSL, we tease apart the factors that contribute significantly to its success in Table 6.

Impact of balanced margin softmax. The balanced margin softmax loss (denoted by "BMS" in Table 6) serves as a crucial component for alleviating aggregated biases in our method. Upon removing the dynamic adjustments to the margins, we notice a significant performance decline across all settings, and particularly noteworthy is the substantial 3.74% decrease in performance observed in the "N4" setting on CIFAR-100, where the labeled data is quite limited. This shows that BMS can effectively erase model biases.

Impact of balanced margin softmax in supervised loss. In our method, we also incorporate dynamically computed margins based on unlabeled data into the labeled loss $\mathcal{L}_s^m$, thereby enhancing the balance and stability of the model. The exclusion of the margins introduced in $\mathcal{L}_s^m$ resulted in an average performance drop of 1.28% on Semi-Aves, emphasizing the necessity of incorporating the margins into the labeled loss.

Impact of dynamic scaling parameter α. The dynamic scaling parameter α_t in Equation (5) plays a crucial role in adjusting the magnitude of class margins. To examine its impact, we set α_t to a constant value in this experiment. From Table 6, we observe a decrease in accuracy across most settings. This suggests that the dynamic adjustment of α is essential for improving the model's ability to handle imbalanced pseudo-label distributions.

Impact of decoupled label smoothing. In this experiment, we remove the auxiliary classifier and reweight unlabeled data using confidence scores generated by the main branch. We observe a consistent performance drop across all settings, averaging 0.67%. Especially in the presence of OOD data on Semi-Aves, the performance decreases by 2.54%. This highlights that discriminative model probabilities play an important role in sample reweighting.

Impact of label smoothing in the auxiliary classifier. We employ label smoothing for learning the auxiliary classifier to address cognitive deviation. Removing label smoothing, which means training the auxiliary classifier using hard pseudo-labels directly, leads to performance deterioration in most settings. Particularly, we observe a 1.63% decrease in accuracy on Semi-Aves when the unlabeled data contains OOD samples.

Impact of the auxiliary classifier on representations. In FINESSL, the auxiliary classifier is detached to prevent hurting the representation learning. We investigate the impact of the auxiliary branch on representation learning by allowing gradient backpropagation to the former layers of the model. The results are denoted by "w/o Detach" in Table 6. We observe a substantial performance drop when the number of labeled samples is extremely scarce, *e.g.,* a 0.59% decrease in the "N2" setting on CIFAR-100. This shows that mitigating overconfidence may hurt representations.

5.5. Tasks with OOD Samples

In Section 5.4, we observe that DLS exhibits significant performance improvement on Semi-Aves with OOD samples.

Table 7. Results for fine-tuning pre-trained ResNet CLIP on CIFAR-100.

STRATEGIES	BN TUNING			BITFIT			SSF		
SETTINGS	N4	N25	N100	N4	N25	N100	N4	N25	N100
FIXMATCH	37.49	62.98	71.48	52.67	66.15	70.19	36.49	53.47	**60.44**
FLEXMATCH	38.22	51.96	69.52	**55.69**	66.33	69.58	36.42	51.89	57.49
FREEMATCH	30.51	61.48	70.11	53.28	66.52	69.07	36.29	52.01	57.35
SOFTMATCH	28.70	61.32	68.80	53.34	66.46	69.38	36.11	52.09	57.51
DEBIASPL	42.72	60.62	69.76	52.90	66.62	70.34	38.83	52.54	58.44
FINESSL	**48.32**	**65.87**	**72.38**	55.21	**67.06**	**70.60**	**39.40**	**54.14**	59.51

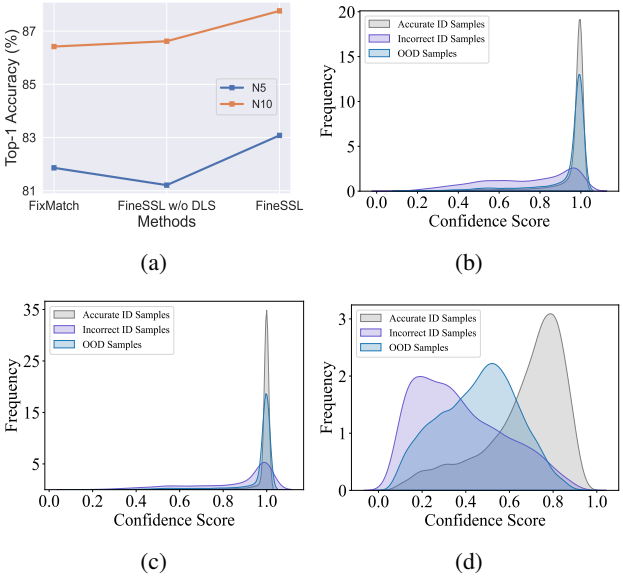


Figure 4. (4a): The accuracy for *N5* and *N10* OpenSSL setting on CIFAR-100 for different methods. (4b to 4d): The distribution of confidence score for ID and OOD samples of FixMatch, FINESSL w/o DLS, and FINESSL for *N5* OpenSSL setting.

Notably, the presence of OOD samples is prevalent in SSL (Cao et al., 2022; Guo et al., 2022). Motivated by this finding, we conduct further experiments to investigate the role of DLS in handling OOD samples. Specifically, following OpenSSL settings in (Cao et al., 2022), we divide the classes on CIFAR-100 into 50% seen classes (ID samples) and 50% novel classes (OOD samples). We select 5 and 10 labeled samples per seen class (i.e., N5 and N10 settings), with the remaining samples unlabeled. We show the accuracy for seen classes of the test set in Figure 4a. The results clearly demonstrate that removing the DLS leads to an average performance decline of 1.87%, indicating the effectiveness of DLS in mitigating the impact of OOD samples.

Why DLS can benefit tasks with OOD samples? We show the confidence score distribution of accurate, incorrect, and OOD samples in Figures 4b to 4d. For FixMatch and FINESSL w/o DLS, we observe significant overlap in the distribution regions of the three types of samples, suggesting that the model faces challenges in distinguishing them using confidence scores. However for FINESSL, there exists much more clearer confidence boundary, which enables the model to assign more reliable weights to each sample, leading to performance improvements for ID and OOD samples.

5.6. Results for pre-trained ResNet model

Although this paper primarily focuses on improving the CLIP model based on ViT structure, we still conduct experiments using the pre-trained ResNet CLIP model. As VPT and other PEFT strategies are not applicable to ResNet, we employ BN Tuning (fine-tuning batchnorm parameters), BitFit (Zaken et al., 2021) (fine-tuning solely the bias terms), and SSF (Lian et al., 2022) for fine-tuning. The results in Table 7 consistently demonstrate the superiority of FINESSL over other comparative methods. Notably, FINESSL achieves an average improvement of 2.19% compared to DebiasPL, highlighting its significant performance advantages in fine-tuning the ResNet structure model.

6. Conclusion

This paper proposes a novel SSL approach FINESSL to improve the generalization performance by efficiently fine-tuning pre-trained foundation models. We identify the inherent aggregated biases and cognitive deviation issues in the foundation model, which hinder the generation and selection of reliable pseudo-labels. We introduce the balanced margin softmax to erase the model biases across different classes. Further, to take full advantage of unlabeled data, we devise a decoupled label smoothing to regularize model confidence and reweight unlabeled samples. Extensive experimental results on five public datasets demonstrate that FINESSL achieves superior performance over the previous state of the arts and decreases the training cost by six times. Moreover, The framework we implemented can smoothly incorporate many fine-tuning and SSL methods.

Acknowledgements

This work was supported by the National Science Foundation of China (62206049), and the Big Data Computing Center of Southeast University

Impact Statement

This paper presents work whose goal is to advance the field of machine learning. Specifically, we propose a new semi-supervised learning algorithm to improve the performance on low-label regimes by fine-tuning pre-trained foundation models. In classification tasks, since pre-trained models can be biased toward certain groups of classes, this paper makes a substantial attempt to erase such inherent prediction biases. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., and McGuinness, K. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *IJCNN*, pp. 1–8, 2020.

Bachman, P., Alsharif, O., and Precup, D. Learning with pseudo-ensembles. *NeurIPS*, 27, 2014.

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. Mixmatch: A holistic approach to semi-supervised learning. *NeurIPS*, 32, 2019.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101–mining discriminative components with random forests. In *ECCV*, pp. 446–461, 2014.

Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. *NeurIPS*, 32, 2019.

Cao, K., Brbic, M., and Leskovec, J. Open-world semi-supervised learning. In *International Conference on Learning Representations*, 2022.

Chan, A., Alaa, A., Qian, Z., and Van Der Schaar, M. Unlabelled data improves bayesian uncertainty calibration under covariate shift. In *ICML*, pp. 1392–1402, 2020.

Chen, G., Liu, F., Meng, Z., and Liang, S. Revisiting parameter-efficient tuning: Are we really there yet? *arXiv preprint*, 2022a.

Chen, H., Tao, R., Fan, Y., Wang, Y., Wang, J., Schiele, B., Xie, X., Raj, B., and Savvides, M. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. *arXiv preprint*, 2023.

Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., and Luo, P. Adaptformer: Adapting vision transformers for scalable visual recognition. *NeurIPS*, 35:16664–16678, 2022b.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *ICML*, pp. 1597–1607, 2020a.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. E. Big self-supervised models are strong semi-supervised learners. *NeurIPS*, 33:22243–22255, 2020b.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint*, 2020.

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

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Guo, L.-Z., Zhang, Y.-G., Wu, Z.-F., Shao, J.-J., and Li, Y.-F. Robust semi-supervised learning when not all classes have labels. In *Neural Information Processing Systems*, 2022.

Han, C., Wang, Q., Cui, Y., Wang, W., Huang, L., Qi, S., and Liu, D. Facing the elephant in the room: Visual prompt tuning or full finetuning? *arXiv preprint*, 2024.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *ICML*, pp. 2790–2799, 2019.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint*, 2021.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, pp. 4904–4916, 2021.

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, 2022.

Kim, J., Hur, Y., Park, S., Yang, E., Hwang, S. J., and Shin, J. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. *NeurIPS*, 33: 14567–14579, 2020.

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

Kumar, A., Raghunathan, A., Jones, R., Ma, T., and Liang, P. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint*, 2022.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. *arXiv preprint*, 2016.

Lee, D., Kim, S., Kim, I., Cheon, Y., Cho, M., and Han, W.-S. Contrastive regularization for semi-supervised learning. In *CVPR*, pp. 3911–3920, 2022.

Lee, D.-H. et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop*, pp. 896, 2013.

Lian, D., Zhou, D., Feng, J., and Wang, X. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35:109–123, 2022.

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., and Raffel, C. A. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *NeurIPS*, 35:1950–1965, 2022.

Menghini, C., Delworth, A., and Bach, S. Enhancing clip with clip: Exploring pseudolabeling for limited-label prompt tuning. *Advances in Neural Information Processing Systems*, 36:60984–61007, 2023.

Menon, A. K., Jayasumana, S., Rawat, A. S., Jain, H., Veit, A., and Kumar, S. Long-tail learning via logit adjustment. *arXiv preprint*, 2020.

Müller, R., Kornblith, S., and Hinton, G. E. When does label smoothing help? *NeurIPS*, 32, 2019.

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

Rizve, M. N., Duarte, K., Rawat, Y. S., and Shah, M. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint*, 2021.

Sajjadi, M., Javanmardi, M., and Tasdizen, T. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *NeurIPS*, 29, 2016.

Shi, J.-X., Wei, T., Zhou, Z., Han, X.-Y., Shao, J.-J., and Li, Y.-F. Parameter-efficient long-tailed recognition. *arXiv preprint*, 2023.

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. *NeurIPS*, 33:596–608, 2020.

Su, J.-C., Cheng, Z., and Maji, S. A realistic evaluation of semi-supervised learning for fine-grained classification. In *CVPR*, pp. 12966–12975, 2021.

Thulasidasan, S., Chennupati, G., Bilmes, J. A., Bhattacharya, T., and Michalak, S. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *NeurIPS*, 32, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *NeurIPS*, 30, 2017.

Wang, X., Gao, J., Long, M., and Wang, J. Self-tuning for data-efficient deep learning. In *ICML*, pp. 10738–10748, 2021.

Wang, X., Wu, Z., Lian, L., and Yu, S. X. Debiased learning from naturally imbalanced pseudo-labels. In *CVPR*, pp. 14647–14657, 2022a.

Wang, Y., Chen, H., Fan, Y., Sun, W., Tao, R., Hou, W., Wang, R., Yang, L., Zhou, Z., Guo, L.-Z., et al. Usb: A unified semi-supervised learning benchmark for classification. *NeurIPS*, 35:3938–3961, 2022b.

Wang, Y., Chen, H., Heng, Q., Hou, W., Fan, Y., Wu, Z., Wang, J., Savvides, M., Shinozaki, T., Raj, B., et al. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv preprint*, 2022c.

Wei, T. and Gan, K. Towards realistic long-tailed semi-supervised learning: Consistency is all you need. In *CVPR*, pp. 3469–3478, 2023.

Weng, Z., Yang, X., Li, A., Wu, Z., and Jiang, Y.-G. Semi-supervised vision transformers. In *European conference on computer vision*, pp. 605–620. Springer, 2022.

Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. Self-training with noisy student improves imagenet classification. In *CVPR*, pp. 10687–10698, 2020.

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, 2021.

Yang, F., Wu, K., Zhang, S., Jiang, G., Liu, Y., Zheng, F., Zhang, W., Wang, C., and Zeng, L. Class-aware contrastive semi-supervised learning. In *CVPR*, pp. 14421–14430, 2022a.

Yang, Z., Ding, M., Guo, Y., Lv, Q., and Tang, J. Parameter-efficient tuning makes a good classification head. *arXiv preprint*, 2022b.

Ye, B., Gan, K., Wei, T., and Zhang, M.-L. Bridging the gap: Learning pace synchronization for open-world semi-supervised learning. *arXiv preprint*, 2023.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zaken, E. B., Ravfogel, S., and Goldberg, Y. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021.

Zhang, B., Wang, Y., Hou, W., Wu, H., Wang, J., Okumura, M., and Shinozaki, T. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *NeurIPS*, 34:18408–18419, 2021.

Zhong, Z., Cui, J., Liu, S., and Jia, J. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16489–16498, 2021.

Zhou, H.-Y., Oliver, A., Wu, J., and Zheng, Y. When semi-supervised learning meets transfer learning: Training strategies, models and datasets. *arXiv preprint*, 2018.

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

Zhu, B., Tang, K., Sun, Q., and Zhang, H. Generalized logit adjustment: Calibrating fine-tuned models by removing label bias in foundation models. *arXiv preprint*, 2023.

A. Implementation Details

A.1. Implementation Details

To ensure the reproducibility of our method's performance, we present experimental details below, and default configs for the experiments are shown in Table 8. For most settings, a batch size of 32 is utilized. However, due to the limited availability of labeled data in the CIFAR-10 "N1" and "N2" settings, batch sizes of 8 and 16 are chosen, respectively. The relative batch size ratio μ between unlabeled and labeled data is set to 2 for the ImageNet dataset, and it is consistently set to 1 for the remaining datasets. Note that training is performed for 30 epochs with each epoch consisting of 500 steps, which is significantly less than the 1024×1024 steps employed by previous training from scratch methods (Sohn et al., 2020; Wang et al., 2022c). For certain datasets, such as ImageNet, additional training epochs and steps are expected to yield further performance improvements, which we intend to explore in the future. For competing methods implemented in our general fine-tuning framework for SSL, we find that the performance achieved with a higher threshold, specifically 0.95, is unsatisfactory. Hence, in our experiments, we adopt a smaller threshold of 0.7 for them, which can achieve superior results.

Table 8. The default parameter configs employed in the experiments.

Configuration	Default Value
Optimizer	SGD
Learning rate	0.03
Scheduler for lr	Cosine decay
Weight decay	5×10^{-4}
Momentum factor	0.9
Batch Size	32
Model	CLIP-ViT
Epochs	30
Steps	500
μ	1
PEFT strategy	VPT deep
VPT length	50
λ	0.5
α	8.0
γ	3.0

B. Additional Experiments

B.1. Long-Tailed Semi-Supervised Learning

Most of the experiments outlined above assume a uniform distribution of labeled and unlabeled data, whereas data in the real world often exhibits a long-tailed distribution. Therefore, we integrate some long-tailed semi-supervised learning (LTSSL) settings to further validate the effectiveness of FINESSL. It is noteworthy that, given our discovery of the existence of aggregated biases, combining long-tailed data distributions makes these settings particularly challenging.

In Table 9, $N1$ and M_1 mean the number of samples for the first class in labeled and unlabeled data, and typically, we have $N_1 \geq N_2 \geq \cdots \geq N_C$ and $M_1 \geq M_2 \geq \cdots \geq M_C$. We use ρ to represent the imbalance ratio for labeled and unlabeled data, where $\rho = \frac{N_1}{N_C}$ and $N_k = N_1 \times \rho^{-\frac{k-1}{C-1}}$ for class $k \in [1, C]$. The results indicate that FINESSL exhibits a significant advantage over all comparative methods, averaging a 2.95% performance improvement over SoftMatch, which suggests the superior robustness of our method against data imbalance. It is noteworthy that DebiasPL also demonstrates favorable performance in LTSSL settings, which is attributed to its incorporation of techniques specifically addressing long-tailed challenges (Menon et al., 2020). Nevertheless, FINESSL still surpasses DebiasPL across the majority of settings.

B.2. Can FFT Behave More Effective?

In Figure 1, we observe that FFT achieves inferior performance in semi-supervised learning settings, particularly when labeled data is extremely scarce. However, we find that a simple modification can enhance the performance of FFT.

Table 9. Comparison for long-tailed semi-supervised learning settings on CIFAR-100 dataset.

SETTINGS	$N_1 = 50$ $M_1 = 400$		$N_1 = 150$ $M_1 = 300$	
ρ	10	20	10	20
FIXMATCH	78.94	72.04	81.63	78.21
FLEXMATCH	78.81	73.36	81.75	78.53
FREEMATCH	79.05	72.82	81.49	78.42
SOFTMATCH	78.42	72.39	81.52	78.90
DEBIASPL	81.60	**78.23**	83.12	80.89
OURS	**81.63**	77.75	**83.44**	**81.35**

Specifically, (Kumar et al., 2022) argues that a two-step strategy of linear probing and then full fine-tuning (LP-FT), can achieve great improvement in ID and OOD accuracy. Interestingly, LP-FT can also achieve excellent performance in SSL settings. From Table 10, FixMatch with LP-FT can achieve an averaged 48.99% enhancement in performance compared to the FFT of FixMatch.

In addition, following PEL (Shi et al., 2023), we find that semantic-aware initialization for the linear classifier can also help FFT achieve better performance. Specifically, we employ textual features associated with class labels to initialize the classifier weights. This straightforward strategy facilitates the classifier to attain a more optimal initial state without incurring additional computational overhead, which accelerates model convergence and maintains a more stable training process. In Table 10, the performance of FFT† closely approximates that of VPT, exhibiting an average superiority of 47.30% over FFT. It is noteworthy that we search the learning rate from $\{0.0005, 0.0003, 0.0001, 0.00005, 0.00003, 0.00001\}$ and choose the best performance for all experiments related to FFT. In addition, it is obvious that FINESSL outperforms FixMatch across all fine-tuning strategies listed in Table 10, averaging 4.59%, indicating the broad applicability and effectiveness of our method.

Overall, LP-FT and FFT† both yield a well-suited weight initialization for the classifier, which aids in stabilizing the model during training. The guidance provided by a well-initialized classifier prevents the model from collapsing, thereby avoiding low performance. We suggest that the question of why initialization assists in achieving superior performance in FFT deserves more attention and research in future work to reveal more profound reasons.

Table 10. More results for LP and FFT on CIFAR-100 dataset. FFT† denotes using semantic-aware initialization.

SETTINGS	N4	N25	N100
FIXMATCH W/ LP	63.64	73.66	78.01
FINESSL W/ LP	68.22	75.57	78.59
FIXMATCH W/ FFT	6.56	32.93	49.76
FINESSL W/ FFT	8.07	41.64	59.88
FIXMATCH W/ LP-FT	70.80	80.25	85.16
FINESSL W/ LP-FT	77.45	83.18	86.04
FIXMATCH W/ FFT†	67.77	79.82	83.56
FINESSL W/ FFT†	78.03	84.04	86.33

B.3. ImageNet-21k Pre-Trained Model

Our experiments predominantly focus on the pre-trained ViT by initializing weights using the image encoder of CLIP. Here we delve into the performance of the model pre-trained with weights derived from ImageNet-21k (Deng et al., 2009). From Table 11, we can see that the fine-tuning of the model pre-trained on ImageNet-21k yields superior performance compared to the CLIP pre-trained model. We assert that this phenomenon can be attributed to the substantial correlation and similarity in image features between the ImageNet-21k and CIFAR-100 as well as Semi-Aves. This further clarifies why we predominantly employ the CLIP as the pre-trained model in most experiments, given its broader semantic feature space and substantially reduced feature similarity between pre-training and downstream training data. In addition, the ViT from Clip is trained using contrastive loss, which might bring a gap for direct downstream fine-tuning for image classification.

Examining the specific numerical results in Table 11, FINESSL continues to show its robust performance superiority, surpassing DebiasPL by an average of 0.36%.

Table 11. Results for fine-tuning the foundation model pre-trained on ImageNet-21k dataset.

	CIFAR-100			SEMI-AVES	
SETTINGS	N4	N25	N100	$\mathcal{D}_{in}^u$	$\mathcal{D}_{in}^u \cup \mathcal{D}_{out}^u$
FIXMATCH	87.05	90.69	91.43	67.89	63.63
FLEXMATCH	88.99	90.95	**91.88**	66.63	61.25
FREEMATCH	82.82	90.76	91.70	66.86	61.36
SOFTMATCH	82.93	90.99	91.59	66.55	61.34
DEBIASPL	89.96	91.00	91.57	69.40	63.93
OURS	**90.30**	**91.24**	91.87	**70.01**	**64.25**

B.4. Performance comparisons for ViT training from scratch

Previous work (Weng et al., 2022) has found that training ViTs from scratch fails to yield satisfactory performance on SSL tasks, particularly with limited labeled data. Furthermore, the significantly longer training time required for training ViTs from scratch is often impractical in real-world scenarios.

To validate this, we conduct experiments comparing the performance of training a ViT-small from scratch using FixMatch for 500 epochs, which is significantly longer than the 30 epochs used for VPT. Actually, ViT-base tends to show inferior performance due to more parameters (Wang et al., 2022b), which provides reason for our adoption of ViT-small. The performance of training from scratch is observed to be considerably poorer, with a 54.77% decrease compared to fine-tuning by VPT.

Additionally, we include a comparison between our proposed method, FINESSL and Semiformer (Weng et al., 2022), a method specifically designed for SSL tasks with ViTs. We focus on comparing results for the ImageNet dataset with 10% labeled data. The results demonstrate a 3.71% performance advantage of FINESSL over Semiformer in this setting, indicating significant advantages over training from scratch.

B.5. Additional comparisons with previous research

We incorporate thorough comparison with the strategies Few-PseudoLabels (FPL), Iterative Refinement of FPL (IFPL), and Grow and Refine Iteratively Pseudolabels (GRIP) as proposed in (Menghini et al., 2023). To ensure fair comparisons, we compare these strategies in both single-modality settings using VPT and multiple-modality settings using CoOp (Zhou et al., 2022) and VPT.

Our results show that FINESSL exhibits significant performance advantages, particularly in scenarios with limited labeled data, such as N4 and N25. It is worth noting that the use of a large volume of unlabeled data by FPL and IFPL is constrained due to the selection of a fixed number of unlabeled samples for each class. Moreover, the selected unlabeled data primarily consists of easy samples that are relatively simple to classify, thereby contributing only marginally to performance improvement. Furthermore, aggregated biases can negatively impact GRIP, resulting in substantial imbalance in pseudo-labels.

C. Further Analysis on FINESSL

C.1. Sensitivity Studies for Hyperparameters

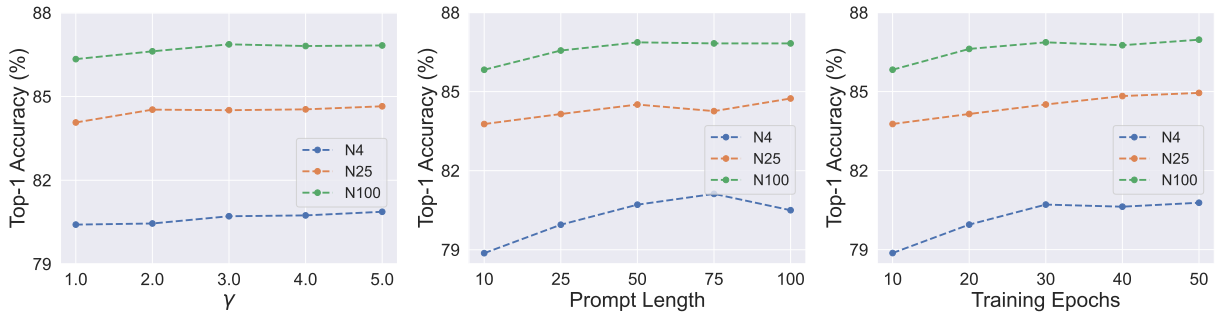
First, we investigate the impact of γ, which is a hyperparameter employed to scale the sample weights when calculating the consistency regularizer in Equation (7). From Figure 5a, we can see that changing the value of γ does not lead to substantial fluctuations in the generalization performance, indicating the robustness of our method.

Next, we study the influence of prompt length for VPT in Figure 5b. We observe a marginal performance enhancement as the VPT length increases, which means the performance can be further improved by adding more learnable model parameters. In our implementation, we set the VPT length to 50, which balances the effectiveness and efficiency.

Table 12. Results for FPL, IFPL, and GRIP in three settings (N4 / N25 / N100) on CIFAR-100 dataset.

DATASET	CIFAR-100		
SETTINGS	N4	N25	N100
FPL w/ VPT	74.80	80.23	84.24
IFPL w/ VPT	74.71	80.93	84.28
GRIP w/ VPT	70.24	80.83	84.32
FPL w/ VPT + CoOp	74.94	80.72	84.49
IFPL w/ VPT + CoOp	74.72	81.10	84.69
GRIP w/ VPT + CoOp	68.65	79.94	84.17
FineSSL	**80.44**	**84.51**	**86.66**

Finally, Figure 5c depicts the performance curve as a function of the number of training epochs. Generally, the test accuracy continues to rise as the training progresses. However, the performance stabilizes after training for 30 epochs. Therefore, we choose to fine-tune the model for 30 epochs in our experiments, which is adequate to obtain sufficiently well performance.

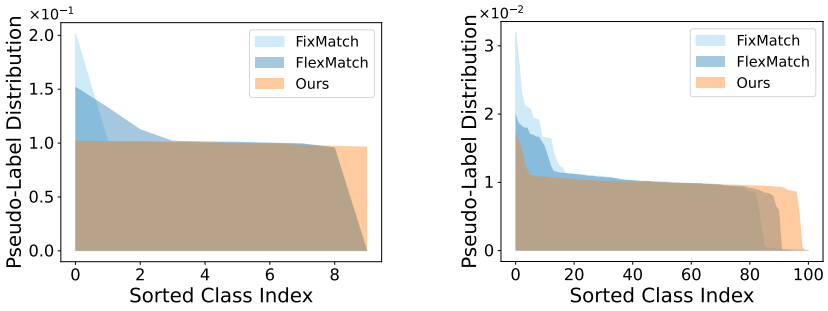


(a) (b) (c)

Figure 5. (5a to 5c): The sensitivity for γ, prompt length and training epochs under various settings on CIFAR-100.

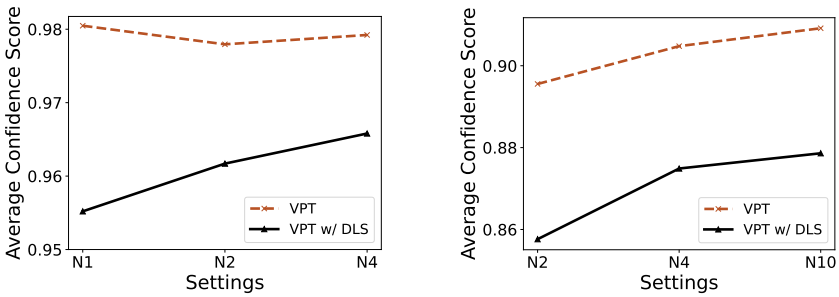
C.2. Alleviation of Aggregated Biases and Cognitive Deviation

We present additional figures to depict the alleviating effects on aggregated biases and cognitive deviation in Figure 6a and Figure 6b. Again, we observe that FineSSL generates a more balanced distribution of pseudo-labels compared to FixMatch and FlexMatch, which is a key factor enabling our method to alleviate aggregated biases. Additionally, as depicted in Figures 7a and 7b, the incorporation of DLS enables the model to more accurately assess the learning difficulty of various SSL tasks, thereby alleviating cognitive deviation. In contrast, the combination of VPT with FixMatch may lead to significant cognitive deviation, as shown in Figure 7a, leading to notable performance deterioration.



(a) CIFAR-10 (b) FOOD-101

Figure 6. Distribution of pseudo-labels for FixMatch, FlexMatch, and FineSSL in the "N2" setting.

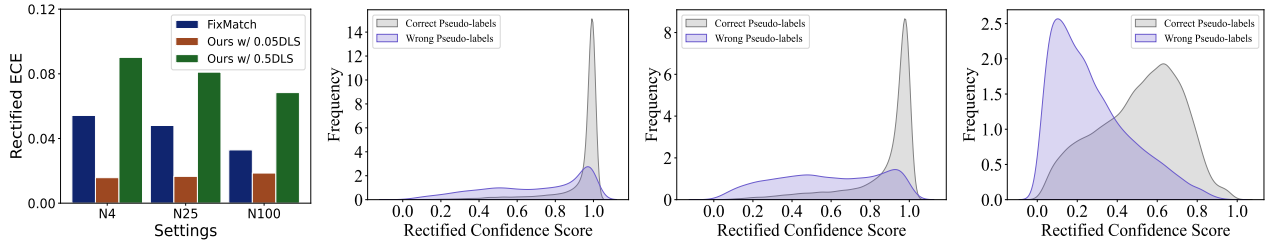


(a) CIFAR-10 (b) FOOD-101

Figure 7. Average confidence in tasks of different numbers of labeled samples for models with and without DLS.

C.3. Discussion on Cognitive Deviation and Calibration

Model calibration (Guo et al., 2017) is to modify the predictive probability which can reflect true correctness likelihood. However, cognitive deviation means the model does not exhibit corresponding confidence across settings of various learning difficulties. While there are indeed certain similarities between these two concepts, we show that mitigating cognitive deviation is better than improving calibration to boost performance. First, we rectify the model output confidence to align with the accuracy of pseudo-labels. We denote $\kappa = \mathrm{ACC}/\frac{1}{\mu B}\sum_{j=1}^{\mu B}\max(\boldsymbol{q}_j)$, where "ACC" denotes the accuracy of pseudo-labels, then rectify confidences by $\kappa \cdot \max(\boldsymbol{q}_j)$. Next, we show that the DLS proposed to overcome cognitive deviation can generate more reasonable sample weights for unlabeled data than models with better calibration. Figure 8a shows the expected calibration error (ECE) of different models, *i.e.,* FixMatch, FineSSL with $\lambda = 0.05$ and $\lambda = 0.5$ set in DLS. It can be seen that applying weak DLS ($\lambda = 0.05$) improves calibration, but it hurts when applying strong DLS ($\lambda = 0.5$). However, in Figures 8b to 8d, we observe that FixMatch exhibits significant overconfidence even in wrong predictions. Although applying weak DLS in FineSSL can partially alleviate this issue, the model assigns high weights for many unlabeled samples with wrong pseudo-labels. In contrast, strong DLS distinguishes correct and wrong pseudo-labels with a much clearer confidence boundary. To sum up, we show that DLS does not improve the calibration, but helps correlate the model confidence with learning difficulties and generate conservative certainty for pseudo-labels.



(a) (b) (c) (d)

Figure 8. (8a): ECE for FixMatch and FineSSL on CIFAR-100. "0.05DLS" denotes $\lambda = 0.05$ in label smoothing, and "0.5DLS" denotes $\lambda = 0.5$. (8b to 8d): Distribution of rectified confidences of FixMatch, FineSSL with $\lambda = 0.05$ and $\lambda = 0.5$, respectively.

C.4. Why VPT is better than full fine-tuning?

We will provide reasons from two perspectives. Firstly, a similar issue has been discussed in (Han et al., 2024), which suggests that full fine-tuning can outperform VPT only when the pre-trained task and downstream task share similar objectives but have dissimilar data feature distributions. It is evident that SSL tasks fall outside the scope of this scenario, indicating that VPT is expected to perform better. Secondly, when considering full fine-tuning, the target tasks often require a substantial amount of data for effective adaptation (Dosovitskiy et al., 2020; Chen et al., 2022a). However, SSL tasks typically operate under the assumption of extremely limited labeled data availability. Therefore, VPT is considered more suitable for SSL tasks compared to full fine-tuning.

D. Limitations of the work

We conclude our limitations concisely below:

- Our work primarily focuses on the pre-trained CLIP model. However, it is important to note that there are other self-supervised pre-training methods available, such as SimCLR (Chen et al., 2020a) and MOCO (He et al., 2020). In future work, we plan to investigate the impact of these alternative pre-training methods on performance.
- Our current approach only utilizes the image encoder in CLIP, neglecting the use of the text encoder. Considering the potential benefits of incorporating the text encoder, we believe it is worth exploring how its inclusion can augment the model's performance.