Imbalanced Semi-Supervised Learning via Label Refinement and Threshold Adjustment

Anonymous authors
Paper under double-blind review

Abstract

Semi-supervised learning (SSL) algorithms often struggle to perform well when trained on imbalanced data. In such scenarios, the generated pseudo-labels tend to exhibit a bias toward the majority class, and models relying on these pseudo-labels can further amplify this bias. Existing imbalanced SSL algorithms explore pseudo-labeling strategies based on either pseudo-label refinement (PLR) or threshold adjustment (THA), aiming to mitigate the bias through heuristic-driven designs. However, through a careful statistical analysis, we find that existing strategies are suboptimal: most PLR algorithms are either overly empirical or rely on the unrealistic assumption that models remain well-calibrated throughout training, while most THA algorithms depend on flawed metrics for pseudo-label selection. To address these shortcomings, we first derive the theoretically optimal form of pseudo-labels under class imbalance. This foundation leads to our key contribution: SEmi-supervised learning with pseudo-label optimization based on VALidation data (SEVAL), a unified framework that learns both PLR and THA parameters from a class-balanced subset of training data. By jointly optimizing these components, SEVAL adapts to specific task requirements while ensuring per-class pseudo-label reliability. Our experiments demonstrate that SEVAL outperforms state-of-the-art SSL methods, producing more accurate and effective pseudo-labels across various imbalanced SSL scenarios while remaining compatible with diverse SSL algorithms.

1 Introduction

Semi-supervised learning (SSL) algorithms are trained on datasets that contain labeled and unlabeled samples (Chapelle et al., 2009). SSL improves representation learning and refines decision boundaries without relying on large volumes of labelled data, which are labor-intensive to collect. Many SSL algorithms have been introduced, with one of the most prevalent assumptions being consistency (Zhou et al., 2003), which requires the decision boundaries to lie in low density areas. As a means of accomplishing this, pseudo-labels are introduced in the context of SSL (Scudder, 1965), and this concept has been extended to several variants employing diverse pseudo-label generation strategies (Laine & Aila, 2016; Berthelot et al., 2019b;a; Sohn et al., 2020; Wang et al., 2022c). In the pseudo-labelling framework, models trained with labelled data periodically classify the unlabelled samples, and samples that are confidently classified are incorporated into the training set.

The performance of pseudo-label-based SSL algorithms depends on the quality of the pseudo-labels (Chen et al., 2023). In real-world applications, the performance of these SSL algorithms often degrades due to the prevalence of class imbalance in real-world datasets (Liu et al., 2019). When trained with imbalanced training data $\mathcal{X}$, the model f will be biased in inference and tends to predict the majority class (Cao et al., 2019; Li et al., 2020). Consequently, this heightened sensitivity negatively impacts the pseudo-labels produced by SSL algorithms, leading to an ever increasing bias in models trained with these labels.

Therefore, developing improved methods for obtaining high-quality pseudo-labels remains a crucial research direction in imbalanced SSL. Current approaches primarily focus on two strategies: Pseudo-label refinement (PLR) (Lai et al., 2022a; Kim et al., 2020) and threshold adjustment (THA) (Zhang et al., 2021; Wang et al., 2022d). PLR modifies the decision boundary of pseudo-label logits to reduce majority-class bias, while THA optimizes confidence thresholds to maintain an better balance between global diversity and per-class accuracy in the selected pseudo-labels. While numerous SSL algorithms have demonstrated improved performance through enhanced pseudo-label quality, the field lacks a theoretical comparison of these predominantly heuristic-driven approaches. This gap makes it difficult to determine which design best suits specific applications. In this work, we provide new theoretical insights into pseudo-label generation from a statistical perspective, deriving principled strategies for PLR and THA in class-imbalanced scenarios. Surprisingly, our analysis reveals that while existing heuristic solutions partially mitigate bias, they remain fundamentally suboptimal—either lacking theoretical grounding or being compromised by improperly designed metrics for threshold selection.

Our key insight (detailed in Section 3) is that both PLR and THA depend on the distribution of the underlying test data rather than that of the unlabeled training data. Hence, we propose using a small fraction of distinct labeled datasets, as a proxy for unseen test data, to improve the quality of pseudo-labels [1]. Our method is named SEVAL, which is short for SEmi-supervised learning with pseudo-label optimization based on VALidation data. At its core, SEVAL refines the decision boundaries of pseudo-labels using a partition of the training dataset before proceeding with the standard training process. Similarly to AutoML (Zoph & Le, 2016; Ho et al., 2019), SEVAL can adapt to specific tasks by learning from the imbalanced data itself, resulting in a better fit. Moreover, SEVAL learns thresholds that can effectively prioritize the selection of samples from the high-**Precision** class, which we find to be critical but typically overlooked by current model confidence-based dynamic threshold solutions which only focus on **Recall** (Zhang et al., 2021; Guo & Li, 2022).

The contributions of this paper are as follows.

- In the context of pseudo-labeling for imbalanced SSL, we derive the theoretically optimal offsets for PLR and the optimal strategies for THA in Section 3. We also demonstrate the failure cases of existing popular pseudo-labeling strategies.
- We propose learning statistically grounded pseudo-label adjustment offsets in Section 4.1. The derived offsets do not rely on a calibrated model and improve accuracy in both pseudo-labeling and inference.
- We propose learning theoretically grounded thresholds for selecting correctly classified pseudo-labels, using a novel optimization function in Section 4.2. Our strategies outperform existing methods by relaxing the trade-off assumption of **Precision** and **Recall**.
- We combine these two techniques into a unified curriculum learning framework based on a training data partition, SEVAL in Section 4.3, and find that it outperforms state-of-the-art pseudo-label based SSL methods such as DARP, Adsh, FlexMatch and DASO under various imbalanced scenarios (c.f. Section 5).

2 Related Work

2.1 Semi-Supervised learning

Semi-supervised learning has been a longstanding research focus. The majority of approaches have been developed under the assumption of consistency, wherein samples with similar features are expected to exhibit proximity in the label space (Chapelle et al., 2009; Zhou et al., 2003). Compared with graph-based methods (Iscen et al., 2019; Kamnitsas et al., 2018), perturbation-based methods (Xie et al., 2020; Miyato

[1]Note that in practice, we learn the parameters with a partition of the labeled training dataset before the standard SSL process, thus not requiring any additional data.

et al., 2018) and generative model-based methods (Li et al., 2017; Gong et al., 2023), using pseudo-labels is a more straightforward and empirically stronger solution for deep neural networks (Van Engelen & Hoos, 2020). It periodically learns from the model itself to encourage entropy minimization (Grandvalet & Bengio, 2004). This process helps to position decision boundaries in low-density areas, resulting in more consistent labeling.

Deep neural networks are particularly suited for pseudo-label-based approaches due to their strong classification accuracy, enabling them to generate high-quality pseudo-labels (Lee et al., 2013; Van Engelen & Hoos, 2020). Several methods have been explored to generate pseudo-labels with a high level of accuracy (Wang et al., 2022c; Xu et al., 2021). For example, Mean-Teacher (Tarvainen & Valpola, 2017) calculates pseudo-labels using the output of an exponential moving average model along the training iterations; MixMatch (Berthelot et al., 2019b) derives pseudo-labels by averaging the model predictions across various transformed versions of the same sample; FixMatch (Sohn et al., 2020) estimates pseudo-labels of a strongly augmented sample with the model confidence on its weakly augmented version. Many of these approaches falter when faced with class imbalance in the training data, a frequent occurrence in real-world datasets.

2.2 Imbalanced Semi-Supervised Learning

Training on imbalanced data biases the model toward majority classes, degrading performance on minority classes (Cao et al., 2019). This bias compounds when using network-generated pseudo-labels: as more samples are pseudo-labeled, the model becomes increasingly skewed toward the majority class. There are three main categories of methods that address this challenge in the literature.

2.2.1 Long-tailed learning-based methods

The first group of methods alters the cost function computed using the labeled samples to train a balanced classifier, consequently leading to improved pseudo-labels. Long-tailed learning presents a complex problem in machine learning, wherein models are trained on data with a distribution characterized by a long tail. In such distributions, classes are imbalanced, with the tail classes consistently being underrepresented.

The research on long-tailed recognition (Chawla et al., 2002; Kang et al., 2019; Menon et al., 2020; Tian et al., 2020), which focuses on building balanced classifiers through adjusted cost functions or model structures in a fully supervised learning setting, often serves as inspiration for works in imbalanced SSL. For example, BiS (He et al., 2021) and SimiS (Chen et al., 2022) resample the labelled and pseudo-labelled training datasets to build balanced classifier. ABC (Lee et al., 2021) and Cossl (Fan et al., 2022) decouple the feature learning and classifier learning with a two head model architecture, where a unbiased classifier is learned through weighted cost function with features learned with the original cost function. The key is that they find feature learned with the unnormalized cost function can produce more robust features than that learned from normalized cost functions. Similarly, L2AC (Wang et al., 2022a) further decouples the feature and classifier learning by building an explicit bias attractor via bi-level optimization. SAW reweights unlabelled samples from different classes based on the learning difficulties (Lai et al., 2022b).

While these strategies can be effectively applied to SSL when assuming correct label generation, their performance remains fundamentally constrained by pseudo-label quality. In contrast, PLR and THA enhance class-wise accuracy by directly modifying pseudo-labels for unlabeled data through distinct mechanisms: PLR reduces class imbalance bias by redirecting selected examples toward adjusted predicted probability distributions; THA balances inter-class trade-offs through selective inclusion criteria for unlabeled examples.

2.2.2 Pseudo-label refinement-based methods

PLR-based methods modify logits of the produced pseudo-labels to alleviate the class bias caused by an imbalanced classifier. DARP (Kim et al., 2020) refines pseudo-labels by aligning their distribution with the target distribution. SaR (Lai et al., 2022a) aligns pseudo-labels to true distributions using distribution alignment-based mitigation vector. On the other hand, logit adjustment (LA) has been explored in the

context of SSL to address class bias by shifting the logit using the logarithm of the class priors' ratio, based on the class frequency from the training dataset (Wei & Gan, 2023).

2.2.3 Threshold adjustment-based methods

THA-based methods resamples unlabeled data to achieve class balance, preferentially incorporating more samples predicted as minority classes. Adsh (Guo & Li, 2022) utilizes an adaptive threshold to ensure that similar portions of pseudo-labels are selected for each class. Built upon FixMatch, FlexMatch and FreeMatch (Zhang et al., 2021; Wang et al., 2022d) choose confidently classified samples based on the model's learning progress, which results in the selection of more samples if the model is not learning well. InPL (Yu et al., 2023) proposes to rely on an energy score instead of maximum class probability to select confident samples. InstanT (Li et al., 2024) derives instance-wise thresholds based on the sample features. UPS (Rizve et al., 2021) utilizes both the confidence and uncertainty of a network prediction to select accurate subsets of pseudo-labels.

Of note, many SSL methods employ hybrid strategies that simultaneously adjust cost functions and refine pseudo-labels. For instance, in addition to a bootstrap sampling strategy, CReST+ (Wei et al., 2021) utilizes distribution alignment to adjust the class bias of pseudo-labels. DASO (Oh et al., 2022) improves pseudo-labels with semantic pseudo-labels and regularizes the feature encoder by aligning balanced semantic prototypes. ACR (Wei & Gan, 2023) is a holistic approach that builds upon the successes of ABC, FixMatch and MixMatch, and utilizes LA to refine pseudo-labels (Menon et al., 2020), yielding impressive results.

Although numerous methods have been proposed in this context, there remains a paucity of theoretical analysis regarding how to genuinely improve pseudo-labels. This theoretical gap makes it challenging for practitioners to understand and compare imbalanced SSL algorithms. In the following section, we shed new light on the problem of imbalanced SSL by providing statistical insights, decoupling SSL algorithm design, and deriving the theoretical foundations underlying various approaches. Building upon this formulation, we present SEVAL—a solution that integrates seamlessly into existing SSL pipelines without requiring modifications to learning strategies, data sampling procedures, or additional pseudo-label computations. Importantly, unlike many imbalanced SSL algorithms (e.g., DARP and CReST+), SEVAL makes no strong assumptions about the label distribution of unlabeled data, making it directly applicable to scenarios where labeled and unlabeled data follow different distributions.

3 Limitations of Current Methods

In this section, we begin by summarizing the framework of current pseudo-label based SSL. Next, we break down the design of current imbalanced SSL into two key components: PLR and THA. We then offer insights into these components through theoretical analysis, highlighting previously overlooked aspects.

3.1 Pseudo-label-based SSL

We consider the problem of C-class semi-supervised classification. Let X be the input space and $Y = \{1, 2, \ldots, C\}$ be the label space. We are given a set of N labelled samples $\mathcal{X} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ and a set of M unlabelled samples $\{\boldsymbol{u}_i\}_{i=1}^M$ in order to learn an optimal function or model f that maps the input feature space to the label space $f : X \to \mathbb{R}_+^C$. f can be viewed as an estimate of the conditional probability distribution $P(Y|X)$. In deep learning, f can be implemented as a network followed by a softmax function σ in order to produce the probability score p_c for each class c: $p_c = \sigma(\boldsymbol{z})_c = \frac{\mathrm{e}^{z_c}}{\sum_{j=1}^C \mathrm{e}^{z_j}}$ where $\boldsymbol{z}$ is the raw output (in C-dim) produced by the network. The predicted class is assigned to the one with the highest probability.

The parameters of f can be optimized by minimizing an empirical risk computed from the labeled dataset using:

$$R_{\mathcal{L},\mathcal{X}}(f) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(y_i, \boldsymbol{p}_i^{\mathcal{X}}), \tag{1}$$

where $\boldsymbol{p}_i^{\mathcal{X}}$ is the model predicted probabilities on the labelled data and $\mathcal{L}$ can be implemented as the most commonly used cross-entropy loss. To further utilize unlabelled data, a common approach is to apply pseudo-labeling to unlabelled data where the estimated label is generated from the network's predicted probability vector $\boldsymbol{q} \in \mathbb{R}^C$ (Lee et al., 2013). As there is no ground truth, we use the current model to produce a pseudo-label probability vector $\boldsymbol{q}_i \in \mathbb{R}^C$ for an unlabelled sample $\boldsymbol{u}_i$ and the pseudo-label $\hat{y}_i$ is determined as $\arg\max_j(q_{ij})$. In this way, we obtain $\hat{\mathcal{U}} = \{(\boldsymbol{u}_i, \hat{y}_i)\}_{i=1}^{M}$, where each $(\boldsymbol{u}_i, \hat{y}_i) \in (X \times Y)$. As the predicted label quality can be very poor especially at the early stage and for some challenging data, it is common to apply a threshold to identify reliable labels for model optimization [2]. The risk function on the unlabelled data can be defined as:

$$\hat{R}_{\mathcal{L},\hat{\mathcal{U}}}(f) = \frac{1}{M}\sum_{i=1}^{M}\mathbb{1}(\max_j(q_{ij}) \geq \tau)\mathcal{L}(\hat{y}_i, \boldsymbol{p}_i^{\mathcal{U}}), \tag{2}$$

where $\mathbb{1}$ is the indicator function and τ is a predefined threshold that filters out pseudo-labels with low confidence, and $\boldsymbol{p}_i^{\mathcal{U}}$ is the predicted probability for the unlabelled data. Unless otherwise specified, we do not have specific requirements for the network architecture in our setting. The neural network is trained using an equally weighted combination of the risk functions for labeled data $R_{\mathcal{L},\mathcal{X}}(f)$ and $\hat{R}_{\mathcal{L},\hat{\mathcal{U}}}(f)$.

In a class imbalanced setting, we have a challenge: the distribution of samples across the C classes is highly uneven with varying numbers of samples per class $\mathbf{n} : n_1, ..., n_c$. Some classes contain abundant samples (majority classes), while others have very few (minority classes). The class imbalance ratio is defined as $\gamma = \frac{max(\mathbf{n})}{min(\mathbf{n})}$, which in typical imbalanced settings exceeds 10 (reaching 300 in naturally collected datasets (Su & Maji, 2021)). In such case, the pseudo-labels on the unlabelled data can be biased to the majority class, which further amplifies the class imbalance problem. The model tends to predict majority classes over true classes during testing.

In this paper, to alleviate the issue of class imbalance-induced bias, we propose a method which can refine the probability vector used for pseudo-labelling $\boldsymbol{q}$ (c.f. Section 4.1) [3]. We also propose to adjust the threshold to operate on a class-specific basis, i.e. we use a vector $\boldsymbol{\tau} \in \mathbb{R}^C$ of threshold values to achieve accuracy fairness (c.f. Section 4.2). The model can then dynamically select the appropriate thresholds based on its prediction. The primary contribution of this paper is the statistical analysis of optimal strategies for adjusting $\boldsymbol{q}$, as well as the theoretical optimal choice of $\boldsymbol{\tau}$. In the following section, we will bypass the computation of pseudo-label probability $\boldsymbol{q}_i$ and concentrate on our contributions.

3.2 Pseudo-Label Refinement (PLR)

For an unlabelled sample $\boldsymbol{u}_i$, we determine its pseudo-label probability $\boldsymbol{q}_i$ based on its corresponding pseudo-label logit $\hat{\boldsymbol{z}}_i^{\mathcal{U}}$. In the process of PLR, we aim to adjust the decision boundaries for $\hat{\boldsymbol{z}}_i^{\mathcal{U}}$ with offset $\boldsymbol{\pi} \in \mathbb{R}^C$ to reduce class biases. Many methods in the literature have been discussed to utilize different $\boldsymbol{\pi}$ from different perspectives, such as distribution alignment (DA) and logit adjustment (LA). However, we find that none of them provides accurate refinement for imbalanced SSL. Here, we aim to shed new light on this problem by analyzing optimal THA strategies from a statistical perspective (Saerens et al., 2002). We assume that the test distribution $\mathcal{T}$ shares identical class conditionals with the training dataset $\mathcal{X}$ (i.e., $P^{\mathcal{X}}(X|Y) = P^{\mathcal{T}}(X|Y)$) and deviates solely in terms of class priors ($P^{\mathcal{X}}(Y) \neq P^{\mathcal{T}}(Y)$), we can assert:

[2] We describe the case of hard pseudo-labels for simplicity, but the method generalizes to the case of soft pseudo-labels.

[3] We use different notations $\boldsymbol{q}$ and $\boldsymbol{p}^{\mathcal{U}}$ to denote the probability obtained for pseudo-labelling and model optimization. This is common in existing SSL algorithms, where the two can be obtained in different ways for better model generalization (Laine & Aila, 2016; Sohn et al., 2020; Berthelot et al., 2019b;a). For example, in FixMatch (Sohn et al., 2020), $\boldsymbol{q}_i = f(\mathcal{O}_w(\boldsymbol{u}_i))$ is estimated on a weakly augmented sample of an input image for reliable supervision whereas $\boldsymbol{p}_i^{\mathcal{U}}$ is estimated using a strongly-augmented (i.e. RandAugment (Cubuk et al., 2020)) version $\mathcal{O}_s(\boldsymbol{u}_i)$ as model input for the same instance i.

Table 1: Theoretical comparisons of SEVAL and other pseudo-label refinement methods including distribution alignment (DA) (Berthelot et al., 2019a; Wei et al., 2021; Lai et al., 2022a; Kim et al., 2020), logit adjustment (LA) (Wei & Gan, 2023; Menon et al., 2020) and DASO (Oh et al., 2022). $\mathcal{X}$: Labelled training data; $\mathcal{U}$: Unlabelled training data; $\mathcal{T}$: Test data; $\mathcal{V}$: An independent labelled data.

	Optimal results (c.f. Eq. 1)	DA (Berthelot et al., 2019a)	LA (Menon et al., 2020)	DASO (Oh et al., 2022)	SEVAL (c.f. Section 4.1)
Estimation of the classifier on re-sampled $\mathcal{U}$, $f^{\mathcal{U}}(X)$	$\frac{f^*(X)P^{\mathcal{T}}(Y)}{P^{\mathcal{X}}(Y)}$	$\frac{f(X)P^{\mathcal{U}}(Y)}{\hat{P}^{\mathcal{U}}(Y)}$	$\frac{f(X)P^{\mathcal{T}}(Y)}{P^{\mathcal{X}}(Y)}$	Blending similarity based pseudo-label	$\frac{f(X)P^{\mathcal{T}}(Y)}{\boldsymbol{\pi}^*}$
Note	In the spirit of statistics.	$\hat{P}^{\mathcal{U}}(Y)$ is the model prediction of $P^{\mathcal{U}}(Y)$. Suboptimal as it ignores $P^{\mathcal{T}}(X,Y)$, especially with class imbalance.	Inaccurate as f is suboptimal and uncalibrated.	Relying on the effectiveness of blending strategies.	Optimizing the decision boundary on $\mathcal{U}$ using $\mathcal{V}$ as a proxy without assuming a specific f.

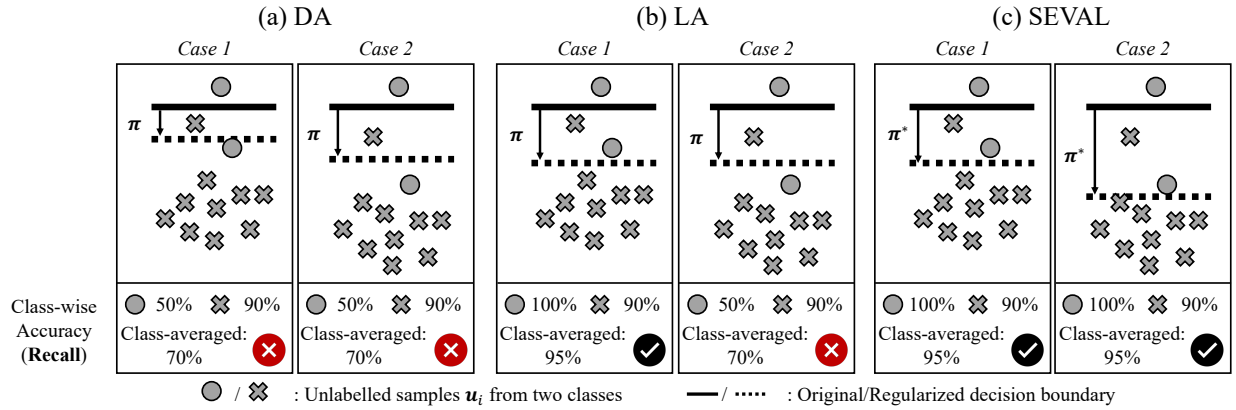

Figure 1: Illustration of limitations in PLR methods for a two-class imbalanced classification task. As in Remark 1, assuming $P^{\mathcal{T}}(Y)$ is uniform, the decision boundary should shift to maximize class-averaged likelihood. Here, we show class-wise accuracy at the bottom, which can be seen as an indictor of the class-averaged likelihood but is more straightforward to compare. (a) DA imposes constraints based on the distribution of $\mathcal{U}$. Even if $\mathcal{U}$ is estimable, the regularized boundary tends to be unfair to minority classes due to class imbalance, as false negatives can have a greater impact on the minority class. (b) LA refines the boundary using training data distributions, but fixed offsets (e.g., $\boldsymbol{\pi}$) is likely to perform poorly as logit distributions shift during training. (c) The proposed scheme directly optimizes the class-averaged likelihood (c.f. Section 4.1) and fits the training process with curriculum learning (c.f. Section 4.3).

Proposition 1 *Given that a classifier $f^*(X)$ is optimized on $P^{\mathcal{X}}(X,Y)$,*

$$f^{\mathcal{T}}(X) \propto \frac{f^*(X)P^{\mathcal{T}}(Y)}{P^{\mathcal{X}}(Y)}, \tag{3}$$

is the optimal Bayes classifier *on $P^{\mathcal{T}}(X,Y)$, where $P^{\mathcal{X}}(X|Y) = P^{\mathcal{T}}(X|Y)$ and $P^{\mathcal{X}}(Y) \neq P^{\mathcal{T}}(Y)$.*

If we have access to the class ratio between underlying test data and unlabeled training data, we can construct a reweighted dataset by adjusting (X,Y) to align the label distribution of the unlabeled training dataset $P^{\mathcal{U}}(Y)$ with that of the target dataset $P^{\mathcal{T}}(Y)$, while preserving the feature-label relationship $P^{\mathcal{U}}(X,Y)$. Using this resampled dataset, we can assert:

Corollary 1 *The Bayes classifier $f^{\mathcal{T}}(X)$ should be also optimal on the resampled unlabelled data $\frac{P^{\mathcal{U}}(X,Y)P^{\mathcal{T}}(Y)}{P^{\mathcal{U}}(Y)}$, where $P^{\mathcal{T}}(X|Y) = P^{\mathcal{U}}(X|Y)$ and $P^{\mathcal{T}}(Y) \neq P^{\mathcal{U}}(Y)$.*

We provide proofs in Appendix Section B.1. This analysis provides insight into the formulation of pseudo-label offsets: *it depends on the label distribution of the test data, $P^{\mathcal{T}}(Y)$, rather than the distribution of unlabeled data, $P^{\mathcal{U}}(Y)$.*

According to Corollary 1, in order to learn the optimal Bayes classifier on $P^{\mathcal{U}}(X,Y)$, we should resample or reweight the unlabelled data based on the class priors ratio of $P^{\mathcal{T}}(Y)/P^{\mathcal{U}}(Y)$. If $P^{\mathcal{T}}(Y)$ follows a uniform distribution [4], the classifier should be trained to achieve optimal performance on $P_{\mathcal{U}}(X,Y)$ across different classes as the term $P^{\mathcal{T}}(Y)/P^{\mathcal{U}}(Y)$ would normalize the optimization functions across classes. Specifically, we claim that:

Remark 1 *The optimal Bayes classifier on $P^{\mathcal{U}}(X,Y)$ should have maximized class-averaged likelihood, if the $P^{\mathcal{T}}(Y)$ is uniform.*

From this perspective, we summarize current PLR approaches in Table 1. We find that although existing approaches can reduce the class bias of pseudo label by moving the decision boundary based on different criteria, their discrepancy with the optimal results would lead to suboptimal results.

DA (Berthelot et al., 2019a; Wei et al., 2021; Kim et al., 2020) is a commonly employed technique to make balanced prediction for different classes which align the predicted class priors to true class priors of $\mathcal{U}$, making the model being fair (Bridle et al., 1991). Typically, it is only applicable to cases where the labeled and unlabeled datasets have the same label distributions, as it requires the assessment of the true class prior of the unlabeled data. However, this is not necessarily true in real-world applications. More importantly, it only reduces the calibration errors but cannot be optimally fair because it relies on an incorrect optimization function that does not take $P^{\mathcal{T}}$ into account. Fig 1(a) illustrates this in imbalanced SSL. The decision boundary obtained through DA would be suboptimal because enforcing proportional predictions (i.e., requiring two sample predictions for class ●) increases false negatives without improving true positives.

LA modifies the network prediction from $\arg\max_c(\hat{z}^{\mathcal{U}}_{ic})$ to $\arg\max_c(\hat{z}^{\mathcal{U}}_{ic} - \beta \log \pi_c)$, where β is a hyper-parameter and $\boldsymbol{\pi}$ is determined as the empirical class frequency (Menon et al., 2020; Zhou & Liu, 2005; Lazarow et al., 2023). It shares similar design with Eq. 1, however, recall that proposition 1 provides a justification for employing logit thresholding when optimal probabilities $f^*(X)$ are accessible. Although neural networks strive to mimic these probabilities, it is not realistic for LA as neural networks are often uncalibrated and over confident (Guo et al., 2017). We think that LA could have more profound errors when applied to SSL, as the classifier is not optimal and the model is likely to be uncalibrated during the training process (Loh et al., 2022). An obvious failing case is illustrated in Fig. 1 (b), where *Case 1* and *Case 2* represent the logit distributions of different training procedures. During the training process, the logits of samples are pushed away from the decision boundary (i.e. *Case 1* $\rightarrow$ *Case 2*), and the optimal regularized decision boundary should also shift. However, the decision boundary refinement in LA is only related to the class prior of the training and test datasets, and it lacks a mechanism to dynamically track the logit distribution.

The accurate estimation of classifier bias requires the calculation of a conditional confusion matrix, which always requires holdout data (Lipton et al., 2018). Therefore, in this study, we estimate the classifier bias using holdout data and explicitly determine the optimal decision boundary following Remark 1. We hypothesize that the optimal decision boundary should remain consistent across datasets with similar training sample sizes. Accordingly, we develop a curriculum learning approach that: (1) learns optimal parameters by splitting the training dataset, and (2) applies the derived patterns to the complete training dataset. This methodology enables optimal decision boundary derivation during training without requiring additional labeled data. The importance of a curriculum in SSL—where dynamic pseudo-labels are used at

[4]For simplicity, here and in the following section, we assume that the test data $P^{\mathcal{T}}(Y)$ is equally distributed across different classes, which is the most common assumption in existing methods. The analysis and methods can easily be generalized to other $P^{\mathcal{T}}(Y)$ assumptions.

different learning stages—is also highlighted in (Cascante-Bonilla et al., 2021). Further details are provided in Section 4.1.

3.3 Threshold Adjustment (THA)

Here, we look into the impact of pseudo-label quality on the SSL. For simplicity, we consider in this section a model f trained on $\hat{\mathcal{U}}$ for *binary classification* using supervised learning loss $\mathcal{L}$. $(\boldsymbol{u}, \hat{y})$ is an arbitrary sample drawn from $\hat{\mathcal{U}}$. In this section we refer $\mathcal{U} = \{(\boldsymbol{u}_i, y_i)\}_{i=1}^M$ to the oracle distribution which contains the inaccessible real label y for $\boldsymbol{u}$. Let ρ be the noise rate of selected pseudo-labels, defined as $\rho = 1 - P_{\hat{\mathcal{U}}}(y = \hat{y})$ with $\rho < 0.5$. The oracle expected risk is defined as $R_{\mathcal{L},\mathcal{U}}(f) := \mathbb{E}_{(\boldsymbol{u},y)\sim\mathcal{U}}[\mathcal{L}(f(\boldsymbol{u}), y)]$.

The theorem presented below indicates that, with the number of samples in set $\mathcal{U}$ is fixed as M, better model performance is achieved through training with a dataset exhibiting a lower noise rate ρ.

Theorem 1 *Given $\hat{f}$ is the model after optimizing with $\hat{\mathcal{U}}$. Assume the loss function $\mathcal{L}$ is L-Lipschitz in all predictions. For any confidence parameter $\delta > 0$, the following holds with probability $\geq 1 - \delta$:*

$$R_{\mathcal{L},\mathcal{U}}(\hat{f}) \leq \min_{f\in\mathcal{F}} R_{\mathcal{L},\mathcal{U}}(f) + 4L_p\mathfrak{R}(\mathcal{F}) + 2\sqrt{\frac{\log(1/\delta)}{2M}}, \tag{4}$$

where the Rademacher complexity $\mathfrak{R}(\mathcal{F})$ is defined by $\mathfrak{R}(\mathcal{F}) := \mathbb{E}_{\boldsymbol{x}_i,\epsilon_i}\left[\sup_{f\in\mathcal{F}} \frac{1}{M}\sum_{i=1}^M \epsilon_i f(\boldsymbol{u}_i)\right]$ for function class $\mathcal{F}$ and $\epsilon_1, \ldots, \epsilon_M$ are i.i.d. Rademacher variables. $L_p \leq \frac{2L}{1-2\rho}$ is the Lipschitz constant.

We provide proofs in Appendix Section B.3.

The insight of Theorem 1 is straightforward: given a fixed label size M, reducing the noise rate ρ tightens the bound toward its upper limit, yielding a better model. To achieve this, we want to increase the accuracy of the pseudo-label, of which the marginal distribution over $\hat{y}$ can be calculated as:

$$P_{\hat{\mathcal{U}}}(y = \hat{y}) = \sum_{j=1}^{C} \underbrace{P_{\hat{\mathcal{U}}}(y = j|\hat{y} = j)}_{\textbf{Precision}} \underbrace{P_{\hat{\mathcal{U}}}(\hat{y} = j)}_{\textit{Accessible}}. \tag{5}$$

This indicates that, if we can have an approximation of **Precision**, we can feasibly tune the pseudo-label accuracy by controlling the sampling strategies based on class-specific thresholds $\boldsymbol{\tau}$. However, it is not possible with

$$P_{\hat{\mathcal{U}}}(y = \hat{y}) = \sum_{j=1}^{C} \underbrace{P_{\hat{\mathcal{U}}}(\hat{y} = j|y = j)}_{\textbf{Recall}} \underbrace{P_{\hat{\mathcal{U}}}(y = j)}_{\textit{Inaccessible}}, \tag{6}$$

because in practice we do not have the information of ground truth label for $\boldsymbol{u}_i$. Formally, we claim that:

Remark 2 *A better thresholding vector $\boldsymbol{\tau}$ for selecting effective pseudo-labels should be derived from class-wise **Precision** rather than class-wise **Recall**.*

However, existing dynamic threshold approaches (Zhang et al., 2021; Wang et al., 2022d; Guo & Li, 2022) derive the threshold for class c based on estimated **Recall**. Specifically, they all rely on the maximum class probability of class c, i.e. P'_c [5]. Although sample-wise aggregation of maximum class probabilities estimates overall accuracy (Guo et al., 2017) of test samples (Garg et al., 2022; Li et al., 2022), it is important to note that when evaluated per-class, P'_c corresponds to **Recall** since negative samples are excluded. In other words, when dynamic techniques like FlexMatch prioritize selecting samples from classes with lower maximum class probabilities, they inherently increase sampling frequency for classes that demonstrate lower **Recall**!

[5]Formally, it is calculated as $P'_c = \frac{1}{K_c}\sum_{i=1}^K \mathbb{1}_{ic} \max_j p_{ij}^{\mathcal{U}}$, where $\mathbb{1}_{ic} = \mathbb{1}(\arg\max_j(p_{ij}^{\mathcal{V}}) = c)$ is 1 if the predicted most probable class of sample i is c and 0 otherwise and $K_c = \sum_{i=1}^K \mathbb{1}_{ic}$ counts samples predicted as c.

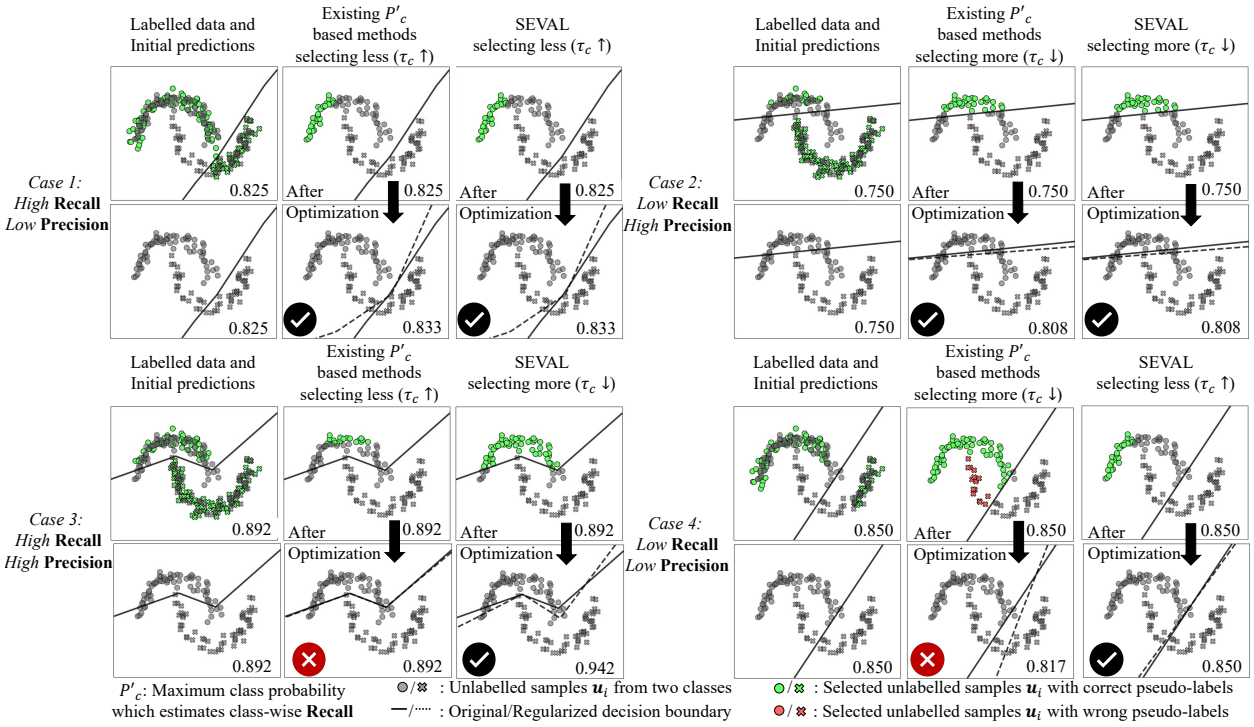

Figure 2: Two-moons toy experiments illustrating the relationship between threshold choice and model performance for class ●. Accuracy appears in the bottom right. Current maximum class probability-based dynamic thresholding methods such as FlexMatch (Zhang et al., 2021), emphasizing **Recall**, may not be reliable for *Case 3* and *Case 4*. In comparison, SEVAL derived thresholds, reflecting **Precision**, fit all cases well (c.f. Section 4.2).

We argue that their strategies are based on the assumption that **Recall** and **Precision** are always in a trade-off relationship, which arises from adjusting decision boundaries—for example, lower **Recall** often leads to higher **Precision**. However, they would fall short if this does not hold. For example, we should choose as much as possible if the class is well-classified, e.g. high **Recall** and high **Precision**. However, following their strategy, they will choose few from them. We illustrate this in Fig. 2 with the two-moons example. While *Case 1* and *Case 2* are the most common scenarios, current maximum class probability-based approaches struggle to estimate thresholds effectively in other cases. We substantiate this assertion in the experimental section, where we find that *Case 3* frequently arises for the minority class in imbalanced SSL and is currently not adequately addressed, as shown in Section 5.6 and Appendix Section E.

Therefore, in this study, we propose to learn the thresholds that can reflect **Precision** and maximum $P_{\hat{\mathcal{U}}}(y = \hat{y})$ as stated in Eq. 5. We find that **Precision** is difficult to estimate directly from the network output because the network's probability does not inherently account for false negatives, which is essential for **Precision** 5.6. Therefore, similar to the approach used for PLR, we utilize a holdout dataset to estimate **Precision** and determine the thresholds based on this estimation. We detail the formulation in Section 4.2.

4 SEVAL

Fig. 3 shows an overview of SEVAL. The comparative advantage of SEVAL over existing methodologies is demonstrated in Table 1, Fig. 1, and Fig. 2. Critically, we propose to optimize both PLR and THA parameters based on our previous analysis, employing properly justified criteria to ensure optimal pseudo-label utilization in class-imbalanced SSL.

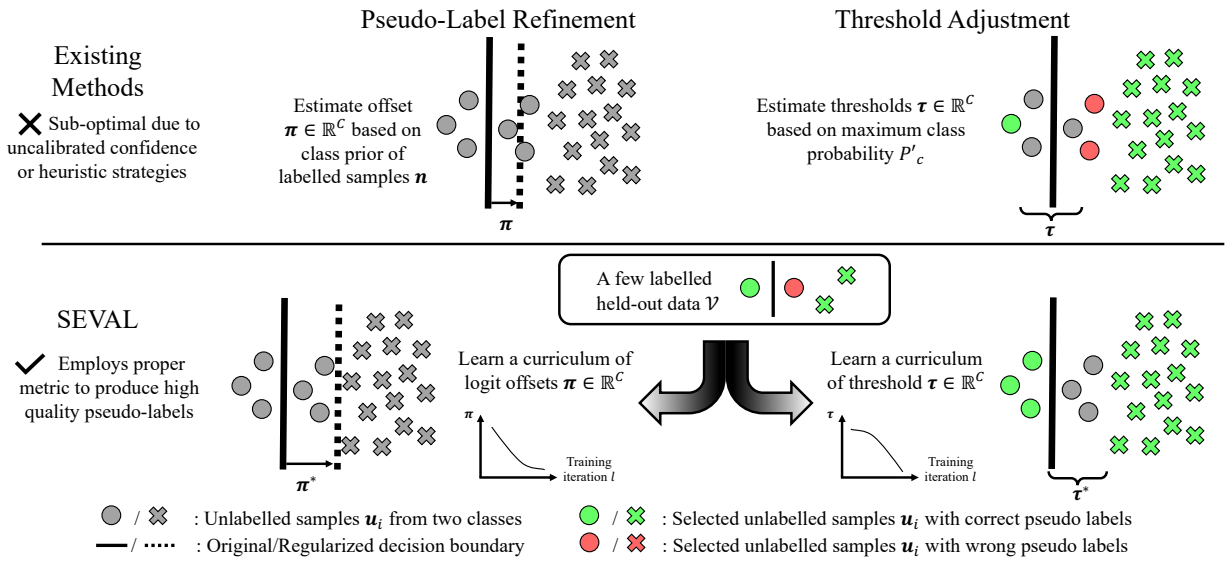

Figure 3: Overview of SEVAL optimization process which consists of two learning strategies aiming at mitigating bias in pseudo-labels within imbalanced SSL scenarios: PLR and THA. The parameter learning curriculum is determined by evaluating held-out data performance, which ensures greater accuracy while preventing overfitting. Note that PLR and THA are entirely complementary, as they address fundamentally different stages of the pseudo-labeling pipeline.

Independent of the training dataset $\mathcal{X}$ and $\mathcal{U}$, we assume we have access to a held-out dataset $\mathcal{V} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^K$, which contains k_c samples for class c. We use this data to learn $\boldsymbol{\pi}$ for PLR and $\boldsymbol{\tau}$ for THA. We make no assumptions regarding k_c; that is, $\mathcal{V}$ can either be balanced or imbalanced [6].

4.1 Learning Pseudo-Label Refinement

4.1.1 Learning Offsets

We want to further harness the potential of pseudo-label refinement by optimizing $\boldsymbol{\pi}$ from the data itself. We propose to directly estimate the optimal decision boundary as required in Proposition 1. As stated in Remark 1, under the assumption that $P^{\mathcal{T}}(Y)$ is uniform, we optimize the parameters by minimizing the cross-entropy loss averaged across classes. Specifically, the optimal offsets $\boldsymbol{\pi}$, are optimized using the labelled held-out data $\mathcal{V}$ with:

$$\begin{aligned} \boldsymbol{\pi}^* &= \arg\min_{\boldsymbol{\pi}} \sum_{j=1}^{C} \frac{1}{Ck_j} \sum_{i=1}^{K} \mathbb{1}(y_i = j) \mathcal{L}(y_i, \boldsymbol{p}_i^{\mathcal{V}}) \\ &= \arg\min_{\boldsymbol{\pi}} \sum_{j=1}^{C} \frac{1}{Ck_j} \sum_{i=1}^{K} \mathbb{1}(y_i = j) \mathcal{L}(y_i, \sigma(\boldsymbol{z}_i^{\mathcal{V}} - \log \boldsymbol{\pi})), \end{aligned} \tag{7}$$

where $\mathcal{L}$ is the cross-entropy loss and $\boldsymbol{z}_i^{\mathcal{V}}$ is the network output for an input sample. Subsequently, we can compute the refined pseudo-label logit as $\hat{\boldsymbol{z}}_i^{\mathcal{U}} - \log \boldsymbol{\pi}^*$, which are expected to become more accurate on a class-wise basis. Of note, because Eq. 7 uses class-averaged cross-entropy loss, it avoids needing assumptions about the class-conditional likelihood for held-out data $\mathcal{V}$.

[6] In practice, $\mathcal{V}$ is normally separated from $\mathcal{X}$, c.f. Section 4.3. In implementation, we do not require a sizable $\mathcal{V}$, as we find that classes with similar class ratios will have similar offsets and thresholds, and thus their parameters can be optimized together at the group level, c.f. Section D.2.

4.1.2 Extension to Test Logit Inference

As established in Proposition 1, the learned offsets $\boldsymbol{\pi}^*$ maintain optimality on test data with a uniform class distribution – a standard evaluation setting. Therefore, we expect $\boldsymbol{\pi}^*$ to not only improve pseudo-labeling on unlabelled data $\mathcal{U}$ but also outperform class frequency-based alternatives like LA (Zhou & Liu, 2005) during inference on the test set. This is because LA assumes the learned networks are calibrated to reflect the true probabilities of $P^{\mathcal{X}}(X|Y)$, which is rarely satisfied in practice.

Particularly, because the learned offsets are intrinsically linked to network parameters, we employ the final $\boldsymbol{\pi}^*$ learned on $\mathcal{V}$ to refine test results, anticipating superior performance compared to alternative methods like LA.

4.2 Learning Threshold Adjustment

4.2.1 Thresholding Criteria

For unlabeled samples, we select samples with largest predicted probabilities. A lower thresholds τ will lead to lower accuracy, but include more samples. As we demonstrated earlier in Remark 2, we should rely on **Precision** to determine class-specific thresholds since we can only access $\hat{y}$ but not y for unlabeled samples. Specifically, we claim that (with the justification to follow later in the text):

Remark 3 *When the number of selected unlabeled samples is fixed, the optimal threshold $\boldsymbol{\tau}$ to achieve the highest pseudo-label accuracy should ensure that every class in the selected samples has the same **Precision**.*

We begin with Eq. 5, which maximizes pseudo-label correctness in our running example:

$$\max_{\boldsymbol{\tau}} \sum_{c=1}^{C} \mathcal{A}(\tau_c, c)\mathcal{S}(\tau_c, c), \quad \text{s.t.} \quad \sum_{c=1}^{C} \mathcal{S}(\tau_c, c) = M, \tag{8}$$

where $\boldsymbol{\tau}$ is the thresholds for sample selection, with its c'th element corresponding to the threshold for class c. $\mathcal{A}(\tau_c, c)$ represents **Precision** of class c when selecting samples using τ_c. Specifically, $\mathcal{A}(\tau_c, c)$ computes the accuracy of samples predicted as class c whose maximum probability exceeds τ_c:

$$\mathcal{A}(\tau_c, c) = \frac{1}{\mathcal{S}(\tau_c, c)} \sum_{i=1}^{K} \mathbb{1}_{ic} \mathbb{1}(y_i = c) \mathbb{1}(\max_j(p_{ij}^{\mathcal{V}}) > \tau_c), \tag{9}$$

where $\mathbb{1}_{ic}$ indicates if the most probable class is c for samples i as we mentioned earlier and $\mathcal{S}(\tau_c, c) = \sum_{i=1}^{K} \mathbb{1}_{ic} \mathbb{1}(\max_j(p_{ij}^{\mathcal{V}}) > \tau_c)$ is the number of samples predicted as class c with confidence larger than τ_c.

Using Lagrange multipliers, we formulate the Lagrangian for this problem with associated multiplier λ as:

$$\Lambda(\tau_c, \lambda) = \sum_{c=1}^{C} \mathcal{A}(\tau_c, c)\mathcal{S}(\tau_c, c) - \lambda\Big(\mathcal{S}(\tau_c, c) - M\Big). \tag{10}$$

Taking the derivative of Λ with respect to τ_c and setting it to zero yields the optimality conditions:

$$\frac{\partial}{\partial \tau_c}\Big[\mathcal{A}(\tau_c, c)\mathcal{S}(\tau_c, c)\Big] = \lambda \frac{\partial \mathcal{S}(\tau_c, c)}{\partial \tau_c}, \quad \forall c. \tag{11}$$

Since our focus is solely on the trade-off in sample selection across classes, we can adjust the thresholds τ_c to ensure that the selected samples are updated at a similar rate. In other words, it is safe to assume that $\frac{\partial \mathcal{S}(\tau_c, c)}{\partial \tau_c} = \lambda_{\mathcal{S}}$ across all classes, where $\lambda_{\mathcal{S}}$ is a constant.

Thus after some rearrangement, we can obtain:

$$\mathcal{A}(\tau_c, c)\lambda_{\mathcal{S}} + \mathcal{S}(\tau_c, c)\frac{\partial \mathcal{A}(\tau_c, c)}{\partial \tau_c} = \lambda\lambda_{\mathcal{S}}, \quad \forall c. \tag{12}$$

We further assume that the changes in accuracy are smooth and tend to be flatten, meaning that adjusting threshold τ_c does not significantly affect the overall accuracy of the specific class. This assumption is reasonable when a sufficient number of unlabeled samples are available per class. Under this condition, the term $\frac{\partial \mathcal{A}(\tau_c,c)}{\partial \tau_c}$ approaches 0, and consequently, $\epsilon = \mathcal{S}(\tau_c, c)\frac{\partial \mathcal{A}(\tau_c,c)}{\partial \tau_c}$ becomes very small. As a result, Eq. 12 can be approximated as:

$$\mathcal{A}(\tau_c, c) = \frac{\lambda\lambda_{\mathcal{S}} - \epsilon}{\lambda_{\mathcal{S}}}, \quad \forall c. \tag{13}$$

This demonstrates that at the optimal solution, $\mathcal{A}(\tau_c, c)$ for each class should be equal. This condition enforces a relationship between the thresholds τ_c across classes, ensuring a unique solution that satisfies Eq. 10. Consequently, we conclude that the optimal solution can be achieved by aligning the **Precision** across all classes to a common level.

4.2.2 Connections to the Neyman-Pearson Lemma

The Neyman-Pearson Lemma (NPL) (Neyman & Pearson, 1933) provides the theoretical foundation for our THA optimization in Remark 3, where the likelihood ratio test optimally maximizes statistical power (i.e. true positive rate) at fixed significance levels (i.e. false positive rate). Our framework generalizes this principle to multi-class settings by maximizing a correctness metric (i.e. $\mathcal{A}$) under resource constraints (i.e. $\mathcal{S}$).

The NPL provides optimal threshold selection for binary decisions through constrained power maximization. Similarly, our method determines optimal multi-class thresholds by balancing two competing factors: per-class accuracy $\mathcal{A}(\tau_c, c)$ against a global prediction budget constraint $\sum_{c=1}^{C} \mathcal{S}(\tau_c, c) = M$. While the NPL relies on likelihood ratios that require known distributions, our approach operates purely through empirical accuracies, making it suitable for data-driven applications where likelihood functions are unavailable.

Through a simple example, here we demonstrate how threshold dynamics govern the optimality of balanced class accuracies. When all classes reach equal accuracy $\mathcal{A}^*$ at their optimal thresholds τ_c^*, any deviation from these thresholds disrupts equilibrium. Introducing samples below τ_c^* (with accuracy $\leq \mathcal{A}^*$) reduces the average accuracy, while excluding samples above τ_c^* (with accuracy $\geq \mathcal{A}^*$) unnecessarily discards correct predictions.

This leads to the key insight: the only stable solution that simultaneously maximizes total correctness and respects the prediction budget occurs when all classes share identical accuracy $\mathcal{A}^*$. At this point, no reallocation of samples between classes can further improve the objective. This equilibrium condition represents a natural generalization of the NPL's optimality criterion, adapted for multi-class prediction with empirical thresholds.

4.2.3 Learning Thresholds

Building on this analysis, we optimize the thresholds to guarantee equal class-wise accuracy - specifically, the **Precision** metric from Eq. 5 - at target level t for selected samples across all classes. As stated earlier, **Precision** cannot be determined solely by network logits $\boldsymbol{z}$, since it is also influenced by non-maximum probabilities (a point we further elucidate in Section 5.6). Thus, here we explore a method to learn optimal thresholds using an external held-out dataset $\mathcal{V}$, aiming to obtain pseudo-labels with maximized accuracy under given budgets. This is achieved by:

$$\tau_c^* = \begin{cases} \arg\min_{\tau_c} \left|\mathcal{A}(\tau_c, c) - t\right| & \text{if} \quad \alpha_c < t \\ 0 & \text{otherwise} \end{cases}, \tag{14}$$

Algorithm 1 SEVAL parameter estimation process, $\boldsymbol{\pi}^*$, $\boldsymbol{\tau}^* \leftarrow$ ESTIM $\left(\mathcal{V}, \{\boldsymbol{z}_i^{\mathcal{V}}\}_{i=1}^K, t\right)$

Require:

$\mathcal{V} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^K$: a held-out dataset, $\{\boldsymbol{z}_i^{\mathcal{V}}\}_{i=1}^K$: the network predicted logits for $\mathcal{V}$, t: Requested per class accuracy of the pseudo-label.

C: Number of classes, $\boldsymbol{k}$: Number of sample per class in $\mathcal{V}$.

$\boldsymbol{\pi}^* = \arg\min_{\boldsymbol{\pi}} \frac{1}{K} \sum_{i=1}^K \mathcal{L}(y_i, \sigma(\boldsymbol{z}_i^{\mathcal{V}} - \log \boldsymbol{\pi}))$

▷ *In practice, the parameter estimation process is achieved by bound-constrained solvers.*

$\boldsymbol{\omega}^{\mathcal{V}} = 1/\boldsymbol{k}$ ▷ *The minority class is assigned higher weights to prioritize class-specific accuracy.*

for c in C **do**

  Get class-wise accuracy $\alpha_c = \frac{1}{K_c} \sum_{i=1}^K \omega_{y_i}^{\mathcal{V}} \mathbb{1}_{ic} \mathbb{1}(y_i = c)$

  ▷ *For each class c, $\mathbb{1}_{ic} = \mathbb{1}(\arg\max_j(p_{ij}^{\mathcal{V}}) = c)$ and $K_c = \sum_{i=1}^K \omega_{y_i}^{\mathcal{V}} \mathbb{1}_{ic}$ is the normalized number of samples predicted as c.*

  if $\alpha_c < t$ **then**

    $\tau_c^* = \arg\min_{\tau_c} \Big| \frac{1}{\mathcal{S}(\tau_c, c)} \sum_{i=1}^K \omega_{y_i}^{\mathcal{V}}$

      $\cdot \mathbb{1}_{ic} \mathbb{1}(y_i = c) \mathbb{1}(\max_j(p_{ij}^{\mathcal{V}}) > \tau_c) - t \Big|$

    ▷ *$\mathcal{S}(\tau_c, c) = \sum_{i=1}^K \omega_{y_i}^{\mathcal{V}} \mathbb{1}_{ic} \mathbb{1}(\max_j(p_{ij}^{\mathcal{V}}) > \tau_c)$ is the normalized number of samples predicted as c with confidence $\geq \tau_c$.*

  else

    $\tau_c^* = 0$ ▷ *The quality of the pseudo-labels is satisfactory, and we make use of all of them.*

  end if

end for

where $\alpha_c = \frac{1}{K_c} \sum_{i=1}^K \mathbb{1}_{ic} \mathbb{1}(y_i = c)$ is the average accuracy of all the samples predicted as class c. Notably, when α_c exceeds threshold t, we think the pseudo-labels are sufficiently accurate to include all instances.

The thresholds optimized with Eq. 14 are inversely related to **Precision** and possess practical utility in handling classes with varying accuracy. We believe this cost function is better suited for fair threshold optimization across diverse class difficulties. This optimization approach introduces no additional hyper-parameters, simply replacing the previous τ with t.

In practical scenarios, we could face difficulties in directly determining the threshold through Eq. 14 due to the imbalances in held-out data and constraints arising from a limited sample size. To address these issues, we employ normalized cost functions and group-based learning, detailed further in Appendix Section D.

After obtaining the optimal refinement parameters, for pseudo-label $\hat{y}_i = \arg\max_j(q_{ij})$ and predicted class $y_i' = \arg\max_j(p_{ij}^{\mathcal{U}})$, we can calculate the unlabelled loss $\hat{R}_{\mathcal{L},\hat{\mathcal{U}}}(f) = \frac{1}{M} \sum_{i=1}^M \mathbb{1}(\max_j(q_{ij}) \geq \tau_{y_i'}) \mathcal{L}(\hat{y}_i, \boldsymbol{p}_i^{\mathcal{U}})$ to update our classification model parameters. The estimation process of $\boldsymbol{\pi}$ and $\boldsymbol{\tau}$ is summarized in Algorithm 1. Note that in Algorithm 1 we rewrite the equations to highlight the importance of normalized cost functions, where a weight term $\boldsymbol{\omega} = 1/\boldsymbol{k}$ is introduced, as discussed in Appendix Section D. Since $\boldsymbol{\pi}$ and $\boldsymbol{\tau}$ address different aspects of the problem, their optimizations can be parallelized for computational efficiency.

4.3 Curriculum Learning

To bypass more data collection efforts, we learn the curriculum of $\boldsymbol{\pi}$ and $\boldsymbol{\tau}$ based on a partition of labelled training dataset $\mathcal{X}$ thus *we do not require additional samples*. Specifically, prior to the standard SSL process, we partition $\mathcal{X}$ into two subset $\mathcal{X}'$ and $\mathcal{V}'$ which contain the same number of samples to learn the curriculum containing L parameters.

In order to ensure curriculum stability, we update the parameters with exponential moving average. Specifically, when we learn a curriculum of length L, after several iterations, we optimize $\boldsymbol{\pi}$ and $\boldsymbol{\tau}$ sequentially based on current model status. We then calculate the curriculum for step l as $\boldsymbol{\pi}^{(l)} = \eta_\pi \boldsymbol{\pi}^{(l-1)} + (1 - \eta_\pi) \boldsymbol{\pi}^{(l)*}$ and $\boldsymbol{\tau}^{(l)} = \eta_\tau \boldsymbol{\tau}^{(l-1)} + (1 - \eta_\tau) \boldsymbol{\tau}^{(l)*}$. We use this to refine pseudo-label before the next SEVAL parameter update. We initialize with FixMatch's thresholds (Sohn et al., 2020) and gradually transition to optimized thresholds to ensure class-wise pseudo-label accuracy exceeds t. We summarize the training process of SEVAL in Algorithm 2.

Algorithm 2 Imbalanced semi-supervised learning with SEVAL.

Require:
$\mathcal{X} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$: labelled training data, $\{\boldsymbol{u}_i\}_{i=1}^M$: unlabelled training data, $f(\cdot)$: network for classification.
t: Requested per class accuracy of the pseudo-label, η_π, η_τ: Momentum decay ratio of offsets and thresholds.
T: Total training iterations, C: Number of classes, L: length of the curriculum.
Initialize the SEVAL parameters as $l = 1$, $\boldsymbol{\pi}^{(l)} = \underbrace{\left[1, 1, \ldots, 1\right]}_{C}$ and $\boldsymbol{\tau}^{(l)} = \underbrace{\left[0.95, 0.95, \ldots, 0.95\right]}_{C}$.

▷ *Estimate a curriculum of the SEVAL parameters based on a partition of the training dataset.*
Randomly partition $\mathcal{X}$ into two subsets, $\mathcal{X}' = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^K$ and $\mathcal{V}' = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^K$, each containing an equal number of data points.
for iter in $[1, \ldots, T]$ **do**
  Calculate the pseudo-label logit for unlabelled data $\mathcal{U}$ and obtain $\{\hat{\boldsymbol{z}}_i^{\mathcal{U}}\}_{i=1}^M$.
  ▷ *Note: FixMatch achieves this by utilizing two augmented versions of the unlabelled data.*
  Calculate the pseudo-label probability $\boldsymbol{q}_i = \sigma(\hat{\boldsymbol{z}}_i^{\mathcal{U}} - \log \boldsymbol{\pi}^{(l)})$.
  For pseudo-label $\hat{y}_i = \arg\max_j q_{ij}$ and predicted class $y'_i = \arg\max_j p_{ij}^{\mathcal{U}}$, calculate the unlabelled loss $\hat{R}_{\mathcal{L},\hat{\mathcal{U}}}(f) = \frac{1}{M}\sum_{i=1}^M \mathbb{1}(\max_j(q_{ij}) \geq \tau_{y'_i}^{(l)})\mathcal{L}(\hat{y}_i, \boldsymbol{p}_i^{\mathcal{U}})$.
  Update the network f with labelled loss $\hat{R}_{\mathcal{L},\mathcal{X}}(f)$ calculated using $\mathcal{X}'$ and $\hat{R}_{\mathcal{L},\hat{\mathcal{U}}}(f)$ via SGD optimizer.
  if iter%$(T/L) = 0$ **then**
    $l =$ **iter**L/T
    Using the exponential moving average version of the network f, compute predictions on $\mathcal{V}'$ and obtain $\{\boldsymbol{z}_i^{\mathcal{V}}\}_{i=1}^K$.
    $\boldsymbol{\pi}^{(l)*}, \boldsymbol{\tau}^{(l)*} = \mathrm{ESTIM}(\mathcal{V}', \{\boldsymbol{z}_i^{\mathcal{V}}\}_{i=1}^K, t)$ ▷ *SEVAL parameter estimation process.*
    $\boldsymbol{\pi}^{(l)} = \eta_\pi \boldsymbol{\pi}^{(l-1)} + (1 - \eta_\pi)\boldsymbol{\pi}^{(l)*}$, $\boldsymbol{\tau}^{(l)} = \eta_\tau \boldsymbol{\tau}^{(l-1)} + (1 - \eta_\tau)\boldsymbol{\tau}^{(l)*}$
  end if
end for
▷ *Standard SSL process.*
for iter in $[1, \ldots, T]$ **do**
  $l = \lceil$**iter**$L/T\rceil$
  Calculate the pseudo-label logit for unlabelled data $\mathcal{U}$ and obtain $\{\hat{\boldsymbol{z}}_i^{\mathcal{U}}\}_{i=1}^M$.
  Calculate the pseudo-label probability $\boldsymbol{q}_i = \sigma(\hat{\boldsymbol{z}}_i^{\mathcal{U}} - \log \boldsymbol{\pi}^{(l)})$.
  Calculate the unlabelled loss $\hat{R}_{\mathcal{L},\hat{\mathcal{U}}}(f) = \frac{1}{M}\sum_{i=1}^M \mathbb{1}(\max_j(q_{ij}) \geq \tau_{y'_i}^{(l)})\mathcal{L}(\hat{y}_i, \boldsymbol{p}_i^{\mathcal{U}})$.
  Update the network f with labelled loss $\hat{R}_{\mathcal{L},\mathcal{X}}(f)$ calculated using $\mathcal{X}$ and $\hat{R}_{\mathcal{L},\hat{\mathcal{U}}}(f)$ via SGD optimizer.
end for
▷ *Post-hoc processing with final learned parameters; skipping over this step results in our SEVAL-PL ablation alternative.*
Given a test sample $\boldsymbol{x}_i$, the logit is adjusted from $\boldsymbol{z}_i$ to $\boldsymbol{z}_i - \log \boldsymbol{\pi}^{(\boldsymbol{L})*}$.

5 Experiments

We conducted main experiments on several imbalanced SSL benchmarks including CIFAR-10-LT, CIFAR-100-LT (Krizhevsky et al., 2009), STL-10-LT (Coates et al., 2011) and ImageNet-127 (Deng et al., 2009) under the same codebase, following (Oh et al., 2022). The training datasets are intentionally resampled from their original balanced versions using an exponential distribution to create class imbalance. We further apply SEVAL to the realistic imbalanced SSL dataset, Semi-Aves (Su & Maji, 2021), which is a dataset for bird species recognition, exhibiting natural class imbalance where some species have over 100 images while others have fewer than 10.

We assume a uniform test label distribution $P^{\mathcal{T}}(Y)$, which aligns with standard test dataset construction and evaluation practices in the literature. For all datasets except ImageNet-127, the test sets are constructed with an equal number of samples per class, and we report accuracy as the primary evaluation metric. For ImageNet-127, where the test dataset is inherently imbalanced, we instead report the averaged class-wise **Recall** to ensure fair performance evaluation.

We choose wide ResNet-28-2 (Zagoruyko & Komodakis, 2016) as the feature extractor and train the network at a resolution of 32×32. We train the neural networks for 250,000 iterations with fixed learning rate of 0.03. We control the imbalance ratios for both labelled and unlabelled data ($\gamma^{\mathcal{X}}$ and $\gamma^{\mathcal{U}}$) and exponentially decrease the number of samples per class. More experiment details are provided in Appendix Section D.

For most experiments, we employ FixMatch to calculate the pseudo-label and make the prediction using the exponential moving average version of the model following (Sohn et al., 2020). We report the average

Table 2: Accuracy on CIFAR10-LT, CIFAR100-LT and STL10-LT. We divide SSL algorithms into different groups including long-tailed learning (LTL), pseudo-label refinement (PLR) and threshold adjustment (THA). PLR and THA based methods only modify pseudo-label probability $\boldsymbol{q}_i$ and threshold τ, respectively. Best results within the same category are in **bold** for each configuration.

Algorithm	Method type			CIFAR10-LT $\gamma^{\mathcal{X}} = \gamma^{\mathcal{U}} = 100$		CIFAR100-LT $\gamma^{\mathcal{X}} = \gamma^{\mathcal{U}} = 10$		STL10-LT $\gamma^{\mathcal{X}} = 20$, $\gamma^{\mathcal{U}}$: *unknown*	
	LTL	PLR	THA	$n_1 = 500$ $m_1 = 4000$	$n_1 = 1500$ $m_1 = 3000$	$n_1 = 50$ $m_1 = 400$	$n_1 = 150$ $m_1 = 300$	$n_1 = 150$ $M = 100,000$	$n_1 = 450$
Supervised				47.3 ±0.95	61.9 ±0.41	29.6 ±0.57	46.9 ±0.22	39.4 ±1.40	51.7 ±2.21
w/ LA (Menon et al., 2020)	✓			53.3 ±0.44	70.6 ±0.21	30.2 ±0.44	48.7 ±0.89	42.0 ±1.24	55.8 ±2.22
FixMatch (Sohn et al., 2020)				67.8 ±1.13	77.5 ±1.32	45.2 ±0.55	56.5 ±0.06	47.6 ±4.87	64.0 ±2.27
w/ DARP (Kim et al., 2020)		✓		74.5 ±0.78	77.8 ±0.63	49.4 ±0.20	58.1 ±0.44	59.9 ±2.17	72.3 ±0.60
w/ FlexMatch (Zhang et al., 2021)			✓	74.0 ±0.64	78.2 ±0.45	49.9 ±0.61	58.7 ±0.24	48.3 ±2.75	66.9 ±2.34
w/ Adsh (Guo & Li, 2022)			✓	73.0 ±3.46	77.2 ±1.01	49.6 ±0.64	58.9 ±0.71	60.0 ±1.75	71.4 ±1.37
w/ FreeMatch (Wang et al., 2022d)		✓	✓	73.8 ±0.87	77.7 ±0.23	49.8 ±1.02	59.1 ±0.59	63.5 ±2.61	73.9 ±0.48
w/ SEVAL-PL		✓	✓	**77.7** ±1.38	**79.7** ±0.53	**50.8** ±0.84	**59.4** ±0.08	**67.4** ±0.79	**75.2** ±0.48
w/ ABC (Lee et al., 2021)	✓			78.9 ±0.82	83.8 ±0.36	47.5 ±0.18	59.1 ±0.21	58.1 ±2.50	74.5 ±0.99
w/ SAW (Lai et al., 2022b)	✓			74.6 ±2.50	80.1 ±1.12	45.9 ±1.85	58.2 ±0.18	62.4 ±0.86	74.0 ±0.28
w/ CReST+ (Wei et al., 2021)	✓	✓		76.3 ±0.86	78.1 ±0.42	44.5 ±0.94	57.1 ±0.65	56.0 ±3.19	68.5 ±1.88
w/ DASO (Oh et al., 2022)	✓	✓		76.0 ±0.37	79.1 ±0.75	49.8 ±0.24	59.2 ±0.35	65.7 ±1.78	75.3 ±0.44
w/ ACR (Wei & Gan, 2023)	✓	✓	✓	80.2 ±0.78	83.8 ±0.13	50.6 ±0.13	60.7 ±0.23	65.6 ±0.11	**76.3** ±0.57
w/ SEVAL	✓	✓	✓	**82.8** ±0.56	**85.3** ±0.25	**51.4** ±0.95	**60.8** ±0.28	**67.4** ±0.69	75.7 ±0.36

test accuracy along with its variance, derived from three distinct random seeds. These random seeds control both dataset splitting and the initialization of optimization algorithms. We ensure rigorous fairness in comparisons by conducting *all* reported experiments using identical codebase implementations under controlled conditions.

5.1 Main Results

We compared SEVAL with different SSL algorithms and summarize the test accuracy results in Table 2. To ensure fair comparison of algorithm performance, in this table, we mark SSL algorithms based on the way they tackle the imbalance challenge. In particular, techniques such as DARP, which exclusively manipulate the probability of pseudo-labels $\boldsymbol{\pi}$, are denoted as PLR. In contrast, approaches like FlexMatch, which solely alter the threshold $\boldsymbol{\tau}$, are termed as THA. We denote other methods that apply regularization techniques to the model's cost function using labelled data as long-tailed learning (LTL). In addition to SEVAL results, we also report the results of SEVAL-PL, which forgoes any post-hoc adjustments on test samples. This ensures that its results are directly comparable with its counterparts.

As shown in Table 2, SEVAL-PL outperforms other PLR and THA based methods such as DARP, FlexMatch and FreeMatch with a considerable margin. This indicates that SEVAL can provide better pseudo-label for the models by learning a better curriculum for $\boldsymbol{\pi}$ and $\boldsymbol{\tau}$. When compared with other hybrid methods including ABC, CReST+, DASO, ACR, SEVAL demonstrates significant advantages in most scenarios. Relying solely on the strength of pseudo-labeling, SEVAL delivers highly competitive performance in the realm of imbalanced SSL. Given its straightforward framework, SEVAL can be integrated with other SSL concepts to enhance accuracy, a point we delve into later in the ablation study.

Semi-Aves contains 200 classes with different long-tailed distribution and captures a situation where a portion of the unlabelled data originates from previously unseen classes. In addition to labelled data, Semi-Aves also contains imbalanced unlabelled data $\mathcal{U}_{\text{in}}$ and unlabelled open-set data $\mathcal{U}_{\text{out}}$ from another 800 classes. Following previous works (Su et al., 2021; Oh et al., 2022), we conducted experiments using $\mathcal{U}_{\text{in}}$ or a combination of $\mathcal{U}_{\text{in}}$ and $\mathcal{U}_{\text{out}}$. We summarize the results in Table 3. This dataset poses a challenge due to the limited number of samples in the tail class, with only around 15 samples per class. It has been observed that SEVAL performs effectively in such a demanding scenario. We additionally present a summary of further experimental results under various realistic settings, along with sensitivity analyses, in Appendix C.

Table 3: Accuracy on Semi-Aves. Best results within the same category are in **bold** for each configuration.

Algorithm	Method type			Semi-Aves	
	LTL	PLR	THA	$\mathcal{U} = \mathcal{U}_{\text{in}}$	$\mathcal{U} = \mathcal{U}_{\text{in}} + \mathcal{U}_{\text{out}}$
FixMatch (Sohn et al., 2020)				59.9 ±0.08	52.6 ±0.14
w/ DARP (Kim et al., 2020)		✓		60.3 ±0.24	54.7 ±0.06
w/ SEVAL-PL		✓	✓	**60.6** ±0.18	**56.4** ±0.10
w/ CReST+ (Wei et al., 2021)	✓	✓		60.0 ±0.03	54.3 ±0.59
w/ DASO (Oh et al., 2022)	✓	✓		59.3 ±0.28	56.6 ±0.32
w/ SEVAL	✓	✓	✓	**60.7** ±0.17	**56.7** ±0.15

Table 4: Accuracy on CIFAR10-LT with different imbalanced ratios. Best results within the same category are in **bold** for each configuration.

Algorithm	Method type			CIFAR10-LT					
				$\gamma^{\mathcal{X}} = 100, \gamma^{\mathcal{U}} = 1$		$\gamma^{\mathcal{X}} = 100, \gamma^{\mathcal{U}} = 1/100$		$\gamma^{\mathcal{X}} = \gamma^{\mathcal{U}} = 150$	
	LTL	PLR	THA	$n_1 = 500$ $m_1 = 4000$	$n_1 = 1500$ $m_1 = 3000$	$n_1 = 500$ $m_1 = 4000$	$n_1 = 1500$ $m_1 = 3000$	$n_1 = 500$ $m_1 = 4000$	$n_1 = 1500$ $m_1 = 3000$
FixMatch (Sohn et al., 2020)				73.0 ±3.81	81.5 ±1.15	62.5 ±0.94	71.8 ±1.70	62.9 ±0.36	72.4 ±1.03
w/ DARP (Kim et al., 2020)		✓		82.5 ±0.75	84.6 ±0.34	70.1 ±0.22	80.0 ±0.93	67.2 ±0.32	73.6 ±0.73
w/ SEVAL-PL		✓	✓	**89.4** ±0.53	**89.2** ±0.02	**77.7** ±0.91	**80.9** ±0.66	**71.9** ±1.10	**74.7** ±0.63
w/ CReST+ (Wei et al., 2021)	✓	✓		82.2 ±1.53	86.4 ±0.42	62.9 ±1.39	72.9 ±2.00	67.5 ±0.45	73.7 ±0.34
w/ DASO (Oh et al., 2022)	✓	✓		86.6 ±0.84	88.8 ±0.59	71.0 ±0.95	80.3 ±0.65	70.1 ±1.81	75.1 ±0.77
w/ SEVAL	✓	✓	✓	**90.3** ±0.61	**90.6** ±0.47	**79.2** ±0.83	**82.9** ±1.78	**79.8** ±0.42	**83.3** ±0.40

5.2 Varied Imbalanced Ratios

Similar to results in Table 2, we evaluate SEVAL on CIFAR10-LT with different imbalanced ratios and summarize results in Table 4. We find that SEVAL consistently outperforms its counterparts across different $\gamma^{\mathcal{X}}$ values. Since SEVAL does not make any assumptions about the distribution of unlabeled data, it can be robustly implemented in scenarios where $\gamma^{\mathcal{X}} \neq \gamma^{\mathcal{U}}$. In these settings, SEVAL's performance advantage over its counterparts is even more pronounced.

5.3 Low Labelled Data Scheme

SEVAL acquires a curriculum of parameters by partitioning the training dataset. This raises a crucial question: can SEVAL remain effective with a very limited number of labeled samples? To explore this, we conduct a stress test by training SEVAL with a minimal amount of labeled data.

In the first experimental configuration, we keep the imbalance ratio constant while reducing the number of labeled samples ($n_1 = 200$). In this extreme case, only two samples are labeled for the tail class. In the second configuration, we use a balanced labeled training dataset, but with a total of 100 and 40 samples for training. The results are summarized in Table 5. We find that SEVAL performs well in both scenarios, indicating that SEVAL can be a reliable option even when the labeled dataset is very small.

5.4 Performance Analysis

To closely examine the distinct contributions of $\boldsymbol{\pi}$ and $\boldsymbol{\tau}$, we carry out an ablation study where SEVAL optimizes just one of them, respectively termed SEVAL-PLR and SEVAL-THA. Note that FixMatch's default setting maintains a constant π of 1.0 and τ of 0.95 for all classes. As summarized in Table 6, SEVAL-PLR and SEVAL-THA can still outperform their counterparts, DARP and FlexMatch, respectively. When tuning both parameters, SEVAL-PL can achieve the best results.

Table 5: Accuracy on CIFAR10-LT under the setting of extremely few labelled samples. Best results within the same category are in **bold** for each configuration.

Algorithm	Method type			CIFAR10-LT		
				$\gamma^{\mathcal{X}} = \gamma^{\mathcal{U}} = 100$	$\gamma^{\mathcal{X}} = 1, \gamma^{\mathcal{U}} = 100$	
	LTL	PLR	THA	$n_1 = 200$ $m_1 = 4000$	$n_1 = 10$ $m_1 = 4000$	$n_1 = 4$ $m_1 = 4000$
FixMatch (Sohn et al., 2020)				64.3 ±0.83	65.3 ±0.80	44.7 ±3.33
w/ FreeMatch (Wang et al., 2022d)		✓	✓	67.4 ±1.09	58.4 ±0.76	50.7 ±1.95
w/ SEVAL-PL		✓	✓	**69.3** ±0.66	**68.3** ±0.56	**51.5** ±1.51
w/ DASO (Oh et al., 2022)	✓	✓		67.2 ±1.25	61.2 ±0.96	48.6 ±2.81
w/ SEVAL	✓	✓	✓	**71.2** ±0.80	**68.9** ±0.25	**52.7** ±1.83

Table 6: Comparison of SEVAL when only optimizing $\boldsymbol{\pi}$ (SEVAL-PLR) or only optimizing $\boldsymbol{\tau}$ (SEVAL-THA). SEVAL outperforms counterparts with identical parameter settings under different imbalanced SSL scenarios. SEVAL-PL, with its sequential optimization of both $\boldsymbol{\pi}$ and $\boldsymbol{\tau}$, yields further improvements in accuracy.

Algorithm	Method type			CIFAR10-LT	CIFAR100-LT
	LTL	PLR	THA	$n_1 = 500$ $m_1 = 4000$ $\gamma^{\mathcal{X}} = \gamma^{\mathcal{U}} = 100$	$n_1 = 150$ $m_1 = 300$ $\gamma^{\mathcal{X}} = \gamma^{\mathcal{U}} = 10$
FixMatch (Sohn et al., 2020)				67.8 ±1.13	56.5 ±0.06
w/ DARP (Kim et al., 2020)		✓		74.5 ±0.78	58.1 ±0.44
w/ SEVAL-PLR		✓		76.7 ±0.82	59.3 ±0.30
w/ FlexMatch (Zhang et al., 2021)			✓	74.0 ±0.64	58.7 ±0.24
w/ SEVAL-THA			✓	77.0 ±0.93	59.1 ±0.18
w/ SEVAL-PL		✓	✓	**77.7** ±1.38	**59.4** ±0.08

5.4.1 Pseudo-Label Refinement

In order to comprehensively and quantitatively investigate the accuracy of pseudo-label refined by different approaches, here we define G as the sum of accuracy gain and balanced accuracy gain of pseudo-label over training iterations. Both sample-wise accuracy and class-wise accuracy are crucial measures for evaluating the quality of pseudo-labels. A low sample-specific accuracy can lead to noisier pseudo-labels, adversely affecting model performance. Meanwhile, a low class-specific accuracy often indicates a bias towards the dominant classes. Therefore, we propose a metric G which is the combination of these two metrics. Specifically, given the pseudo-label $\hat{y}_i$ and predicted class y'_i of unlabelled dataset $\mathcal{U}$, we calculate G as:

$$G = \underbrace{\frac{\sum_{i=1}^{M}[\mathbb{1}(\hat{y}'_i = y_i) - \mathbb{1}(\hat{y}_i = y_i)]}{M}}_{\text{Sample-Wise Accuracy Gain}} + \underbrace{\sum_{c=1}^{C}\sum_{i=1}^{M}\frac{\mathbb{1}(\hat{y}'_i = c)\mathbb{1}(\hat{y}'_i = y_i) - \mathbb{1}(\hat{y}_i = c)\mathbb{1}(\hat{y}_i = y_i)}{m_c C}}_{\text{Class-Wise Accuracy Gain}}. \tag{15}$$

To evaluate the cumulative impact of pseudo-labels, we calculate $\mathbf{Gain}(\mathbf{iter})$ as the accuracy gain at training iteration $\mathbf{iter}$ and monitor $\mathbf{Gain}(\mathbf{iter}) = \sum_{j=1}^{\mathbf{iter}} G(j)/\mathbf{iter}$ throughout the training iterations. The results of SEVAL along with DARP and adjusting pseudo-label logit $\hat{z}_c^{\mathcal{U}}$ with LA are summarized in Fig. 4(a). We note

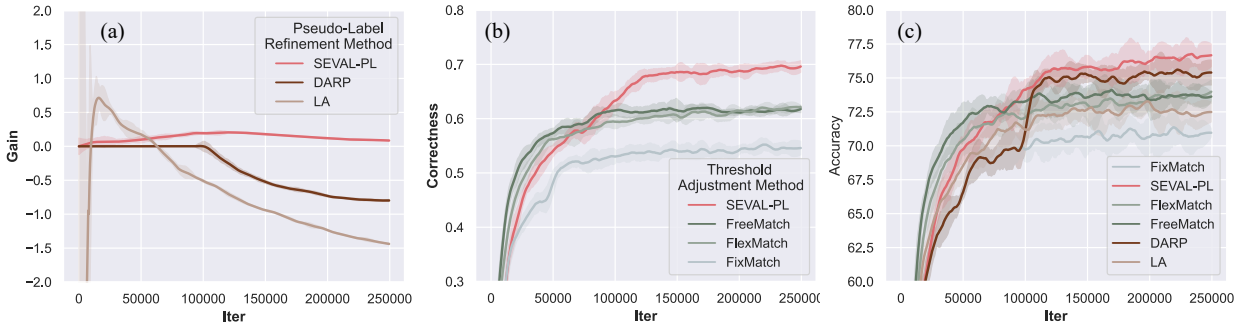

Figure 4: (a) The evolution of **Gain** across training iterations. SEVAL accumulates a higher accuracy of pseudo-label than its counterparts. (b) The evolution of **Correctness** across training iterations. SEVAL can build better trade-off between quality and quantity. (c) The evolution of test accuracy across training iterations. SEVAL-PL outperforms other pseudo-label refinement methods.

that SEVAL consistently delivers a positive **Gain** throughout the training iterations. In contrast, DARP and LA tend to reduce the accuracy of pseudo-labels during the later stages of the training process. After a warm-up period, DARP adjusts the distribution of pseudo-labels to match the inherent distribution of unlabelled data. However, it doesn't guarantee the accuracy of the pseudo-labels, thus not optimal. While LA can enhance class-wise accuracy, it isn't always the best fit for every stage of the model's learning. Consequently, noisy pseudo-labels from the majority class can impede the model's training. SEVAL learns a smooth curriculum of parameters for pseudo-label refinement from the data itself, therefore bringing more stable improvements. The test accuracy curves in Fig. 4(c) provide further validation of SEVAL's effectiveness, with SEVAL-PL demonstrating superior performance over both LA and DARP.

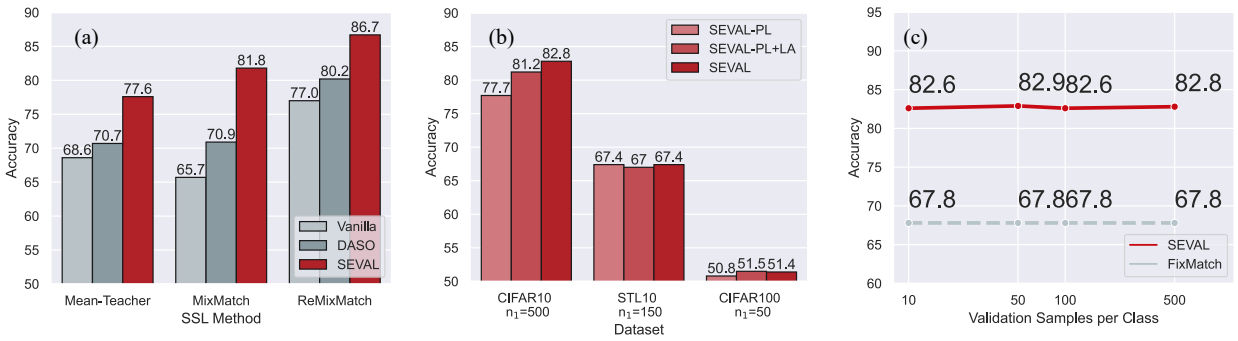

Figure 5: (a) Test accuracy when SEVAL is adapted to pseudo-label based SSL algorithms other than FixMatch under the setting of CIFAR-10 $n_1 = 1500$. SEVAL can readily improve the performance of other SSL algorithsm. (b) Test accuracy when SEVAL employs varied types of post-hoc adjustment parameters. The learned post-hoc parameters consistently enhance performance, particularly in CIFAR-10 experiments. (c) Test accuracy when SEVAL is optimized using different validation samples under the setting of CIFAR-10 $n_1 = 500$. SEVAL requires few validation samples to learn the optimal curriculum of parameters.

5.4.2 Threshold Adjustment

Quantity and quality are two essential factors for pseudo-labels, as highlighted in (Chen et al., 2023). Quantity refers to the number of correctly labeled samples produced by pseudo-label algorithms, while quality indicates the proportion of correctly labeled samples after applying confidence-based thresholding.

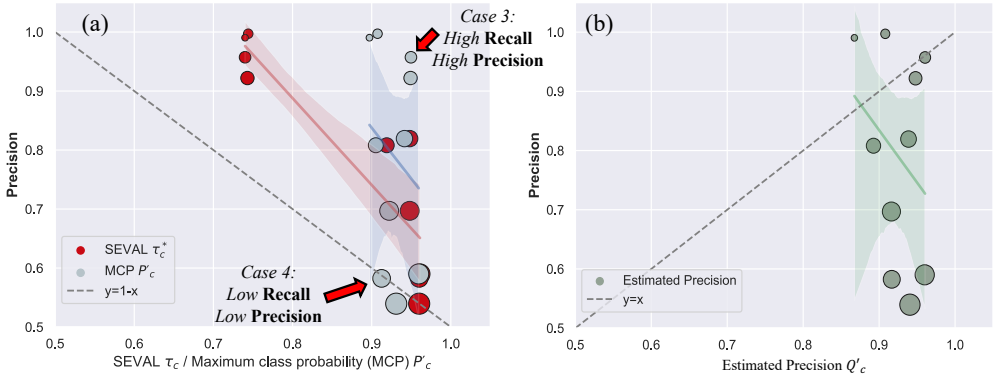

Figure 6: The correlation of different metrics between test **Precision** of FixMatch on CIFAR10-LT $n_1 = 500$. (a) The correlation of SEVAL learned τ_c and maximum class probability P'_c between test **Precision**. Each point represents a class c and the size of the points indicate the number of samples in the labelled training dataset n_c. Note that maximum class probability P'_c is the basis of current dynamic threshold method to derive thresholds. For example, FlexMatch selects more samples for classes associated with lower P'_c. However, as highlighted by red arrows, P'_c does not correlated with **Precision** thus P_c based on methods will fail *Case 3: High* **Recall** & *High* **Precision** and *Case 4: Low* **Recall** & *Low* **Precision** in accordance with Fig. 2. (b) Due to the lack of calibration in the network output probability, the estimated precision derived from the probability does not align with the actual **Precision**, thus cannot be a reliable metric to directly derive thresholds.

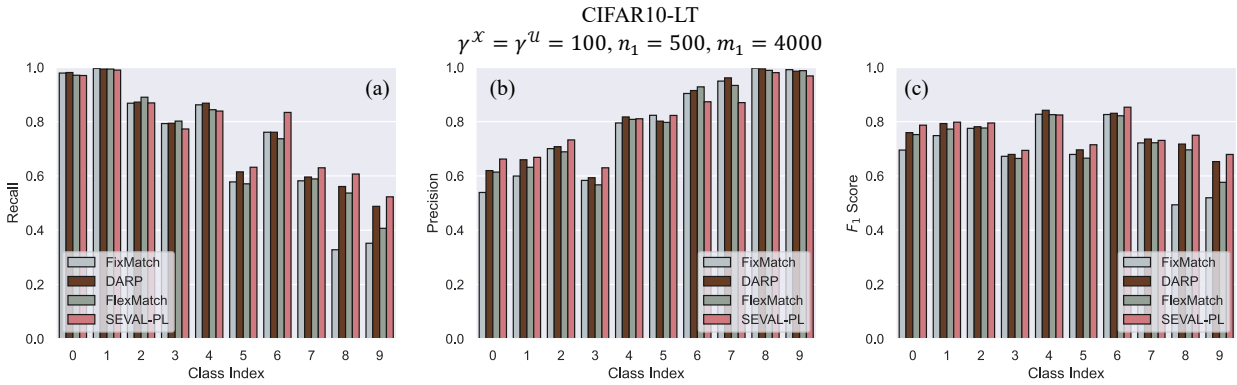

Figure 7: Class-wise performance for different SSL methods. Class indexes are arranged in descending order according to their class frequencies. We find that SEVAL achieve better overall performance than its counterparts by making neural networks more sensitive to minority classes.

In order to access the effectiveness of pseudo-label, we propose a metric called **Correctness**, which is a combination of quantity and quality. Having just high quantity or just high quality isn't enough for effective pseudo-labels. For instance, setting exceedingly high thresholds might lead to the selection of a limited number of accurately labelled samples (high quality). However, this is not always the ideal approach, and the opposite holds true for quantity. Therefore, the proposed **Correctness** metric combines both quality and quantity for fair evaluation. In particular, factoring in the potential imbalance of unlabelled data, we utilize a class frequency based weight term $\boldsymbol{\omega}^{\mathcal{U}} = 1/\boldsymbol{m}$ to normalize this metric, yielding:

$$\textbf{Correctness} = \underbrace{\frac{\mathcal{C}}{\sum_{i=1}^{M} \omega_{y_i}^{\mathcal{U}}}}_{\text{Quantity}} \underbrace{\frac{\mathcal{C}}{\sum_{i=1}^{M} \omega_{y_i}^{\mathcal{U}} \mathbb{1}(\max_j(q_{ij}) \geq \tau_{y_i'})}}_{\text{Quality}}, \tag{16}$$

where, $\mathcal{C} = \sum_{i=1}^{M} \omega_{y_i}^{\mathcal{U}} \mathbb{1}(\hat{y}_i = y_i)\mathbb{1}(\max_j(q_{ij}) \geq \tau_{y_i'})$ is the relative number of correctly labelled samples. We show **Correctness** of SEVAL with FixMatch, FlexMatch and FreeMatch in Fig. 4(b). We observe that FlexMatch and FreeMatch can both improve **Correctness**, while SEVAL can boost even more. We observe that the test accuracy follows a trend similar to **Correctness**, as shown in Fig. 4(c). This demonstrates that the thresholds set by SEVAL not only ensure a high quantity but also attain high accuracy for pseudo-labels, making them efficient in the model's learning process.

5.5 Ablation Study

5.5.1 Flexibility and Compatibility

We apply SEVAL to other pseudo-label based SSL algorithms including Mean-Teacher, MixMatch and ReMixMatch and report the results with the setting of CIFAR-100 $n_1 = 50$ in Fig. 5(a). We find SEVAl can bring substantial improvements to these methods and is more effective than DASO. Of note the results of ReMixMatch w/SEVAL is higher than the results of FixMatch w/ SEVAL in Table 2 (86.7 vs 85.3). This may indicates that ReMixMatch is fit imbalanced SSL better. Due to its simplicity, SEVAL can be readily combined with other SSL algorithms that focus on LTL instead of PLR and THA. For example, SEVAL pairs effectively with the semantic alignment regularization introduced by DASO. By incorporating this loss into our FixMatch experiments with SEVAL, we were able to boost the test accuracy from 51.4 to 52.4 using the CIFAR-100 $n_1 = 50$ configuration.

We compare with the post-hoc adjustment process with LA in Fig. 5(b). We find that those post-hoc parameters can improve the model performance in the setting of CIFAR-10. In other cases, our post-hoc adjustment doesn't lead to a decrease in prediction accuracy. However, LA sometimes does, as seen in the case of STL-10. This likely stems from the complexity of the confusion matrix in such cases, where simple offsets fail to sufficiently address class bias.

We should acknowledge that for each setting, we only conduct experiments with fixed hyper-parameters (i.e., learning rate and batch size). This may be a limitation, as different hyper-parameters could lead to different results (Sohn et al., 2020). We summarize all hyper-parameters in detail in Appendix Section D, with the hope that others can build upon our work or extend the experiments to other settings.

5.5.2 Data-Efficiency

Here we explore if SEVAL requires a substantial number of validation samples for curriculum learning. To do so, we keep the training dataset the same and optimize SEVAL parameters using balanced validation dataset with varied numbers of labelled samples using the CIFAR-10 $n_1 = 500$ configuration, as shown in Fig. 5(c). We find that SEVAL consistently identifies similar $\boldsymbol{\pi}$ and $\boldsymbol{\tau}$. When we train the model using these curricula, there aren't significant differences even when the validation samples per class ranges from 10 to 500. This suggests that SEVAL is both data-efficient and resilient.

5.6 Analysis of Learned Thresholds

We try to determine the effectiveness of thresholds by looking into **Precision** of different classes, which should serve as approximate indicators of suitable thresholds. We illustrate an example of optimized thresholds and the learning status of FixMatch on CIFAR10-LT $n_1 = 500$ in Fig. 6(a), where SEVAL learns τ_c to be low for classes that have high **Precision**. In contrast, maximum class probability P_c', does not show clear correction with **Precision**. Specifically, as highlighted with the red arrows, P_c remains high for classes that exhibit high **Precision**. Consequently, maximum class probability-based threshold methods such as FlexMatch will tune the threshold to be high for classes with large P_c, inadequately addressing *Case 3* and *Case 4* as elaborated in Section 3.

Instead of depending on an independent labelleddataset, we also attempt to estimate **Precision** using the model probability, so as to leverage the estimated precision to determine the appropriate thresholds. Specifically, we estimate the **Precision** of class c as:

$$Q_c' = \frac{\sum_{i=1}^{K} \mathbb{1}_{ic} \max_j p_{ij}^{\mathcal{U}}}{\sum_{i=1}^{K} p_{ic}^{\mathcal{U}}}. \tag{17}$$

We visualize the estimated precision in Fig. 6(b). We find the the estimated precision does not align with the actual **Precision**. This is because the model is uncalibrated and the Q_c' is heavily decided by the true positives parts (e.g. numerator in Eq. 17), thus cannot reflect the real model precision. Thus it is a essential to utilize a holdout labelled dataset to derive the optimal thresholds.

Finally, we look into the class-wise performance of SEVAL and its counterparts in Fig. 7. When compared with alternative methods, SEVAL achieves overall better performance with higher **Recall** on minority classes and higher **Precision** on majority classes. In this case, class 6 falls into *Case 3: High* **Recall** & *High* **Precision** while class 5 falls into *Case 4: Low* **Recall** & *Low* **Precision**. SEVAL shows advantages on these classes.

6 Conclusion and Future Work

In this study, we provide a theoretical analysis of pseudo-labeling strategies and introduce SEVAL, grounded in statistical principles, demonstrating its benefits for imbalanced SSL across diverse application scenarios. SEVAL sheds new light on pseudo-label generalization, which is a foundation for many leading SSL algorithms. SEVAL is both straightforward and potent, requiring no extra computation once the curriculum is acquired. As such, it can be effortlessly integrated into other SSL algorithms and paired with LTL methods to address class imbalance.

We believe that the concept of optimizing parameters or accessing unbiased learning status using a partition of the labeled training dataset could spark further innovations in long-tailed recognition and SSL. For example, our PLR optimization could be adapted to learn class-specific margins (i.e., LA during training), as discussed in LDAM (Cao et al., 2019) and other approaches (Li et al., 2020; Menon et al., 2020). This adaptation would enhance neural network learning under class imbalance and consequently benefit imbalanced SSL by yielding less biased classifiers.

We feel that the specific interplay between label refinement and threshold adjustment remains an intriguing question for subsequent research. In practice, we believe a more sophisticated curriculum incorporating a warm-up phase without unlabeled samples in early stages could further improve performance (Cascante-Bonilla et al., 2021). However, we consistently use baseline methods throughout our experiments to ensure clear interpretation of results.

In the future, by leveraging Bayesian or bootstrap techniques, we may eliminate the need for internal validation in SEVAL by improving model calibration (Loh et al., 2022; Vucetic & Obradovic, 2001). We also plan to analyze SEVAL within the theoretical framework of SSL (Mey & Loog, 2022) to acquire deeper insights. Moreover, we also want to investigate whether our findings hold and are extendable to pseudo-labels generated by pre-trained foundation models, as these models are also trained on long-tailed datasets from the wild (Wang et al., 2022b).

Appendix

Table of Contents

A Glossary of Notations **23**

B Proofs **24**
B.1 Proof of Proposition 1 24
B.2 Proof of Corollary 1 24
B.3 Proof of Theorem 1 24
B.4 Proof of Lemma 1 25

C Additional Experiments **26**
C.1 Sensitivity Analysis 26
C.2 Results on ImageNet-127 26
C.3 Integration with Other SSL Frameworks 26

D Implementation Details **27**
D.1 Learning with Imbalanced Held-Out Data 27
D.2 Learning Thresholds within Groups 27
D.3 Benchmarks 29
D.4 Hyper-Parameters 29
D.5 SEVAL with Other SSL Algorithms 29

E Classes of Different Performance **30**

A Glossary of Notations

Notation	Description	Section
$f(\cdot)$	A classification model.	§ 1
$\sigma(\cdot)$	Softmax calculation.	§ 3.1
C	Number of classes.	§ 3.1
$\boldsymbol{n} \in \mathbb{R}^C$	Number of samples per class of $\mathcal{X}$.	§ 3.1
γ	Class imbalance ratio.	§ 3.2
ρ	Noise rate of pseudo-labels.	§ 3.3
$\boldsymbol{k} \in \mathbb{R}^C$	Number of samples per class of $\mathcal{V}$.	§ 4.1
$\boldsymbol{\omega}^{\mathcal{V}} \in \mathbb{R}^C$	Class weights as the reciprocal of $\boldsymbol{k}$.	§ 4.2, D
$X \times Y$	Input and label space.	§ 3.1
$P(Y\|X)$	Conditional probability distribution.	§ 3.1
$P(X, Y)$	Joint Probability Distribution.	§ 3.2
$P(Y)$	Marginal Probability Distribution.	§ 3.2
$\mathcal{X} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$	Labelled training dataset.	§ 3.1
$\hat{\mathcal{U}} = \{(\boldsymbol{u}_i, \hat{y}_i)\}_{i=1}^M$	Unlabelled training data and its associated pseudo-labels.	§ 3.1
$\mathcal{U} = \{(\boldsymbol{u}_i, y_i)\}_{i=1}^M$	Unlabelled training data alongside its oracle labels.	§ 3.3
$\mathcal{T}$	Test data.	§ 3.2
$\mathcal{V}$	An independent labelled data, split from $\mathcal{X}$.	§ 3.2
$\mathcal{L}$	Supervised loss (e.g., cross-entropy).	§ 3.1
$R_{\mathcal{L},\mathcal{X}}(f)$, $\hat{R}_{\mathcal{L},\hat{\mathcal{U}}}(f)$	Empirical risk computed on labeled and unlabeled data.	§ 3.1
$\mathcal{O}_w(\cdot), \mathcal{O}_s(\cdot)$	Weak and strong operations for data augmentation.	§ 3.1
$\boldsymbol{p}_i \in \mathbb{R}^C$	Model predicted probabilities.	§ 3.1
$\boldsymbol{q}_i \in \mathbb{R}^C$	Pseudo-label probability.	§ 3.1
$\boldsymbol{\tau} \in \mathbb{R}^C$	Class-specific filtering thresholds for pseudo-labels.	§ 3.1, 4.2
$\hat{\boldsymbol{z}}_i \in \mathbb{R}^C$	Pseudo-label logit used to derive $\boldsymbol{q}_i$.	§ 3.2
$\boldsymbol{\pi} \in \mathbb{R}^C$	Offsets for sample selection based on $\hat{\boldsymbol{z}}_i^{\mathcal{U}}$.	§ 3.2, 4.1
t	The expected accuracy of pseudo-labels.	§ 4.2
L	The length of curriculum.	§ 4.3
T	Total training iterations.	§ 4.3
η_π, η_τ	Momentum decay ratio of offsets and thresholds.	§ 4.3
λ	The Lagrange multiplier.	§ 4.2
$\mathcal{A}(\tau_c, c)$	**Precision** for class c when selecting samples with threshold τ_c.	§ 4.2
$\mathcal{S}(\tau_c, c)$	Number of class c samples selected by τ_c.	§ 4.2, D
P'_c	Maximum class probability of class c, estimating **Recall**.	§ 3.3
Q'_c	The estimation of **Precision** for class c.	§ 5.6
$\mathbb{1}_{ic}$	Whether sample i's most probable class is c.	§ 3.3, 4.2
α_c	Accuracy of samples predicted as class c.	§ 4.2
K_c	(Normalized) number of samples predicted as c.	§ 3.3, D
B	The size of the class group.	§ D
$\tilde{\boldsymbol{\tau}} \in \mathbb{R}^{\lceil C/B \rceil}$	Group-specific filtering thresholds for pseudo-labels.	§ D
$\tilde{\mathcal{S}}(\tilde{\tau}_b, b)$	Number of group b samples selected by τ_b.	§ D
$\tilde{\alpha}_b$	Accuracy of samples predicted within group b.	§ D
$\tilde{K}_b$	(Normalized) number of samples predicted within group c.	§ D

B Proofs

B.1 Proof of Proposition 1

$$
\begin{aligned}
f^{\mathcal{T}}(X) &= P^{\mathcal{T}}(Y|X) \\
&= \frac{P^{\mathcal{T}}(X|Y)P^{\mathcal{T}}(Y)}{P^{\mathcal{T}}(X)} \\
&= \frac{P^{\mathcal{T}}(X|Y)P^{\mathcal{T}}(Y)}{P^{\mathcal{T}}(X)} \cdot \frac{P^{\mathcal{X}}(X)}{P^{\mathcal{X}}(X)},
\end{aligned} \tag{18}
$$

when we assume there does not exist conditional shifts between training and test dataset following (Saerens et al., 2002), e.g. $P_{\mathcal{X}}(X|Y) = P_{\mathcal{T}}(X|Y)$, we can rewrite Eq. 18 as:

$$
\frac{P^{\mathcal{X}}(X|Y)P^{\mathcal{T}}(Y)}{P^{\mathcal{T}}(X)} \cdot \frac{P^{\mathcal{X}}(X)}{P^{\mathcal{X}}(X)}. \tag{19}
$$

Since there is no covariate shift between the training and test domains, the relative density of X is preserved across domains (i.e., $P_{\mathcal{T}}(X) \propto P_{\mathcal{X}}(X)$). Therefore, Eq. 19 can be rewritten as:

$$
\begin{aligned}
&\frac{P^{\mathcal{X}}(X|Y)P^{\mathcal{T}}(Y)}{P^{\mathcal{X}}(X)} \cdot \frac{P^{\mathcal{X}}(X)}{P^{\mathcal{T}}(X)} \\
&\propto \frac{P^{\mathcal{X}}(X|Y)P^{\mathcal{T}}(Y)}{P^{\mathcal{X}}(X)} \\
&= \frac{P^{\mathcal{X}}(X,Y)P^{\mathcal{T}}(Y)}{P^{\mathcal{X}}(X)P^{\mathcal{X}}(Y)} \\
&= \frac{P^{\mathcal{X}}(Y|X)P^{\mathcal{T}}(Y)}{P^{\mathcal{X}}(Y)} \\
&= \frac{f^{*}(X)P^{\mathcal{T}}(Y)}{P^{\mathcal{X}}(Y)}.
\end{aligned} \tag{20}
$$

■

B.2 Proof of Corollary 1

$$
\begin{aligned}
&P^{\mathcal{T}}(X,Y) \\
&= P^{\mathcal{T}}(X|Y)P^{\mathcal{T}}(Y).
\end{aligned} \tag{21}
$$

If we assume the test distribution $\mathcal{T}$ shares identical class conditionals with the unlabeled training dataset $\mathcal{U}$ (i.e. $P^{\mathcal{T}}(X|Y) = P^{\mathcal{U}}(X|Y)$), Eq. 21 can be rewritten as:

$$
\begin{aligned}
&P^{\mathcal{T}}(X|Y)P^{\mathcal{T}}(Y) \\
&= P^{\mathcal{U}}(X|Y)P^{\mathcal{T}}(Y) \\
&= \frac{P^{\mathcal{U}}(X,Y)P^{\mathcal{T}}(Y)}{P^{\mathcal{U}}(Y)}.
\end{aligned} \tag{22}
$$

Therefore, a Bayes classifier $f^{\mathcal{T}}$ that is optimal on $P^{\mathcal{T}}(X,Y)$ should also be optimal for the resampled unlabeled dataset with distribution $P^{\mathcal{U}}(X,Y)P^{\mathcal{T}}(Y)/P^{\mathcal{U}}(Y)$.

■

B.3 Proof of Theorem 1

We derive the generalization bound based on the Rademacher complexity method (Bartlett & Mendelson, 2002) following the analysis for noisy label training (Natarajan et al., 2013; Liu & Tao, 2015).

Lemma 1 *We define:*

$$\hat{\mathcal{L}}(\cdot, y) := \frac{(1-\rho)\mathcal{L}(\cdot, y) - \rho\mathcal{L}(\cdot, 3-y)}{1-2\rho}. \tag{23}$$

With this loss function, we should have:

$$\mathbb{E}_{\hat{y}}[\hat{\mathcal{L}}(\cdot, \hat{y})] = \mathcal{L}(\cdot, y). \tag{24}$$

Given $\hat{R}_{\hat{\mathcal{L}},\hat{\mathcal{U}}}(f) := \frac{1}{M}\sum_{i=1}^{M}\hat{\mathcal{L}}(f(\boldsymbol{u}_i), \hat{y}_i)$ the empirical risk on $\hat{\mathcal{U}}$ and $R_{\hat{\mathcal{L}},\hat{\mathcal{U}}}(f)$ is the corresponding expected risk, we have the basic generalization bound as:

$$\max_{f\in\mathcal{F}} |\hat{R}_{\hat{\mathcal{L}},\hat{\mathcal{U}}}(f) - R_{\hat{\mathcal{L}},\hat{\mathcal{U}}}(f)| \leq 2\mathfrak{R}(\hat{\mathcal{L}}\circ\mathcal{F}) + \sqrt{\frac{\log(1/\delta)}{2M}}, \tag{25}$$

where:

$$\mathfrak{R}(\hat{\mathcal{L}}\circ\mathcal{F}) := \mathbb{E}_{\boldsymbol{u}_i,\hat{y}_i,\epsilon_i}\left[\sup_{f\in\mathcal{F}}\frac{1}{M}\sum_{i=1}^{M}\epsilon_i\hat{\mathcal{L}}(f(\boldsymbol{u}_i), \hat{y}_i)\right]. \tag{26}$$

If $\mathcal{L}$ is L-Lipschitz then $\hat{\mathcal{L}}$ is L_ρ Lipschitz with:

$$L_\rho = \frac{L}{1-2\rho}. \tag{27}$$

Based on Talagrand's Lemma (Mohri et al., 2018), we have:

$$\mathfrak{R}(\hat{\mathcal{L}}\circ\mathcal{F}) \leq L_\rho\mathfrak{R}(\mathcal{F}), \tag{28}$$

where $\mathfrak{R}(\mathcal{F}) := \mathbb{E}_{\boldsymbol{x}_i,\epsilon_i}\left[\sup_{f\in\mathcal{F}}\frac{1}{M}\sum_{i=1}^{M}\epsilon_i f(\boldsymbol{u}_i)\right]$ is the Rademacher complexity for function class $\mathcal{F}$ and $\epsilon_1, \ldots, \epsilon_M$ are i.i.d. Rademacher variables.

Let $\hat{f}$ be the model after optimizing with $\hat{\mathcal{U}}$, and let f^* be the minimization of the expected risk $R_{\mathcal{L},\mathcal{U}}$ over $\mathcal{F}$. Then, we have:

$$\begin{aligned}
&R_{\mathcal{L},\mathcal{U}}(\hat{f}) - R_{\mathcal{L},\mathcal{U}}(f^*) \\
&= R_{\hat{\mathcal{L}},\hat{\mathcal{U}}}(\hat{f}) - R_{\hat{\mathcal{L}},\hat{\mathcal{U}}}(f^*) \\
&= \left(\hat{R}_{\hat{\mathcal{L}},\hat{\mathcal{U}}}(\hat{f}) - \hat{R}_{\hat{\mathcal{L}},\hat{\mathcal{U}}}(f^*)\right) + \\
&\quad \left(\hat{R}_{\hat{\mathcal{L}},\hat{\mathcal{U}}}(f^*) - R_{\hat{\mathcal{L}},\hat{\mathcal{U}}}(f^*)\right) + \left(R_{\hat{\mathcal{L}},\hat{\mathcal{U}}}(\hat{f}) - \hat{R}_{\hat{\mathcal{L}},\hat{\mathcal{U}}}(\hat{f})\right) \\
&\leq 0 + 2\max_{f\in\mathcal{F}} |\hat{R}_{\hat{\mathcal{L}},\hat{\mathcal{U}}}(f) - R_{\hat{\mathcal{L}},\hat{\mathcal{U}}}(f)|
\end{aligned} \tag{29}$$

After we combine Eq. 25 and Eq. 29, we obtain the bound described in Theorem 1.

■

B.4 Proof of Lemma 1

Based on Eq. 24, we calculate the cases for $y = 1$ and $y = 2$ and obtain:

$$(1-\rho)\hat{\mathcal{L}}(\cdot, 1) + \rho\hat{\mathcal{L}}(\cdot, 2) = \mathcal{L}(\cdot, 1), \tag{30}$$

$$(1-\rho)\hat{\mathcal{L}}(\cdot, 2) + \rho\hat{\mathcal{L}}(\cdot, 1) = \mathcal{L}(\cdot, 2). \tag{31}$$

By solving the two equations, we yield:

$$\hat{\mathcal{L}}(\cdot, 1) = \frac{(1-\rho)\mathcal{L}(\cdot, 1) - \rho\mathcal{L}(\cdot, 2)}{1-2\rho}, \tag{32}$$

$$\hat{\mathcal{L}}(\cdot, 2) = \frac{(1-\rho)\mathcal{L}(\cdot, 2) - \rho\mathcal{L}(\cdot, 1)}{1-2\rho}. \tag{33}$$

■

C Additional Experiments

In this section, we present additional experimental results conducted under various settings to assess the generalizability of SEVAL.

C.1 Sensitivity Analysis

Table 7: Sensitivity analysis of hyper-parameters t, η_π and η_τ. Best results are in **bold** for each configuration.

Hyper-parameter	CIFAR10-LT, $\gamma^{\mathcal{X}} = 100$ $n_1 = 500$, $m_1 = 4000$
$t = 0.6$	82.5 ±0.45
$t = 0.7$	82.2 ±0.11
$t = 0.75$ (reported)	**82.8** ±0.56
$\eta_\pi = 0.995$	81.4 ±0.36
$\eta_\pi = 0.999$ (reported)	**82.8** ±0.56
$\eta_\pi = 0.9995$	82.5 ±0.35
$\eta_\tau = 0.995$	81.5 ±0.38
$\eta_\tau = 0.999$ (reported)	82.8 ±0.56
$\eta_\tau = 0.9995$	**82.9** ±0.09

We perform experiments with SEVAL, varying the core hyperparameters, and present the results in Table 7. Our findings indicate that SEVAL exhibits robustness, showing insensitivity to hyper-parameter variations within a reasonable range.

C.2 Results on ImageNet-127

We conduct experiments on Small ImageNet-127 (Su & Maji, 2021) with a reduced image size of 32×32 following (Fan et al., 2022). ImageNet-127 consolidates the 1000 classes of ImageNet into 127 categories according to the WordNet hierarchy. This results in a naturally long-tailed class distribution with an imbalance ratio of $\gamma^{\mathcal{X}} = \gamma^{\mathcal{U}} = 286$. We randomly select 10% of the training samples as the labeled set and utilize the remainder as unlabeled data. We conduct experiments using ResNet-50 and Adam optimizer. Results are summarized in Table 8.

Table 8: Averaged class recall on Small-ImageNet-127. Best results within the same category are in **bold** for each configuration.

Algorithm	Method type			Small-ImageNet-127
	LTL	PLR	THA	
FixMatch				29.4
w/ FreeMatch		✓	✓	30.0
w/ SEVAL-PL		✓	✓	**31.0**
w/ SAW	✓			29.4
w/ DASO	✓	✓		29.4
w/ SEVAL	✓	✓	✓	**34.8**

C.3 Integration with Other SSL Frameworks

As an extension to results in Fig. 5, we summarize the results when introducing SEVAL into other SSL frameworks in Table 9. We summarize the implementation details of those methods in Section D.

Table 9: Accuracy on CIFAR10-LT based on SSL methods other than FixMatch. Best results within the same category are in **bold** for each configuration.

Algorithm	CIFAR10-LT $n_1 = 1500$ $m_1 = 3000$ $\gamma^{\mathcal{X}} = \gamma^{\mathcal{U}} = 100$	CIFAR100-LT $n_1 = 150$ $m_1 = 300$ $\gamma^{\mathcal{X}} = \gamma^{\mathcal{U}} = 10$
Mean Teacher (Tarvainen & Valpola, 2017)	68.6 ±0.88	52.1 ±0.09
w/ DASO (Oh et al., 2022)	70.7 ±0.59	52.5 ±0.37
w/ SEVAL	**77.6** ±0.63	**53.8** ±0.24
MixMatch (Berthelot et al., 2019b)	65.7 ±0.23	54.2 ±0.47
w/ DASO (Oh et al., 2022)	70.9 ±1.91	55.6 ±0.49
w/ SEVAL	**81.8** ±0.82	**57.8** ±0.26
ReMixMatch (Berthelot et al., 2019a)	77.0 ±0.55	61.5 ±0.57
w/ DASO (Oh et al., 2022)	80.2 ±0.68	62.1 ±0.69
w/ SEVAL	**86.7** ±0.71	**63.1** ±0.38

D Implementation Details

D.1 Learning with Imbalanced Held-Out Data

As the labelled training dataset $\mathcal{X}$ is imbalanced, in practice, it is hard to obtain a balanced split $\mathcal{V}$ to learn a curriculum of threshold $\boldsymbol{\tau}$. However, when we optimize $\boldsymbol{\tau}$ using an imbalanced validation $\mathcal{V}$ following Eq. 14, the optimized results would be biased. More precisely, the majority class consistently exhibits high **Precision**, leading to a lower threshold, while the opposite holds true for the minority class. Therefore, we utilize the class frequency of the labelled held-out data $\boldsymbol{k}$ to normalize the cost function. Specifically, we calculate the class weight as $\boldsymbol{\omega}^{\mathcal{V}} = 1/\boldsymbol{k}$. This parameter would assign large weight to the minority class and small weight to the majority classes. Then we replace all the $\mathbb{1}_{ic}$ with $\omega^{\mathcal{V}}_{y_i}\mathbb{1}_{ic}$ in Eq. 14, obtaining:

$$\tau_c^* = \begin{cases} \arg\min_{\tau_c} \left| \frac{1}{\mathcal{S}(\tau_c, c)} \sum_{i=1}^{K} \omega^{\mathcal{V}}_{y_i} \mathbb{1}_{ic} \right. & \text{if } \alpha_c < t, \\ \left. \cdot\, \mathbb{1}(y_i = c)\mathbb{1}\left(\max_j (p^{\mathcal{V}}_{ij}) > \tau_c \right) - t \right| & \\ 0 & \text{otherwise.} \end{cases} \tag{34}$$

where $\mathcal{S}(\tau_c, c) = \sum_{i=1}^{K} \omega^{\mathcal{V}}_{y_i} \mathbb{1}_{ic} \mathbb{1}(\max_j(p^{\mathcal{V}}_{ij}) > \tau_c)$ is the normalized number of samples predicted as class c with confidence larger than τ_c, where $\alpha_c = \frac{1}{K_c} \sum_{i=1}^{K} \omega^{\mathcal{V}}_{y_i} \mathbb{1}_{ic} \mathbb{1}(y_i = c)$ is the average balanced accuracy of all the samples predicted as class c and $K_c = \sum_{i=1}^{K} \omega^{\mathcal{V}}_{y_i} \mathbb{1}_{ic}$ is the normalized number of samples predicted as c. This modification can normalize the number of samples within the cost function. Consequently, we can directly learn the thresholds $\boldsymbol{\tau}$ using imbalanced held-out data.

D.2 Learning Thresholds within Groups

When we learn $\boldsymbol{\tau}$ based on the held-out data $\mathcal{V}$, the optimization process could be unstable as sometimes we have very few samples per class (e.g. less than 10 samples). In this case, even if we can re-weight the validation samples based on their class prior $\boldsymbol{k}$, it is hard to have enough samples to obtain stable $\boldsymbol{\tau}$ curriculum for the minority classes, especially when $\min_c(k_c) < 10$. Assuming equal class priors should result in similar thresholds, we propose to optimize thresholds within groups, pinpointing the ideal ones that fulfill the accuracy requirement for every classes within the group.

We assume the samples of different classes k_c are arranged in descending order. In other words, k_1 is the maximum, and k_C is the minimum. Instead of optimizing τ_c for an individual class c, we optimize for groups such that the learned $\tilde{\tau}_b$ can satisfy the accuracy requirements for B classes. Specifically, the optimal $\tilde{\boldsymbol{\tau}} \in \mathbb{R}^{\lceil C/B \rceil}$ is determined as:

$$\tilde{\tau}_b^* = \begin{cases} \arg\min_{\tilde{\tau}_b} \left| \frac{1}{\tilde{\mathcal{S}}(\tilde{\tau}_b, b)} \sum_{c=bB+1}^{bB+B} \sum_{i=1}^{K} \mathbb{1}_{ic} \right. & \text{if } \tilde{\alpha}_b < t, \\ \left. \cdot \mathbb{1}(y_i = c) \mathbb{1}(\max_j(p_{ij}^{\mathcal{V}}) > \tilde{\tau}_b) - t \right| & \\ 0 & \text{otherwise.} \end{cases}, \tag{35}$$

where $\tilde{\mathcal{S}}(\tilde{\tau}_b, b) = \sum_{c=bB+1}^{bB+B} \sum_{i=1}^{K} \mathbb{1}_{ic} \mathbb{1}(\max_j(p_{ij}^{\mathcal{V}}) > \tilde{\tau}_b)$ is the number of samples that are chosen in this group based on the threshold $\tilde{\tau}_b$ and $\tilde{\alpha}_b = \frac{1}{\sum_{c=bB+1}^{bB+B} K_c} \sum_{c=bB+1}^{bB+B} \sum_{i=1}^{K} \mathbb{1}_{ic} \mathbb{1}(y_i = c)$ is the average accuracy of all the samples predicted as class in this group. If we set $B = 1$, Eq. 35 becomes equivalent to Eq. 14.

Furthermore, in practice, we find that in imbalanced SSL settings, the minority classes sometimes have very few samples, making it difficult to optimize the thresholds correctly based on Eq. 35. In this case, we also set the learned $\tilde{\tau}_b^*$ to be 0, in order to leverage more data from the minority classes. Formally, we denote $\tilde{K}_b = \sum_{c=bB+1}^{bB+B} \sum_{i=1}^{K} \omega_{y_i}^{\mathcal{V}} \mathbb{1}_{ic}$ as the normalized number of predicted samples within group b. When $\tilde{K}_b < \sum_{i=1}^{K} \frac{B\omega_{y_i}^{\mathcal{V}}}{e_1 C}$ or $\sum_{i=1}^{K} \mathbb{1}(y_i = c) < e_2$, where e_1 and e_2 are hyper-parameters that we both set to 10 for all experiments, we also have $\tilde{\tau}_b^* = 0$ and keep their corresponding π_c within group b as low as $\pi_c = \min_j(\pi_j)$. This implies:

- In instances where the models exhibit a pronounced bias, limiting their capability to detect over 10% of the samples within a particular group, we adjust the associated thresholds and consequently increase our sample selection.
- When a group comprises fewer than 10 samples, the feasibility of optimizing thresholds based on proportion diminishes, necessitating an enhanced sample selection.

Table 10: Experiment-specific hyper-parameters. $t^{\mathcal{V}}$ is the required accuracy if we directly optimize $\boldsymbol{\tau}$ along the training process using a separate validation dataset.

Hyper-parameter	CIFAR10-LT, $\gamma^{\mathcal{X}} = 100$		CIFAR100-LT, $\gamma^{\mathcal{X}} = 10$		STL10-LT, $\gamma^{\mathcal{X}} = 20$		Semi-Aves	
	$n_1 = 500$ $m_1 = 4000$	$n1 = 1500$ $m_1 = 3000$	$n_1 = 50$ $m_1 = 400$	$n_1 = 150$ $m_1 = 300$	$n_1 = 150$ $M = 100,000$	$n_1 = 450$ $M = 100,000$	$\mathcal{U} = \mathcal{U}_{\text{in}}$	$\mathcal{U} = \mathcal{U}_{\text{in}} + \mathcal{U}_{\text{out}}$
C	10		100		10		200	
T	250,000		250,000		250,000		45,000	
t	0.75		0.5	0.65	0.7	0.6	0.9	0.99
$t^{\mathcal{V}}$	0.9		0.65	0.7	0.95	0.85	—	
L	500		100		500		90	
B	2		25	10	2	1	10	
η_π	0.999		0.95	0.9	0.995		0.99	0.9
η_τ	0.999		0.95	0.9	0.9995	0.999	0.99	0.9

Table 11: The number of classes of different performance when trained with FixMatch. We demonstrate the prevalent occurrence of the four class performance scenarios in existing semi-supervised learning tasks.

Model Performance	CIFAR10-LT $\gamma^{\mathcal{X}} = \gamma^{\mathcal{U}} = 100$		CIFAR100-LT $\gamma^{\mathcal{X}} = \gamma^{\mathcal{U}} = 10$		STL10-LT $\gamma^{\mathcal{X}} = 20$, $\gamma^{\mathcal{U}}$: *unknown*		Semi-Aves	
	$n_1 = 500$ $m_1 = 4000$	$n_1 = 1500$ $m_1 = 3000$	$n_1 = 50$ $m_1 = 400$	$n_1 = 150$ $m_1 = 300$	$n_1 = 150$ $M = 100,000$	$n_1 = 450$ $M = 100,000$	$\mathcal{U} = \mathcal{U}_{\text{in}}$	$\mathcal{U} = \mathcal{U}_{\text{in}} + \mathcal{U}_{\text{out}}$
Case 1: High **Recall** & *Low* **Precision**	4	5	24	29	2	4	36	45
Case 2: Low **Recall** & *High* **Precision**	4	4	18	20	2	2	35	45
Case 3: High **Recall** & *High* **Precision**	1	1	33	29	3	2	69	58
Case 4: Low **Recall** & *Low* **Precision**	1	0	25	22	3	2	60	52
Total classes	10	10	100	100	10	10	200	200

D.3 Benchmarks

We conduct experiments upon the code base of (Oh et al., 2022) for experiments of CIFAR10-LT, CIFAR100-LT and STL10-LT. We take some baseline results from the DASO paper (Oh et al., 2022) to Table 2, Table 4 and Table 9. including the results of supervised baselines, DARP, CReST+, ABC and DASO.

As DASO (Oh et al., 2022) does not supply the code for the Semi-Aves experiments, we conduct all the experiments for this setting ourselves. We train ResNet-50 (He et al., 2016) which is pretrained on ImageNet (Deng et al., 2009) for the task of Semi-Aves following (Su & Maji, 2021). In accordance with (Oh et al., 2022), we merge the training and held-out datasets provided by the challenge, yielding a total of 5,959 samples for training which come from 200 classes. We conduct experiments utilizing 26,640 unlabelled samples which share the same label space with $\mathcal{X}$ in the $\mathcal{U} = \mathcal{U}_{in}$ setting, and 148,848 unlabelled samples of which 122,208 are from open-set classes in the $\mathcal{U} = \mathcal{U}_{in} + \mathcal{U}_{out}$ setting. For experiments on Semi-Aves, we set the base learning rate as 0.005. We train the network for 45,000 iterations. The learning rate is linear warmed up during the first 25,00 iterations, and degrade after 15,000 and 30,000, with a factor of 10. We choose training batch size as 32. The images are firstly cropped to 256×256. During training, the images are then randomly cropped to 224×224. At inference time, the images are cropped in the center with size 224×224.

D.4 Hyper-Parameters

To facilitate reproducibility, we summarize all selected hyperparameters for our experiments in Table 10.

We find that the value of t does not significantly affect performance, and SEVAL works well across a variety of reasonable choices, as shown in Table 7. For curriculum length L, smaller values should be chosen when the number of classes C is large, since parameter optimization becomes more time-consuming with larger C. Additionally, the exponential momentum parameters (η_π and η_τ) should be adjusted accordingly. We also suggest using larger momentum values (η_π and η_τ) and a larger group size B when training data is scarce, as this helps smooth the learned parameters along the training process.

D.5 SEVAL with Other SSL Algorithms

Here, we provide implementation details of how SEVAL can be integrated into other pseudo-labeling based SSL algorithms.

Specifically, we apply SEVAL to Mean Teacher (Tarvainen & Valpola, 2017), MixMatch (Berthelot et al., 2019b) and ReMixMatch (Berthelot et al., 2019a). These algorithms produce pseudo-label $\hat{y}_i$ based on its corresponding pseudo-label probability $\boldsymbol{q}_i$ and logit $\hat{\boldsymbol{z}}_i^{\mathcal{U}}$ in different ways. SEVAL can be easily adapted by refining $\boldsymbol{q}_i$ using the learned offset $\boldsymbol{\pi}^*$.

It should be noted that these SSL algorithms do not include the process of filtering out pseudo-labels with low confidence. Therefore, for simplicity and fair comparison, we do not include the threshold adjustment into these methods. We expect that SEVAL can further enhance performance through threshold adjustment and plan to explore this further in the future.

D.5.1 Mean Teacher

Mean Teacher generates pseudo-label logit $\hat{\boldsymbol{z}}_i^{\mathcal{U}}$ based on a EMA version of the prediction models. SEVAL calculates the pseudo-label probability as $\boldsymbol{q}_i = \sigma(\hat{\boldsymbol{z}}_i^{\mathcal{U}} - \log \boldsymbol{\pi}^*)$, which is expected to have less bias towards the majority class.

D.5.2 MixMatch

MixMatch calculates $\hat{y}_i$ based on multiple transformed version of an unlabelled sample $\boldsymbol{u}_i$. SEVAL adjusts each one of them with $\boldsymbol{\pi}^*$, separately.

D.5.3 ReMixMatch

ReMixMatch proposes to refine pseudo-label probability $\boldsymbol{q}_i$ with distribution alignment to match the marginal distributions. SEVAL adjusts the the probability using $\boldsymbol{q}_i = \sigma(\hat{\boldsymbol{z}}_i^{\mathcal{U}} - \log \boldsymbol{\pi}^*)$ before ReMixMatch's process including distribution alignment and temperature sharpening.

E Classes of Different Performance

Here we summarize the number of classes of different performance, as demonstrated in Table. 11. We report the model performance of FixMatch trained on different datasets. Here we consider a class to have high **Recall** when its performance surpasses the average **Recall** of the classes. The same principle applies to **Precision**.

We observe that $Case1$ is frequently encountered in the majority class, while $Case2$ is commonly observed in the minority class. $Case3$ and $Case4$ also widely exist in different scenarios. Theoretically, SEVAL's threshold learning strategies better fit Case 3 and Case 4 compared to existing methods, consequently achieving superior performance.

References

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019a.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019b.

John Bridle, Anthony Heading, and David MacKay. Unsupervised classifiers, mutual information and'phantom targets. *Advances in neural information processing systems*, 4, 1991.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 6912–6920, 2021.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Hao Chen, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Marios Savvides, and Bhiksha Raj. An embarrassingly simple baseline for imbalanced semi-supervised learning. *arXiv preprint arXiv:2211.11086*, 2022.

Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. *arXiv preprint arXiv:2301.10921*, 2023.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Yue Fan, Dengxin Dai, Anna Kukleva, and Bernt Schiele. Cossl: Co-learning of representation and classifier for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14574–14584, 2022.

Saurabh Garg, Sivaraman Balakrishnan, Zachary C Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. *arXiv preprint arXiv:2201.04234*, 2022.

Shizhan Gong, Cheng Chen, Yuqi Gong, Nga Yan Chan, Wenao Ma, Calvin Hoi-Kwan Mak, Jill Abrigo, and Qi Dou. Diffusion model based semi-supervised learning on brain hemorrhage images for efficient midline shift quantification. In *International Conference on Information Processing in Medical Imaging*, pp. 69–81. Springer, 2023.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Lan-Zhe Guo and Yu-Feng Li. Class-imbalanced semi-supervised learning with adaptive thresholding. In *International Conference on Machine Learning*, pp. 8082–8094. PMLR, 2022.

Ju He, Adam Kortylewski, Shaokang Yang, Shuai Liu, Cheng Yang, Changhu Wang, and Alan Yuille. Rethinking re-sampling in imbalanced semi-supervised learning. *arXiv preprint arXiv:2106.00209*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *International conference on machine learning*, pp. 2731–2741. PMLR, 2019.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5070–5079, 2019.

Konstantinos Kamnitsas, Daniel Castro, Loic Le Folgoc, Ian Walker, Ryutaro Tanno, Daniel Rueckert, Ben Glocker, Antonio Criminisi, and Aditya Nori. Semi-supervised learning via compact latent space clustering. In *International conference on machine learning*, pp. 2459–2468. PMLR, 2018.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. *Advances in neural information processing systems*, 33:14567–14579, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical report*, 2009.

Zhengfeng Lai, Chao Wang, Sen-ching Cheung, and Chen-Nee Chuah. Sar: Self-adaptive refinement on pseudo labels for multiclass-imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4091–4100, 2022a.

Zhengfeng Lai, Chao Wang, Henrry Gunawan, Sen-Ching S Cheung, and Chen-Nee Chuah. Smoothed adaptive weighting for imbalanced semi-supervised learning: Improve reliability against unknown distribution data. In *International Conference on Machine Learning*, pp. 11828–11843. PMLR, 2022b.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Justin Lazarow, Kihyuk Sohn, Chen-Yu Lee, Chun-Liang Li, Zizhao Zhang, and Tomas Pfister. Unifying distribution alignment as a loss for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5644–5653, 2023.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896. Atlanta, 2013.

Hyuck Lee, Seungjae Shin, and Heeyoung Kim. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. *Advances in Neural Information Processing Systems*, 34:7082–7094, 2021.

Chongxuan Li, Taufik Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. *Advances in neural information processing systems*, 30, 2017.

Muyang Li, Runze Wu, Haoyu Liu, Jun Yu, Xun Yang, Bo Han, and Tongliang Liu. Instant: Semi-supervised learning with instance-dependent thresholds. *Advances in Neural Information Processing Systems*, 36, 2024.

Zeju Li, Konstantinos Kamnitsas, and Ben Glocker. Analyzing overfitting under class imbalance in neural networks for image segmentation. *IEEE transactions on medical imaging*, 40(3):1065–1077, 2020.

Zeju Li, Konstantinos Kamnitsas, Mobarakol Islam, Chen Chen, and Ben Glocker. Estimating model performance under domain shifts with class-specific confidence scores. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 693–703. Springer, 2022.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pp. 3122–3130. PMLR, 2018.

Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2537–2546, 2019.

Charlotte Loh, Rumen Dangovski, Shivchander Sudalairaj, Seungwook Han, Ligong Han, Leonid Karlinsky, Marin Soljacic, and Akash Srivastava. On the importance of calibration in semi-supervised learning. *arXiv preprint arXiv:2210.04783*, 2022.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

Alexander Mey and Marco Loog. Improved generalization in semi-supervised learning: A survey of theoretical results. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4747–4767, 2022.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26, 2013.

Jerzy Neyman and Egon Sharpe Pearson. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231(694-706):289–337, 1933.

Youngtaek Oh, Dong-Jin Kim, and In So Kweon. Daso: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9786–9796, 2022.

Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*, 2021.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41, 2002.

Henry Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

Jong-Chyi Su and Subhransu Maji. The semi-supervised inaturalist-aves challenge at fgvc7 workshop, 2021.

Jong-Chyi Su, Zezhou Cheng, and Subhransu Maji. A realistic evaluation of semi-supervised learning for fine-grained classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12966–12975, 2021.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Junjiao Tian, Yen-Cheng Liu, Nathaniel Glaser, Yen-Chang Hsu, and Zsolt Kira. Posterior re-calibration for imbalanced datasets. *Advances in Neural Information Processing Systems*, 33:8101–8113, 2020.

Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.

Slobodan Vucetic and Zoran Obradovic. Classification on data with biased class distribution. In *Machine Learning: ECML 2001: 12th European Conference on Machine Learning Freiburg, Germany, September 5–7, 2001 Proceedings 12*, pp. 527–538. Springer, 2001.

Renzhen Wang, Xixi Jia, Quanziang Wang, Yichen Wu, and Deyu Meng. Imbalanced semi-supervised learning with bias adaptive classifier. *arXiv preprint arXiv:2207.13856*, 2022a.

Xudong Wang, Zhirong Wu, Long Lian, and Stella X Yu. Debiased learning from naturally imbalanced pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14647–14657, 2022b.

Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, et al. Usb: A unified semi-supervised learning benchmark for classification. *Advances in Neural Information Processing Systems*, 35:3938–3961, 2022c.

Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, et al. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv preprint arXiv:2205.07246*, 2022d.

Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10857–10866, 2021.

Tong Wei and Kai Gan. Towards realistic long-tailed semi-supervised learning: Consistency is all you need. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3469–3478, 2023.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268, 2020.

Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In *International Conference on Machine Learning*, pp. 11525–11536. PMLR, 2021.

Zhuoran Yu, Yin Li, and Yong Jae Lee. Inpl: Pseudo-labeling the inliers first for imbalanced semi-supervised learning. *arXiv preprint arXiv:2303.07269*, 2023.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021.

Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16, 2003.

Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on knowledge and data engineering*, 18(1):63–77, 2005.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.