Grouping and Transporting Enable Robust Thresholding for Semi-supervised Learning

Anonymous authors
Paper under double-blind review

Abstract

Semi-supervised learning (SSL) digs unlabeled data through pseudo-labeling when labeled data is limited. Despite various auxiliary strategies to enhance SSL training, the main challenge lies in how to determine reliable pseudo labels with a robust thresholding algorithm based on quality indicators (*e.g.*, confidence scores). However, the latest methods for distinguishing low or high-quality labels require complex-designed thresholding strategies but still fail to guarantee robust and efficient selection. Empirically, we group the quality indicators of pseudo labels into three clusters (easy, semi-hard, and hard) and statistically reveal the real bottleneck of threshold selection lying in the sensitivity of separating semi-hard samples. To this end, we propose an adaptive **G**rouping and **T**ransporting for **R**obust thresholding (dubbed as GTR) that efficiently selects semi-hard samples with test-time augmentations and consistency constraints while saving the selection budgets of easy and hard samples. Our proposed GTR can effectively determine high-quality data when applied to existing SSL methods while reducing redundant selection costs. Extensive experiments on eleven SSL benchmarks across three modalities verify that GTR achieves significant performance gains and speedups over Pseudo Label, FixMatch, and FlexMatch.

1 Introduction

Over the past decades, deep learning (DL) has made significant strides across diverse applications and modalities (He et al., 2016; Devlin et al., 2018; Dong et al., 2018). However, the majority of tasks operate under supervised learning (SL), which necessitates manual data labeling that is constrained by limited quantity and resource-intensive efforts. To overcome these limitations and leverage extensive unlabeled data, semi-supervised learning (SSL) has emerged as a promising solution. Holistically, SSL exploits information from both unlabeled and limited labeled data (Tarvainen & Valpola, 2017; Sohn et al., 2020) within the self-training paradigm of pseudo-labeling (Lee et al., 2013), where models are designed to be trained using unlabeled data and pseudo-labels assigned by their own predictions.

As SSL continues to develop, a crucial avenue for advancing mainstream methods lies in establishing a well-designed selection method (Zhang et al., 2021) or a robust quality indicator (Li et al., 2024) for more accurate pseudo label selection. Existing approaches predominantly rely on threshold-based pseudo-labeling strategies (Sohn et al., 2020; Kim et al., 2022) based on confidence scores (Lee et al., 2013), designing refined class-wise thresholding schemes (Wang et al., 2022b) or dynamic thresholding policies throughout the whole training process (Zhang et al., 2021). However, these thresholding methods, with their complex thresholding values or schedules, are still linear classification algorithms to separate whether the pseudo labels are reliable and thereby exhibit instability, which requires substantial manual intervention but fail to leverage the inherent distributions of indicators. Taking FlexMatch (Zhang et al., 2021) as an example, the density estimation in Figure 1a demonstrates that training leads to instability and a lack of distinct class differentiation. The overlapping confidence distributions also indicate the model's struggle to distinguish between classes both before and after training clearly. Recent methods such as FreeMatch (Wang et al., 2022b) and SoftMatch (Chen et al., 2022b) also face similar challenges. These methods focus on sample level but employ a simple mean threshold that only captures the inter-class properties of labels, making them sensitive to threshold variations and thus leading to instability.

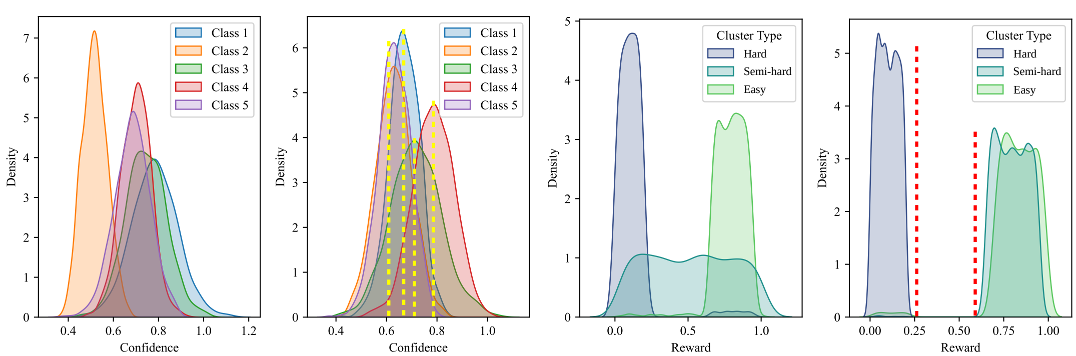
Table 1: Characteristics of the pseudo-label selection process, comparing typical SSL algorithms and the proposed GTR. The compared characteristics or strategies include Robust τ (the thresholding guarantees robustness or not), Speedup (boosting the convergence or not), Gain (improving performance or not), and Thresholding (the method of filtering pseudo labels). G&T denotes the proposed Grouping and Transporting as a robust thresholding way.

Method	Pseudo Labeling	FixMatch	FlexMatch	FreeMatch	SemiReward	**GTR**
Robust τ	✗	✗	✗	✗	✗	✓
Speedup	✗	✓	✗	✓	✓	✓
Gain	✗	✗	✓	✓	✓	✓
Thresholding	None	Hard	Dynamic	Adaptive	Mean	G&T

Our study addresses these challenges at once by constructing a robust thresholding mechanism, termed **G**rouping and **T**ransporting **R**obust thresholding (**GTR**), tailored for SSL. Unlike traditional methods that solely rely on inter-class separation, our GTR leverages the inherent properties of the indicator distribution through unsupervised clustering. As shown in Figure 1b, GTR mitigates the threshold sensitivity by focusing on the intra-class properties, particularly in those semi-hard groups. This innovative grouping design enables effective pseudo-label selection, enhanced by the transportation method, which refines the indicator distribution. Table 1 compares existing schemes and their characteristics, finding Grouping and Transporting mechanism in GTR ensures effective pseudo-label thresholding, leading to improved convergence speed and performance gains, setting it apart as a superior approach for SSL tasks. We further conduct a detailed analysis with grouping to gain an in-depth understanding of the intrinsic characteristics of the entire SSL training pipeline from a data perspective.

Empirical research and statistical analysis show that the proposed GTR can accelerate model training and achieve excellent results with fast convergence and no extra computations. Based on the popular USB benchmarks (Wang et al., 2022a), we selected representative SSL methods to conduct comparative experiments for verifying the versatility and robustness of our GTR method. Our main contributions are threefold:

- We empirically reveal that the impediment of existing thresholding techniques lies in their inability to separate the semi-hard group of the indicator when selecting high-quality pseudo labels. This insight highlights the need for a specially designed method to address the issue.
- We design a transporting method tailored for three groups of samples: easy, semi-hard, and hard. By employing kernel density estimation, we analyze the SSL training pipeline and leverage the inherent nature of indicator distribution to elucidate how our method promotes the semi-hard group towards a better-optimized distribution, such as that of the easy group.



(a) Confidence distributions by class (FlexMatch) (b) Reward score distributions (GTR)

Figure 1: Distribution of pseudo-label indicators and selection boundaries on CIFAR-100 (400 labels). (a) In FlexMatch, confidence score distributions show slight changes before and after training, with separation boundaries (yellow lines) located at density peaks, making it difficult to distinguish classes effectively. (b) In **GTR**, leveraging intra-class properties for pseudo-label selection, separation boundaries are placed at low-density regions. The grouping of three types of samples (red lines) captures essential label characteristics. Combining grouping with transporting significantly enhances distribution separability, addressing the instability issues seen in existing methods.

- We seamlessly integrate GTR into existing SSL algorithms without incurring any additional overhead. Extensive experiments across eleven SSL benchmarks further validate the reliability and effectiveness of GTR, showcasing its applicability over diverse SSL modalities.

2 Problem Definition

Notations. Semi-Supervised Learning (SSL) extends Supervised Learning (SL) by using a small labeled dataset $\mathcal{D}_L = \{(x_i^l, y_i^l)\}_{i=1}^{N_L}$ and a large unlabeled dataset $\mathcal{D}_U = \{x_i^u\}_{i=1}^{N_U}$ with $N_L \ll N_U$. For a given classification task, the model prediction $f_S(x) = y \in \mathbb{R}^C$, where C is the label dimension. The SSL training involves three processes: (i) **pseudo-label generation** produces pseudo labels $y^u = f_T(x^u)$ by a trained teacher model f_T on $\mathcal{D}_L$ and converts them to one-hot encoding; (ii) **pseudo-label filtering** selects high-quality pseudo-labels $\hat{y}^u$ using a pseudo-label quality indicator $\mathcal{I}(\cdot)$ and thresholds, *e.g.,* $\hat{y}^u = \mathcal{I}(y^u) > \tau$ with a single threshold τ; (iii) **learning objectives** are computed by the sum of supervised and unsupervised losses, $\mathcal{L} = \mathcal{L}_s + \mathcal{L}_u$.

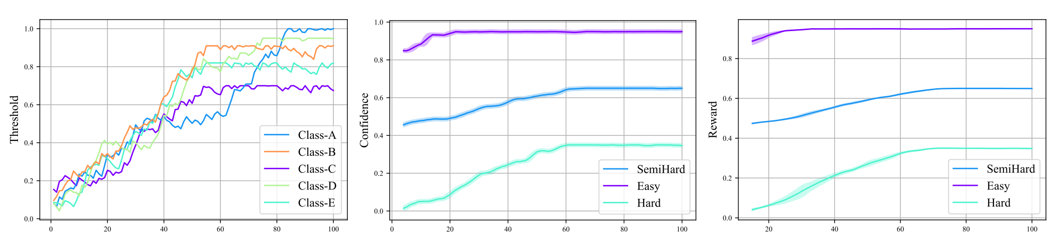
$$\mathcal{L}_S = \frac{1}{B_L} \sum_{i=1}^{B_L} \mathcal{H}\Big(y_i^l, f_S\big(\omega(x_i)\big)\Big), \tag{1}$$

where $\omega(\cdot)$ denotes weak data augmentations, and $\mathcal{H}(\cdot, \cdot)$ is the loss function for SL tasks (e.g., cross-entropy, ℓ_1 loss). For a mini-batch of B_U unlabeled data, the unsupervised loss is:

$$\mathcal{L}_U = \frac{1}{B_U} \sum_{i=1}^{B_U} \mathbb{I}(p_i^u, \tau)\mathcal{H}\Big(\hat{y}_i^u, f_S\big(\Omega(x_i^u)\big)\Big), \tag{2}$$

where $\Omega(x_i^u)$ denotes strong augmentations. Consistency regularization typically involves updating f_S parameters to f_T via copying or exponential moving average (EMA) and requires predicted classification confidence to identify reliable labels.

The Devil Lies in Thresholding. In SSL frameworks, the pseudo-label filtering process is the most crucial part (Arazo et al., 2020; Zhang et al., 2021), which can be regarded as a binary classification task: *a thresholding algorithm predicts whether the pseudo label y^u is reliable (as positive) or inaccurate (as negative) according to the quality indicator $\mathcal{I}(y^u)$.* With two widely employed indicators (confidence scores (Lee et al., 2013; Xie et al., 2020a) and reward scores (Li et al., 2024)), existing SSL methods designed numerous thresholding strategies. However, no matter how adaptive or fine-grained thresholds are adopted (Wang et al., 2022b), existing thresholding algorithms are equal to linear classifiers and neglect the intrinsic binary distributions of distinguishing two types of pseudo labels. As shown in Figure 1a (right), it is difficult to separate the Gaussian-like indicator distributions by linear decision boundaries at the densest locations (*i.e.*, the yellow lines), which will cause instability filtering issues in the existing thresholding methods with class confidences shown in Figure 2a. To reveal the cause of instabilities, we first cluster the indicator distributions into three consistent groups by a clustering algorithm (Reynolds et al., 2009) to investigate the properties of the thresholding task. As indicated in Figure 1b (left) or 4a, we found that both the indicator values of unreliable and reliable pseudo labels are clustered into two distinct distributions (dubbed as hard and easy groups), while the middle group (dubbed as semi-hard) is similar to both the hard and



(a) Thresholding by Confidence (b) Grouping by Confidence (c) Grouping by Reward Scores

Figure 2: Pseudo-label selection with 100-epoch training on CIFAR-100 (400 labels) with FixMatch. (a) Changing trend of confidence threshold for each class of five randomly selected classes. (b) The variation trend of mean and variance statistics for three groups clustered on the confidence scores. (c) The variation trend of mean and variance statistics of three groups clustered on the reward indicators.

easy groups. The semi-hard distribution nearly corresponds to the dense region of original indicator distributions, which can be hard to separate and cause instabilities during the entire SSL training as shown in Figure 2.

3 Robust Grouping and Thresholding for Unlabeled Data

To address the instability and poor class differentiation discussed in Section 2, we introduce GTR, which employs robust thresholding through grouping and transporting. Unlike traditional methods that use simple linear thresholds, GTR clusters pseudo-labels into distinct groups, effectively filtering high-quality labels. This approach mitigates the instability caused by overlapping indicator distributions, ensuring more accurate and stable pseudo-label selection and improving SSL task performance.

3.1 Grouping: Indicator-based Property Mining

Within a single epoch, each unlabeled sample is selected by an evaluation criterion, such as quality indicators like confidence scores. We employ the unsupervised Gaussian Mixture Model (GMM) (Reynolds et al., 2009) to divide samples into three clusters and calculate related statistics (μ, σ), resulting in the distribution of three types of samples: $\mathcal{D}_U = \left\{\mathcal{X}_\alpha^u, \mathcal{X}_\beta^u, \mathcal{X}_\gamma^u\right\}$, corresponding to easy, semi-hard, and hard groups, respectively. The size of each group in a mini-batch is denoted as A, B, Γ. We define the probability of each data point belonging to each cluster as $P_\alpha(x_i|\theta_\alpha)$, $P_\beta(x_i|\theta_\beta)$, $P_\gamma(x_i|\theta_\gamma)$. In this probability distribution, each data point has associated probabilities of belonging to the easy, semi-hard, and hard groups, summing up to 1. Thus, we accomplish sample-level grouping. The choice of the GMM method is due to its effectiveness in forming non-spherical clusters with ambiguous points, allowing better modeling of elongated clusters. As shown in Figure 2, compared to class-level grouping, the variations among groups obtained through this method are relatively stable and align with the intuition of modeling the label space, which typically involves both intra-class and inter-class modeling. Figure 1a illustrates that class-level grouping mainly considers inter-class attributes, reflecting only part of the properties. Different samples within the same class can have varying difficulty levels, leading to more uncertainty during thresholding. Whether using a hard, class-level, or adaptive threshold, traditional methods essentially separate labels below a threshold under limited modeling. The grouping method avoids this rigid thresholding and includes the nature of intra-class properties, making the preparation for thresholding more comprehensive. Meanwhile, using more robust indicators like a reward score $r_i = R(x_i^u, y_i^p)$ (Li et al., 2024) further enhances the stability in Figure 2c.

3.2 Transporting: Promoting Semi-hard to Esay

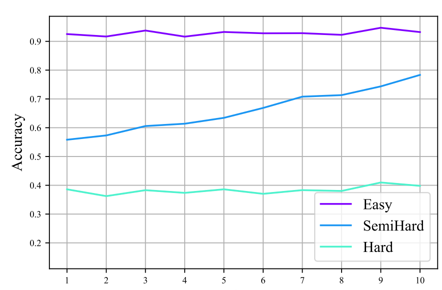


Figure 3: The average quality indicator for each group is calculated on CIFAR-100 (400 labels) after grouping the unlabeled data. The number of filters applied and resulting changes in the quality indicator are mapped out. Thresholds are set as the mean for each group. After filtering, samples are scored and re-grouped.

Building upon the foundation of the grouping method, we further contemplate how to utilize the properties from the label space to achieve more robust processing. Hence, we introduce the transporting method. As shown in Figure 3, grouping can capture the intrinsic properties of indicator distributions, reflecting that the semi-hard group is sensitive to thresholds during SSL training, and easy/hard groups are robust and determined. Statistically, we also introduced the Pearson correlation coefficient (Cohen et al., 2009) to derive the characteristics of each group further. First, we collected the accuracy data corresponding to the three groups after different filtering times as follows:

$$r = \frac{n(\sum XY) - (\sum X)(\sum Y)}{\sqrt{[n\sum X^2 - (\sum X)^2][n\sum Y^2 - (\sum Y)^2]}}, \tag{3}$$

where n is the maximum number of filterings, X represents the vector $X = \{1, 2, 3 \ldots, 9, 10\}$ corresponding to the filtering number array. Y is the accuracy rate of each group after the corresponding filtering times. The result calculation can be obtained as $P_\alpha = 0.189$, $P_\beta = 1.415 \times 10^{-7}$, $P_\gamma = 0.067$. Observing P_β, a notable $P < 0.01$ is evident within the semi-hard group, underscoring a pronounced association between semi-hard samples and the thresholding frequency. This implies the sensitivity of semi-hard samples to filtering. Consequently, the advanced threshold design is a great way to improve the SSL method.

Transporting leverages the intrinsic properties of pseudo-label indicators. Our approach consists of three steps: **(i) Accepting easy samples:** Easy samples are likely to produce high-quality pseudo-labels, which we use to compute $\mathcal{L}_U$. **(ii) Addressing semi-hard samples:** We aim to align the distribution of semi-hard samples with that of easy samples during the transporting step. This group exhibits high sensitivity, fluctuating between high and low quality with input variations. To address this, we propose multiple selection and consistency constraints to reduce uncertainty and enhance pseudo-label accuracy. By leveraging Test-Time Augmentation (TTA) (Shanmugam et al., 2021), we generate multiple augmented samples for the student model and select pseudo-labels above a certain threshold for $\mathcal{L}_U$. The augmented samples also serve as regularization. Using the highest-scoring pseudo-label as the target, we compute a consistency loss to align all augmented data to the distribution of high-quality samples. This method extracts high-quality pseudo-labels, enhancing the efficiency and robustness of semi-hard samples. For TTA, we randomly apply horizontal and vertical flipping. **(iii) Addressing hard samples:** In each iteration, we discard half of the pseudo-labels in this group. Using the mean indicator score of the hard group as a threshold, we retain samples above the threshold and transfer them to the semi-hard group for the next iteration.

Overall, the final equation of unlabeled loss is written as:

$$\begin{aligned}\mathcal{L}_U = \frac{1}{\mathrm{B_U}} \sum_{i=1}^{\mathrm{B_U}} \mathbb{I}(q_i^u, \tau_\beta, \tau_\gamma) \mathcal{H}\Big(\hat{y}_i^u, f_S\big(\Omega(x_i^u)\big)\Big) + \\ \frac{1}{\mathrm{B}} \sum_{i=1}^{\mathrm{B}} \mathcal{H}\Big(\hat{y}_i^{u,\beta}, f_S\big(\mathrm{T}(x_i^{u,\beta})\big)\Big),\end{aligned} \tag{4}$$

where B denotes the size of semi-hard group in mini-batch, and T represents TTA. Also, q_i^u is the quality indicator corresponding to each unlabeled sample, τ_β is the filtering threshold of the semi-hard group, whose value is $(\bar{\mathcal{X}}_\alpha^u + \bar{\mathcal{X}}_\beta^u)/2$, τ_γ is threshold of hard group equivalent to $\bar{\mathcal{X}}_\gamma^u$.

3.3 Essential Characteristics of SSL Training

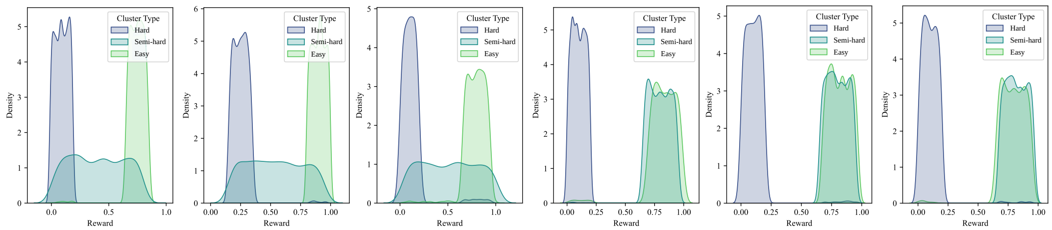
As mentioned in Sec. 2, most SSL methods focus on constructing appropriate quality indicators (metrics) and designing methods based on these indicators. Previous research has established suitable indicators but lacks an analysis from the perspective of the entire SSL training process. Meanwhile, it is essential to explore the related properties of the grouping and transporting pipeline to ensure reliability and robustness. To accurately map input samples to the label space, it is essential to use appropriate methods for identifying intrinsic properties for effective thresholding. In the process of empirical experiments, we find the label space distribution is typically elongated. Grouping methods, such as GMM, can identify these properties. We use a GMM to group pseudo-labels by quality indicators $\mathbf{z} \in \mathbb{R}^d$:

$$p(\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{z}|\mu_k, \Sigma_k), \tag{5}$$

where K is the number of components, π_k is the mixture weight, μ_k and Σ_k denote the mean and the covariance matrix. Parameters are estimated via EM algorithm (MacQueen et al., 1967). The Mahalanobis distance $d_M(\mathbf{z}_i, \mu_{k_i})$ assesses pseudo-label fit to find high distances indicate lower reliability, which guides thresholding decisions:

$$d_M(\mathbf{z}_i, \mu_{k_i}) = \sqrt{(\mathbf{z}_i - \mu_{k_i})^T \Sigma_{k_i}^{-1} (\mathbf{z}_i - \mu_{k_i})}. \tag{6}$$

In our training pipeline, the key issue is to monitor the changes in the indicator distributions. Without performing transporting, although the overall quality indicator trend is upward, the changes in the semi-hard group are negligible, as shown in Figure 4a. Since SSL training is a process from easy to hard, there inevitably exists uncertainty in the student model in the early stages. Previous



(a) Without Transporting (b) With Transporting (c) Indicator Change by Epochs

Figure 4: Illustration of the sample pseudo-label quality indicator kernel density estimation and compares the difference in the sample pseudo-label quality indicator kernel density distribution obtained before and after training. The abscissa denotes the reward score, which is the indicator we selected, and the ordinate is the density distribution of the quality indicator for each sample after kernel density estimation. (a) Before and after distribution without transporting. (b) The figure on the left is the result before transporting, and the figure on the right is the result after transporting. (c) When $t > T$, Changes are distributed in two adjacent epochs.

methods attribute these changes to inter-class sample properties and ignore the presence of key samples. Therefore, they may not effectively capture the subtle differences required for performance improvement. In contrast, GTR can model the intra-class distribution through grouping, associate relevant features, and fully utilize transporting for targeted processing.

After transporting, the semi-hard group's distribution is closely aligned with the easy group throughout training, as shown in Figure 4b. Clustering results are updated at each epoch with a step size defined as t. Notably, for $t < T$, transporting eliminates subpar semi-hard samples without adding them in training. As the semi-hard samples are aligned to the easy distribution, some samples may lie on the cluster boundary, stabilizing distributions via transporting. After $t-1$, semi-hard and easy samples may be re-grouped randomly at t, akin to merging the groups. When $t > T$, the previous process can lead to convergence to a more stable state, reflecting the advantages of transporting. The process is formulated as follows:

$$\begin{aligned} &\mathcal{X}^u_{\alpha,t-1} \rightarrow \mathcal{X}^u_{\alpha,t}\mathbb{I}\left(\mathbf{r}^u_i > \tau_{\beta,t-1}\right), \\ &\mathcal{X}^u_{\beta,t-1} \rightarrow \mathcal{X}^u_{\beta,t}, \end{aligned} \tag{7}$$

when the process reaches the next epoch, we will have:

$$\begin{aligned} &\mathcal{X}^u_{\alpha,t} \rightarrow \mathcal{X}^u_{\alpha,t+1}(x^u_i \in \mathcal{X}^u_{\beta,t})\mathbb{I}\left(\mathbf{r}_i > \tau_{\beta,t}\right), \\ &\mathcal{X}^u_{\beta,t} \rightarrow \mathcal{X}^u_{\beta,t+1}(x^u_i \in \mathcal{X}^u_{\alpha,t}), \end{aligned} \tag{8}$$

where $\tau_{\beta,t}$ is defined in Eq. 4, t refers to the epoch for training. After training N epochs, when the model is nearly converged, we will have:

$$\mathcal{X}^u_{\alpha,t+N} \sim \mathcal{X}^u_{\beta,t+N}, \bar{\mathcal{X}}^u_{\alpha,t+N} \approx \bar{\mathcal{X}}^u_{\beta,t+N}, \tag{9}$$

which signifies a favorable convergence condition. Therefore, as distinctly illustrated in Figure 4c, we found that the quality indicator distributions of the two groups in two consecutive epochs closely mirror each other after convergence.

4 Experiments

4.1 Experimental Setup

Comparison Methods for Tasks. To unveil the efficiency of GTR, we conduct a comprehensive comparison with mainstream SSL algorithms, including FlexMatch, FixMatch, and Pseudo Label (Lee et al., 2013; Arazo et al., 2020), which establish performance baselines. The essential differences between these methods are explained in Table 1. Our evaluation initially focuses on assessing the algorithms' performance regarding classification error rate and training convergence speed, undertaking a two-fold comparison. Firstly, we introduce FlexMatch and Pseudo Label as baselines, SemiReward as one of the comparison objects, and then use GTR based on the reward indicator as our method for comparative analysis. Secondly, when confidence scores or reward scores served as the indicator, we introduce confidence-based and reward-based GTR for further analysis.

Table 2: Top-1 error rate (%), performance gain (%), and training speedup times on nine classification datasets across CV, NLP, and Audio modalities in various label settings. R.GTR denotes GTR with the reward indicator, and its gains and speedup times are calculated upon baselines (Base).

Domain	Dataset (Setting)	Pseudo Label			FlexMatch			Average	
		Base	+SR	**R.GTR**	Base	+SR	**R.GTR**	Gain	Speed.
Audio	ESC-50 (250)	38.42±0.85	33.33±0.97	**32.12**±0.19	36.83±0.51	32.58±0.51	**30.11**±1.04	**+6.51**	×**2.62**
	ESC-50 (500)	28.92±0.24	27.65±0.32	**26.91**±0.61	27.75±0.41	25.92±0.31	**25.11**±0.21	**+2.33**	×**2.46**
	FSDnoisy18k (1773)	34.60±0.55	33.24±0.82	**31.10**±0.88	26.29±0.17	25.63±0.28	**25.10**±0.18	**+2.35**	×**1.39**
	UrbanSound8k (100)	37.74±0.96	36.47±0.65	**36.11**±0.32	37.88±0.46	36.06±0.93	**35.17**±0.92	**+2.17**	×**3.13**
	UrbanSound8k (400)	27.45±0.96	25.27±0.65	**24.01**±0.71	23.78±0.46	23.45±0.93	**21.02**±0.54	**+3.10**	×**1.37**
NLP	AG News (40)	13.89±0.11	12.63±0.21	**11.32**±0.52	11.11±1.19	10.60±0.69	**10.23**±0.70	**+1.73**	×**5.09**
	AG News (200)	13.10±0.39	12.10±0.58	**11.24**±0.51	13.27±0.13	11.05±0.14	**10.11**±0.29	**+2.15**	×**2.64**
	Yahoo! Answer (500)	34.87±0.50	35.08±0.40	**33.41**±0.51	34.73±0.09	33.64±0.73	**31.03**±0.61	**+2.58**	×**1.53**
	Yahoo! Answer (2000)	33.14±0.70	32.50±0.42	**31.33**±0.18	31.06±0.32	29.97±0.10	**29.21**±0.09	**+1.83**	×**6.41**
	Yelp Review (250)	46.09±0.15	42.99±0.14	**42.43**±0.66	46.09±0.15	42.76±0.33	**42.32**±0.44	**+3.72**	×**1.31**
	Yelp Review (1000)	44.06±0.14	42.08±0.15	**38.96**±0.64	40.38±0.33	37.58±0.19	**36.21**±0.34	**+4.64**	×**1.47**
CV	CIFAR-100 (200)	32.78±0.20	31.94±0.57	**30.17**±0.27	25.72±0.35	23.74±1.39	**22.61**±0.97	**+2.86**	×**1.27**
	CIFAR-100 (400)	25.16±0.67	23.84±0.20	**21.41**±0.52	17.80±0.57	17.59±0.35	**16.03**±0.36	**+2.76**	×**1.29**
	STL-10 (40)	20.53±0.12	17.37±0.47	**16.31**±0.95	11.82±0.51	10.20±1.11	**9.83**±0.52	**+3.11**	×**1.82**
	STL-10 (100)	11.25±0.81	10.88±1.48	**9.05**±0.27	7.13±0.20	7.59±0.57	**7.02**±0.69	**+1.16**	×**2.73**
	Euro-SAT (20)	25.25±0.72	23.65±0.41	**22.11**±0.52	5.54±0.16	4.86±1.00	**4.09**±0.43	**+2.30**	×**1.64**
	Euro-SAT (40)	12.82±0.81	8.33±0.33	**7.69**±0.82	4.51±0.24	3.88±0.69	**3.69**±0.32	**+2.98**	×**1.52**
	Semi Aves 3959 (3959)	40.35±0.30	37.93±0.45	**37.15**±0.76	32.48±0.15	31.23±0.09	**30.75**±0.41	**+2.47**	×**2.21**

Table 3: Top-1 error rate (%), performance gain (%), and training speedup times on SSL classification datasets with CV in various label settings under FixMatch. C.GTR refers to confidence indicator-based GTR, while R.GTR denotes reward indicator-based GTR. Performance gain and speedup times for R.GTR are compared to the baseline (Base).

Dataset (Setting)	FixMatch				Average	
	Base	**+C.GTR**	+SR	**+R.GTR**	Gain	Speed.
CIFAR-100 (200)	29.6±0.90	**28.72**±2.44	28.42±0.56	**26.14**±1.09	**+3.46**	×**2.12**
CIFAR-100 (400)	19.56±0.52	**19.04**±0.10	18.21±0.25	**17.79**±0.55	**+1.77**	×**1.67**
STL-10 (40)	16.15±1.89	**14.97**±1.07	12.92±0.71	**11.80**±0.74	**+4.35**	×**1.98**
STL-10 (100)	8.11±0.68	**7.68**±0.48	7.72±0.41	**7.22**±0.46	**+0.89**	×**1.51**
Euro-SAT (20)	13.44±3.53	**11.56**±0.21	10.69±0.26	**9.36**±0.80	**+4.08**	×**1.93**
Euro-SAT (40)	5.91±2.02	**5.13**±0.28	4.91±0.17	**4.35**±0.57	**+1.56**	×**2.13**

Task Configurations. Our experiments cover eleven SSL datasets across three popular modalities, each with specific settings outlined below. Details of datasets and experiment configurations are provided in Appendix A.1.

(i) For CV tasks, we investigate challenging datasets including CIFAR-100 (Krizhevsky et al., 2009), STL-10 (Coates et al., 2011), EuroSAT (Helber et al., 2019), and ImageNet (Deng et al., 2009). The backbone architectures used were the ImageNet pre-trained Vision Transformers (ViT) (Dosovitskiy et al., 2021) or randomly initialized ResNet-50 (He et al., 2016).

(ii) In NLP, we consider three datasets: AG News (Zhang et al., 2015), Yahoo! Answers (Chang et al., 2008), and Yelp Review (yel, 2014). The backbone encoder for these tasks is the self-supervised pre-trained BERT (Devlin et al., 2018).

(iii) In audio tasks, our study covers three datasets: UrbanSound8k (Salamon et al., 2014), ESC-50 (Piczak, 2015), and FSDNoisy18k (Fonseca et al., 2019). The pre-trained backbone adopts HuBERT (Hsu et al., 2021).

Implementations. GTR does not require tunable hyperparameters except for using GMM for the grouping step, which follows the default setting given by (Reynolds et al., 2009). As for the quality indicators of confidence and reward scores in the baselines, we follow the official hyper-parameters and training settings in FixMatch and SemiReward. More specific training and hyperparameter settings are provided in Appendix A.2.

4.2 Comparison Results on Semi-supervised Benchmarks

Table 2 illustrates the significant performance improvements achieved by integrating reward indicator-based GTR with two representative SSL algorithms, significantly improving training efficiency and final performance. Notably, GTR exhibits an average performance gain of **6.51%** on ESC-50 with 250 labels. Relative to SemiReward, GTR also performs well on fine-grained data sets. The GTR method further promotes the convergence of the model training process, as can be seen from the reduction in training time, as detailed in Appendix B. Table 3 illustrates that GTR based on confidence continues to exhibit a positive impact on model convergence. Using FixMatch as the baseline, we conducted comparisons by introducing SemiReward and employing confidence indicator-based GTR and reward indicator-based GTR to highlight their respective effects. Notably, GTR based on confidence, as discussed in Sec. 3.1, exhibits a smooth grouping strategy with a commendable promotional effect. On CIFAR-100, confidence indicator-based GTR achieves a comparable effect to SemiReward but with lower overhead, omitting additional gradient calculations. In contrast, reward indicator-based GTR incurs no extra overhead while reducing the number of student model forwards. Our approach thus achieves improved convergence and acceleration outcomes efficiently and robustly. Sec. 3.3 has explained such results and further demonstrated the superiority of GTR through these experiments.

Table 4: Top-1 error rate, performance gain, and training speedup times on ImageNet with 100 labels per class. GTR utilizes reward scores.

Method	Top-1 (%)	Gain (%)	Speedup
FixMatch	43.66	+0.00	×1.00
FixMatch+SR	41.72	+1.94	×1.98
FixMatch+GTR	**41.12**	**+2.54**	**×2.58**
FlexMatch	41.85	+0.00	×0.00
FreeMatch	40.57	+1.28	×1.50
SoftMatch	40.52	+1.33	×1.46
FlexMatch+GTR	**39.72**	**+1.49**	**×2.95**

Moreover, on the large-scale SSL benchmark ImageNet, as shown in Table 4, GTR noticeably reduces training time and achieves lower error rates, *e.g.,* FlexMatch+GTR outperforms previous SOTA methods Freematch and Softmatch. The basic method FixMatch also significantly benefits from combining with GTR and outperforms FixMatch simply combined with SemiReward.

4.3 Analysis and Ablation

This section provides an empirical analysis of the proposed modules, verifies their functionalities, and examines the key issues in the SSL training process, evaluating the impact of the proposed GTR.

Resource-Friendly SSL Training. Existing SSL training pipelines, like in SemiReward, require multiple forwards of the student model to generate pseudo-label candidates (*e.g.*, 6 times), leading to increased resource consumption in each iteration. GTR can dramatically optimizes the training process. Assuming k student model forwards per batch and denoting the proportions of easy, semi-hard, and hard samples as α, β, γ, respectively, easy and hard samples do not need multiple forwards, while semi-hard samples only need one additional forward with TTA. Thus, the total forwards per batch reduce to 2 while the computational cost of re-grouping after each epoch is also negliectable.

Confirmation of Group Filtering Thresholds. As described in Sec. 3.2, we screened samples from the semi-hard and hard groups during training. The hard group is less sensitive to filtering than the semi-hard group, but it still impacts training due to clustering updates each epoch. For semi-hard samples, we aim to align their distribution with the easy group, using the mean of both groups as an indicator. To test this, we conducted ablation experiments. Table 5 shows results for different thresholds: τ_1 (average of means of easy and semi-hard groups), τ_2 (geometric mean of means), and τ_3 (mean within semi-hard group). For the hard group, we evaluated the training impact. The results show that using the geometric mean as the threshold increases time cost, likely due to first-order distance separability. Notably, using the mean within the group slows convergence and reduces accuracy. Hard sample screening does not significantly affect final performance but does influence convergence speed.

Table 5: Ablation experiments for two groups of thresholds on Semihard and Hard groups with FlexMatch on CIFAR-100 (400 labels). Top-1 accuracy (%) and training iterations are reported.

Group	SemiHard	Acc.	Iterations
Threshold	τ_1	**84.01**	**108544 iters**
	τ_2	83.95	165888 iters
	τ_3	83.21	139263 iters
Group	**Hard**	**Acc.**	**Iterations**
Threshold	✓	**84.01**	**108544 iters**
	✗	83.82	165888 iters

Selection of Clustering Methods and Grouping Numbers. As discussed in Sec. 3.1, we use GMM due to the linear distribution of our clustered data, which enables non-spherical clusters and handles fuzzy points better. We also tested alternative unsupervised methods for a clearer illustration. Figure 5 shows the indicator data distribution on CIFAR-100, highlighting that GMM effectively models the flat and narrow distribution, which is difficult for other methods. Experiments on CIFAR-100 with 400 labels further validate the necessity of GMM. Table 6 shows these results, with GMM achieving the highest accuracy (84.01%), demonstrating its effectiveness in capturing the probability distribution of such data and confirming it as the most suitable unsupervised method. Moreover, we conduct further analysis of the number of groups. Since the semi-hard labels are likely to become easy with further training, they help improve label quality progressively. As shown in Table 7, using only two groups (easy and hard) would result in high misclassification at the decision boundary, destabilizing training, while more than three groups introduce unnecessary complexity without more performance gains.

Table 6: Ablation of various clustering methods for the Grouping step on CIFAR-100 (400 labels). The classification accuracy (%) and the total training iterations are reported. HC denotes Hierarchical Clustering.

Types	Acc.	Iterations
GMM Kambhatla & Leen (1994)	**84.01**	**108544 iters**
K-means MacQueen et al. (1967)	83.25	139263 iters
HC Eppstein (2000)	83.21	145408 iters
DBSCAN Ester et al. (1996)	82.31	77824 iters

Table 7: Error (%) for different group numbers. Setting on Flexmatch with GTR using the reward indicator.

Group Number	Error
3 group	16.03$\pm$0.36
2 group	17.64$\pm$0.61
4 group	15.97$\pm$0.42
5 group	16.09$\pm$0.18

Rethinking GTR Thresholding. Sec. 3.3 explores the SSL training process using the GTR method. Also, Figure 5 illustrates the distribution of quality indicators on CIFAR-100, showing that the data is distributed in a narrower rather than a hyperellipse, making other unsupervised methods (*e.g.,* K-means) less effective. Our GMM-based approach identifies high-density regions, addressing limitations of prior class-wise methods and improving data-centric analyses. The proposed method offers a nuanced understanding of pseudo-label quality, enhancing SSL training. Future work will extend this approach to complex data distributions and integrate it with other SSL strategies.

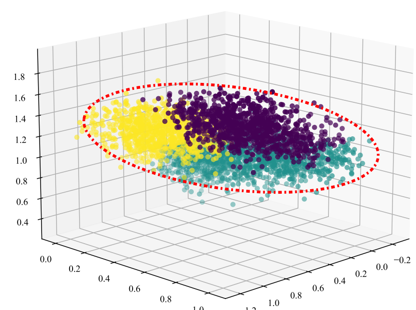


Figure 5: Illustration of the distribution of the quality indicator on CIFAR-100, which is distributed in a narrower rather than a hyperellipse pattern.

5 Related Work

Pseudo Label (Lee et al., 2013) pioneered generating synthetic labels for unlabeled data using a model trained on labeled data, laying the foundation for semi-supervised learning (SSL). Consistency regularization (Samuli & Timo, 2017) followed, ensuring consistent predictions for diverse perspectives of the same data. Subsequent SSL advancements focus on (i) refining high-quality pseudo-label identification and (ii) developing robust thresholding methodologies. Incorporating curriculum learning further enhances deep learning training by structuring data into a curriculum and integrating grouping concepts (Bengio et al., 2009a; Elman, 1993b).

Thresholding High-Quality Pseudo Labels. Confidence-based SSL methods have designed numerous thresholding as pivotal strategies (Xie et al., 2020a; Sohn et al., 2020; Zheng et al., 2022), developing from predefined single threshold (Lee et al., 2013) to considering class-wise adaptive thresholds changing during the SSL training process (Zhang et al., 2021; Yang et al., 2023). FlexMatch (Zhang et al., 2021) introduces class-level thresholds to alleviate the class imbalance in FixMatch (Sohn et al., 2020). SoftMatch (Chen et al., 2022b) balances the quantity and quality of pseudo-labels using a truncated Gaussian function. FreeMatch (Wang et al., 2022b) dynamically adjusts thresholds based on the model's learning state. ShrinkMatch (Yang et al., 2023) and SimMatch (Zheng et al., 2022) integrate self-supervised contrastive learning principles. However, these methods often lack generality and may require extensive tuning for specific tasks or datasets. CR-Match (Fan et al., 2021) introduces FeatDistLoss for regression tasks but falls short. In contrast, the proposed GTR allows for multiple rounds of selection and feedback evaluation by dividing pseudo labels into groups based on kernel density, improving pseudo-label quality.

Tolerance to Inaccurate Pseudo Labels. Early SSL models face heightened sensitivity to low-quality pseudo-labels. The Π model (Rasmus et al., 2015) uses dual perturbations to input samples, while Temporal Ensembling (Samuli & Timo, 2017) maintains an EMA of label predictions. Mean Teacher (Tarvainen & Valpola, 2017) averages model weights, reducing label dependency. Robust training strategies address noisy labels in labeled datasets (Xu et al., 2021; Li et al., 2019a). The GTR method achieves adaptive grouping of pseudo-labels and addresses consistency variations for semi-hard pseudo-labels, resulting in greater robustness.

Curriculum Learning. Curriculum learning enhances deep neural network (DNN) training by structuring data into a progressively challenging curriculum (Bengio et al., 2009b; Elman, 1993a). Initially, models are exposed to simpler samples, gradually introducing more complex ones. Various strategies classify ”easy” and ”hard” samples (Cascante-Bonilla et al., 2021; Castells et al., 2020; Dogan et al., 2020; Hacohen & Weinshall, 2019; Sinha et al., 2020) based on loss, label, feature space, or using fixed or dynamic curricula. Loss-based curricula sequence data using teacher or student network confidence (Hacohen & Weinshall, 2019). Label-based curricula manipulate labels for imbalanced data or increased usage (Zhang et al., 2021; Wang et al., 2019). Feature-based curricula leverage feature density for training from clean to noisy examples (Guo et al., 2018). Fixed curricula employ strategies like EMA of loss (Kong et al., 2021) or reducing contrastive loss weight (Peng et al., 2021). Dynamic curricula use adjustable parameters (Saxena et al., 2019; Li & Gong, 2017), and SuperLoss de-emphasizes high-loss samples (Castells et al., 2020).

6 Conclusion

This paper introduces GTR, a versatile method tailored for SSL scenarios with the aim of enhancing robust thresholding to improve overall performance and convergence speed. Through a comprehensive analysis of the SSL training process and evolving data distributions, we devised the Grouping and Transporting methods, enabling targeted processing for each distinct group. Extensive experiments across diverse classification and regression datasets demonstrate that integrating GTR with popular SSL algorithms yields substantial performance improvements and accelerates convergence. Our approach, grounded in a data-centric perspective and the inherent characteristics of data, not only presents an effective technique for SSL but also holds the potential for broader applicability across various areas.

References

Yelp dataset. `https://www.yelp.com/dataset`, 2014. 7, 16

Eric Arazo, Diego Ortego, Paul Albert, Noel E O’Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2020. 3, 6, 18

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009a. 9

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML ’09, pp. 41–48, New York, NY, USA, 2009b. Association for Computing Machinery. ISBN 9781605585161. 10

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. 18, 19

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019b. 18

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100, 1998. 18

Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Self-paced pseudo-labeling for semi-supervised learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pp. 6912–6920, May 2021. 10

Thibault Castells, Philippe Weinzaepfel, and Jerome Revaud. Superloss: A generic loss for robust curriculum learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 4308–4319. Curran Associates, Inc., 2020. 10

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. Importance of semantic representation: Dataless classification. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 2, pp. 830–835, 2008. 7, 16

Baixu Chen, Junguang Jiang, Ximei Wang, Jianmin Wang, and Mingsheng Long. Debiased pseudo labeling in self-training. *arXiv preprint arXiv:2202.07136*, 2022a. 18

Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality tradeoff in semi-supervised learning. In *The Eleventh International Conference on Learning Representations*, 2022b. 1, 9, 18

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020. 19

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. 7, 16

Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. *Noise reduction in speech processing*, pp. 1–4, 2009. 4

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018. 17

Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Russ R Salakhutdinov. Good semi-supervised learning that requires a bad gan. *Advances in neural information processing systems*, 30, 2017. 19

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 (CVPR)*, pp. 248–255, 2009. 7, 16

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 7

Ürün Dogan, Aniket Anand Deshmukh, Marcin Bronislaw Machura, and Christian Igel. Label-similarity curriculum learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 174–190, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58526-6. 10

Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5884–5888. IEEE, 2018. 1

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 7

Jeffrey L. Elman. Learning and development in neural networks: the importance of starting small. *Cognition*, 48(1):71–99, 1993a. ISSN 0010-0277. 10

Jeffrey L Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99, 1993b. 9

David Eppstein. Fast hierarchical clustering and other applications of dynamic closest pairs. *ACM Journal of Experimental Algorithmics*, 2000. 9

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery and Data Mining*, 1996. 9

Yue Fan, Anna Kukleva, and Bernt Schiele. Revisiting consistency regularization for semi-supervised learning. In *DAGM German Conference on Pattern Recognition*, pp. 63–78. Springer, 2021. 9, 19

Eduardo Fonseca, Manoj Plakal, Daniel PW Ellis, Frederic Font, Xavier Favory, and Xavier Serra. Learning sound event classifiers from web audio with noisy labels. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 21–25. IEEE, 2019. 7, 16

Yixiao Ge, Dapeng Chen, and Hongsheng Li. Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. In *International Conference on Learning Representations*, 2019. 18

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

Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R. Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision – ECCV 2018*, pp. 139–154, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01249-6. 10

Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks, 2019. 10

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 (CVPR)*, pp. 770–778, 2016. 1, 7

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12:2217–2226, 2019. 7, 16

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021. 7

Nanda Kambhatla and Todd K. Leen. Classifying with gaussian mixtures and clusters. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 681–688, Cambridge, MA, USA, 1994. MIT Press. 9

Jiwon Kim, Youngjo Min, Daehwan Kim, Gyuseong Lee, Junyoung Seo, Kwangrok Ryoo, and Seungryong Kim. Conmatch: Semi-supervised learning with confidence-guided consistency regularization. In *European Conference on Computer Vision (ECCV)*, 2022. 1

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 16

Yajing Kong, Liu Liu, Jun Wang, and Dacheng Tao. Adaptive curriculum learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5067–5076, October 2021. 10

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 7, 16

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*, pp. 896, 2013. 1, 3, 6, 9, 17

Hao Li and Maoguo Gong. Self-paced convolutional neural networks. In *International Joint Conference on Artificial Intelligence*, pp. 2110–2116, 2017. 10

Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2019a. 10, 18

Junnan Li, Caiming Xiong, and Steven Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *International Conference on Computer Vision (ICCV)*, 2021. 19

Siyuan Li, Weiyang Jin, Zedong Wang, Fang Wu, Zicheng Liu, Cheng Tan, and Stan Z Li. Semireward: A general reward model for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2024. 1, 3, 4

Xingjian Li, Haoyi Xiong, Hanchao Wang, Yuxuan Rao, Liping Liu, and Jun Huan. Delta: Deep learning transfer using feature map with attention for convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2019b. 19

Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning (ICML)*, 2018. 19

Zicheng Liu, Siyuan Li, Ge Wang, Cheng Tan, Lirong Wu, and Stan Z. Li. Harnessing hard mixed samples with decoupled regularizer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 18

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 16

James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pp. 281–297. Oakland, CA, USA, 1967. 5, 9

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 (TPAMI)*, 41(8):1979–1993, 2018. 19

Augustus Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016. 19

Sungrae Park, JunKeon Park, Su-Jin Shin, and Il-Chul Moon. Adversarial dropout for supervised and semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 19

Jizong Peng, Ping Wang, Christian Desrosiers, and Marco Pedersoli. Self-paced contrastive learning for semi-supervised medical image segmentation with meta-labels. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 16686–16699. Curran Associates, Inc., 2021. 10

Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11557–11568, 2021. 18

Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 1015–1018, 2015. 7, 16

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. *Advances in neural information processing systems*, 28, 2015. 10, 18

Douglas A Reynolds et al. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009. 3, 4, 7

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *2005 Seventh IEEE Workshops on Applications of Computer Vision*. Carnegie Mellon University, 2005. 17

Sebastian Ruder and Barbara Plank. Strong baselines for neural semi-supervised learning under domain shift. In *The 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2018. 18

Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 1041–1044, 2014. 7, 16

Laine Samuli and Aila Timo. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, volume 4, pp. 6, 2017. 9, 10, 18

Shreyas Saxena, Oncel Tuzel, and Dennis DeCoste. Data parameters: A new family of parameters for learning a differentiable curriculum. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 11093–11103, 2019. 10

Divya Shanmugam, Davis Blalock, Guha Balakrishnan, and John Guttag. Better aggregation in test-time augmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1214–1223, 2021. 5

Samarth Sinha, Animesh Garg, and Hugo Larochelle. Curriculum by texture. *CoRR*, abs/2003.01367, 2020. 10

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 9, 18

Jong-Chyi Su and Subhransu Maji. The semi-supervised inaturalist-aves challenge at fgvc7 workshop. In *proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2020. 16

Teppei Suzuki and Ikuro Sato. Adversarial transformations for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5916–5923, 2020. 19

Cheng Tan, Jun Xia, Lirong Wu, and Stan Z Li. Co-learning: Learning from noisy labels with self-supervision. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1405–1413, 2021. 18

Cheng Tan, Zhangyang Gao, Lirong Wu, Siyuan Li, and Stan Z Li. Hyperspherical consistency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7244–7255, 2022. 19

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *31st Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 1, 10, 18

Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, Heli Qi, Zhen Wu, Yu-Feng Li, Satoshi Nakamura, Wei Ye, Marios Savvides, Bhiksha Raj, Takahiro Shinozaki, Bernt Schiele, Jindong Wang, Xing Xie, and Yue Zhang. Usb: A unified semi-supervised learning benchmark. In *Neural Information Processing Systems (NeurIPS)*, 2022a. 2, 16

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. In *The Eleventh International Conference on Learning Representations*, 2022b. 1, 3, 9, 18

Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. Dynamic curriculum learning for imbalanced data classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 10

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020a. 3, 9, 17, 18

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10687–10698, 2020b. 18

Wang Ximei, Gao Jinghan, Long Mingsheng, and Wang Jianmin. Self-tuning for data-efficient deep learning. In *International Conference on Machine Learning (ICML)*, 2021. 19

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 (ICML)*, pp. 11525–11536. PMLR, 2021. 10, 18

Ismet Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Kumar Mahajan. Billion-scale semi-supervised learning for image classification. *ArXiv*, abs/1905.00546, 2019. 18

Lihe Yang, Zhen Zhao, Lei Qi, Yu Qiao, Yinghuan Shi, and Hengshuang Zhao. Shrinking class space for enhanced certainty in semi-supervised learning. *arXiv preprint arXiv:2308.06777*, 2023. 9, 19

David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pp. 189–196, 1995. 17

Kaichao You, Zhi Kou, Mingsheng Long, and Jianmin Wang. Co-tuning for transfer learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 19

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1, 3, 9, 10, 16, 18

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, volume 28, 2015. 7, 16

Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. Simmatch: Semi-supervised learning with similarity matching. *arXiv preprint arXiv:2203.06915*, 2022. 9, 19

Zhi-Hua Zhou and Ming Li. Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 24:415–439, 2010. 18

Supplement Material

The appendix is structured as follows:

(A) In Appendix A, we provide implementation details, including dataset settings, hyperparameter settings, and training schedule.

(B) In Appendix B, we provide additional experimental results, including detailed training time statistics across different datasets and settings.

(C) In Appendix C, we describe the extensive background of semi-supervised learning methods from three aspects.

A Implementation Details

A.1 Dataset Setting

For a fair comparison, we train and evaluate all methods with the same ViT backbones and hyperparameters in Table A2 based on USB (Wang et al., 2022a). As for CV, we evaluate SemiReward on common benchmarks: CIFAR-100 (Krizhevsky et al., 2009), Euro-SAT (Helber et al., 2019), STL-10 (Coates et al., 2011), and ImageNet (Deng et al., 2009) for image modality. Euro-SAT contains Sentinel-2 satellite images covering 13 spectral bands, which is not a natural image dataset as the other three. As for NLP, AG News (Zhang et al., 2015) (news topic material), Yahoo! Answer (Chang et al., 2008) (topic classification), and Yelp Review (yel, 2014) (sentiment classification) to evaluate SSL algorithms on more fine-grained sentiment NLP classification tasks. For audio classification, we choose UrbanSound8k (Salamon et al., 2014) with a maximum length of 4 seconds, ESC-50 (Piczak, 2015) with a maximum length of 5 seconds, and FSDNoisy18k (Fonseca et al., 2019) with the length between 3 seconds and 30 seconds.

Table A1: Settings and details classification datasets in various modalities.

Domain	Dataset	#Label per class	#Training data	#Validation data	#Test data	#Class
CV	CIFAR-100	2 / 4	50,000	-	10,000	100
	STL-10	4 / 10	5,000 / 100,000	-	8,000	10
	EuroSat	2 / 4	16,200	-	5,400	10
	ImageNet	100	1,28,167	-	5,0000	1000
NLP	Yelp Review	50 / 200	250,000	25,000	50,000	5
	AG News	10 / 50	100,000	10,000	7,600	4
	Yahoo! Answer	50 / 200	500,000	50,000	60,000	10
Audio	ESC-50	5 / 10	1,200	400	400	50
	UrbanSound8k	10 / 40	7,079	816	837	10
	FSDnoisy18k	52-171	1,772 / 15,813	-	947	20

A.2 Hyperparameter and Training Settings

Basic Settings. As for classification tasks, regarding hyperparameter settings of SSL classification benchmarks constructed in USB (Wang et al., 2022a), we adopted the original settings with pre-trained Transformers as the backbone and made a few adjustments to adapt to SemiReward, as shown in Table A2. The total training iterations are set to 2^{20}, and an early stop technique is used for calculating the convergence times. Meanwhile, we use the full experimental settings in FlexMatch (Zhang et al., 2021) for ImageNet, which uses 100 classes per class with ResNet-50 as the backbone. All methods are trained from scratch by SGD (Loshchilov & Hutter, 2016) optimizer with a momentum of 0.9, a basic learning rate of 0.03, and a cosine learning rate decay as USB. Note that Semi-AVES (Su & Maji, 2020) uses 224×224 input resolutions and ViT-S-P16-224 with the labeled and unlabeled batch size of 32, and other settings are the same as STL-10. We apply ℓ_1 loss as the basic regression loss. All experiments are implemented with PyTorch and run on NVIDIA A100 GPUs, using 4GPUs training by default.

Settings of GTR with SemiReward. We provide detailed hyper-parameters and settings for SemiReward training. The two-stage online training of the rewarder $\mathcal{R}$ and generator $\mathcal{G}$ is trained by Adam (Kingma & Ba, 2014) optimizer with a learning rate of 0.0005 for all tasks, independent of

Table A2: Hyper-parameters and training schemes of SSL classification tasks based on USB.

Domain	CV			NLP			Audio		
Dataset	CIFAR-100	STL-10	Euro-SAT	AG News	Yahoo! Answer	Yelp-5	UrbanSound8k	FSDNoisy	ESC-50
Image Size	32	96	32	–			–		
Max Length	–			512			4.0	5.0	5.0
Sampling Rate	–			–			16,000		
Model	ViT-S-P4-32	ViT-B-P16-96	ViT-S-P4-32	BERT-Base			HuBERT-Base		
Weight Decay	5e-4			1e-4			5e-4		
Labeled Batch size	16			4			8		
Unlabeled Batch size	16			4			8		
Learning Rate	5e-4	1e-4	5e-5	5e-5	1e-4	5e-5	5e-5	5e-4	1e-4
Layer Decay Rate	0.5	0.95	1.0	0.65	0.65	0.75	0.75	0.75	0.85
Scheduler	$\eta = \eta_0 \cos(\frac{7\pi k}{16K})$								
Model EMA	0.999								
Eval EMA	0.999								
Weak Augmentation	Random Crop, Random Horizontal Flip			–			Random Sub-sample		
Strong Augmentation	RandAugment(Cubuk et al., 2018)			Back-Translation (Xie et al., 2020a)			Random Sub-sample, Gain, Pitch, Speed		

Table A3: Top-1 error rate (%), performance gain, and training speedup times on nine SSL classification datasets with CV, NLP, and Audio modalities in various label settings. R.GTR refers to Reward-based GTR. Performance gains and training speedup times with R.GTR are compared to the baseline (Base).

Domain	Dataset (Setting)	Pseudo Label			FlexMatch			Avg. Speedup
		Base	+SR	**R.GTR**	Base	+SR	**R.GTR**	
Audio	ESC-50 (250)	5.700	7.125	**5.500**	10.053	3.142	**2.395**	×**2.617**
	ESC-50 (500)	6.750	3.214	**3.014**	10.806	4.912	**4.026**	×**2.462**
	FSDnoisy18k (1773)	7.467	8.297	**7.267**	12.133	8.089	**6.954**	×**1.386**
	UrbanSound8k (100)	5.250	5.833	**5.050**	4.728	1.525	**0.905**	×**3.131**
	UrbanSound8k (400)	4.217	6.024	**4.017**	2.833	2.361	**1.676**	×**1.370**
NLP	AG News (40)	2.400	1.714	**1.514**	6.267	1.333	**0.728**	×**5.095**
	AG News (200)	2.889	1.699	**1.499**	3.556	1.693	**1.060**	×**2.641**
	Yahoo! Answer (500)	0.178	0.445	**0.222**	8.711	5.807	**3.851**	×**1.532**
	Yahoo! Answer (2000)	8.689	1.889	**1.689**	8.122	1.692	**1.059**	×**6.406**
	Yelp Review (250)	22.400	22.400	**22.200**	20.066	20.066	**12.393**	×**1.314**
	Yelp Review (1000)	1.822	4.673	**1.622**	21.411	16.470	**11.742**	×**1.473**
CV	CIFAR-100 (200)	9.320	11.314	**9.120**	54.280	49.345	**35.977**	×**1.265**
	CIFAR-100 (400)	14.920	13.564	**13.364**	100.240	94.044	**68.929**	×**1.285**
	STL-10 (20)	0.528	1.320	**0.328**	11.760	8.400	**5.792**	×**1.820**
	STL-10 (40)	0.268	0.693	**0.068**	9.556	7.351	**6.274**	×**2.732**
	Euro-SAT (20)	1.196	5.980	**0.996**	14.320	17.900	**6.887**	×**1.640**
	Euro-SAT (40)	1.092	5.460	**0.892**	21.040	23.378	**11.572**	×**1.521**
	Semi Aves 3959 (3959)	19.212	16.720	**9.375**	82.064	71.248	**35.922**	×**2.167**

the student model's optimization. For each training step after T iterations, $\mathcal{R}$ infers once and selects high-quality pseudo labels for the student with the *average reward score* as the threshold τ. The generator $\mathcal{G}$ utilizes a 4-layer MLP (only containing FC layers and ReLU) with 256, 128, and 64 hidden dimensions.

B Extensive Experiment Results

B.1 Details in Speedup

In Sec. 4, we give the average speed gain but not the specific training time. Table A3 gives the different training times corresponding to the nine sets of data sets in the three modes in the main text. We stipulate that the calculation is on a single NVIDIA A100 GPU to carry out relevant statistics, and the reported unit is the total hours.

C Extensive Related Work

C.1 Self-training

In semi-supervised learning (SSL), self-training frameworks (Rosenberg et al., 2005; Grandvalet & Bengio, 2004; Yarowsky, 1995) play a very important role in unlabeled data utilization. Then, pseudo-labeling (Lee et al., 2013), as one of the classic self-training ways, pioneered the generation of artificial labels for unlabeled data. However, this embodiment faces the need for high-quality labels due to the problem of confirmation bias (Arazo et al., 2020). Subsequent work will mainly address this problem from two perspectives: one is to design a class or combine multiple methods to improve the quality of pseudo-label generation and application, and the other is to consider enhancing the network's acceptance of pseudo-labels, that is, a small number of low-quality pseudo-labels will not affect the overall prediction of the network.

Consistency Regularization. Temporal Ensembling (Samuli & Timo, 2017) first proposed consistency regularization to ensure consistent predictions for similar data points, which has become a basic method for generating high-quality pseudo labels. Based on this, MixMatch (Berthelot et al., 2019b) and its variants (Berthelot et al., 2019a; Liu et al., 2023) performs data augmentation on unlabeled data, inputs multiple data into the same classifier, obtains different predicted classification probabilities, and uses a class method to make the average variance of multiple probability distributions smaller. UDA (Xie et al., 2020a) goes a step further and starts to use two branches of weak and strong augmented samples and regards the predictions of the weak augmentation branch as the target of the strong augmentation branch to improve the consistency of the pseudo-label and predictions. Then, ReMixMatch (Berthelot et al., 2019a) uses the distribution alignment method to encourage the marginal distribution of predictions for unlabeled data to be close to the marginal distribution of ground truth labels. Fixmatch (Sohn et al., 2020) designs a fixed confidence threshold to filter pseudo labels so that the high-quality pseudo-labels can be used in the SSL training process. The following works, like FlexMatch (Zhang et al., 2021), deeply explore the idea of confidence thresholds and propose curriculum learning to dynamically adjust the thresholds generated by pseudo labels based on the training process. Additionally, softmatch (Chen et al., 2022b) shows the trade-off between the quantity and quality of pseudo labels and also derives a truncated Gaussian function to weight sample confidence. Freematch (Wang et al., 2022b) proposes a free matching method that adaptively adjusts confidence thresholds based on the model's learning state. The above methods essentially follow the strategy of training teacher-student distillation. Even the most advanced methods still rely on the manual design of confidence thresholds for screening. Although Meta Pseudo Labels (Pham et al., 2021) proposes to generate more accurate pseudo labels with a meta learner through bi-level optimization, it doubles training times and requires large-scale teacher models.

Tolerance to Inaccurate Pseudo Labels. Early SSL models have a certain sensitivity to low-quality pseudo labels. Then, another aspect of work starts by improving the model's tolerance to errors or low-quality labels. Π-Model (Rasmus et al., 2015) adds two different perturbations to an input sample, inputs the network twice to get the result, and then compares the consistency of the two results. This weakens the impact of low-quality labels but may be less efficient since two forward propagations are required to calculate the loss. Based on this, Temporal Ensembling (Samuli & Timo, 2017) maintains an EMA of label predictions on each training example and penalizes predictions that are inconsistent with this goal. Mean Teacher (Tarvainen & Valpola, 2017) further averages model weights instead of label predictions. This allows the use of fewer labels than sequential integration during training and also improves the accuracy of testing. Meanwhile, another branch of research assumes the labeled datasets are noisy and designs robust training or ad-hoc label selection policies to discriminate inaccurate labels (Xu et al., 2021; Li et al., 2019a; Tan et al., 2021).

C.2 Disagreement-Based Models

From the view of disagreement SSL, it is required to train two or three different networks simultaneously and label unlabeled samples with each other (Zhou & Li, 2010) so that they are less affected by model assumptions and loss functions. Co-training (Blum & Mitchell, 1998) assumes that each data point has two different and complementary views, and each view is sufficient to train a good classifier. Noisy Student (Xie et al., 2020b) is assigned pseudo-labels by a fixed teacher from the previous round, while (Yalniz et al., 2019) scales up this training paradigm to billion-scale unlabeled datasets. MMT (Ge et al., 2019), DivideMix (Li et al., 2019a) learn through multiple models or classifiers through online mutual teaching. Multi-head Tri-training (Ruder & Plank, 2018) uses training to learn three classifiers from three different training sets obtained using bootstrap sampling. In these methods, each classifier head is still trained using potentially incorrect pseudo-labels generated by other heads. Afterward, the classifier for pseudo-labels generated by DST (Chen et al., 2022a) is trained with unused pseudo-labels, thus having better tolerance to inaccurate pseudo-labels.

C.3 Self-supervised Learning for SSL

Self-supervised contrastive learning (CL) approaches (Chen et al., 2020) are also applied to SSL, such as CoMatch (Li et al., 2021) that first introduced CL to the consistency regularization framework. ShrinkMatch (Yang et al., 2023) allows the model to search for contracted class space adaptively. In detail, for each uncertain sample, ShrinkMatch dynamically defines a shrunk class space, including the original top-1 class and less likely classes. Similarly, SimMatch (Zheng et al., 2022) uses semantic and instance similarity for mutual calibration. It uses the labeled data to train a semantic classifier and uses this classifier to generate pseudo labels for the unlabeled data. Meanwhile, ReMixMatch (Berthelot et al., 2019a) and CR-Match (Fan et al., 2021) utilize rotation prediction as the auxiliary task for SSL. Moreover, fine-tuning a pre-trained model on labeled datasets is a widely adopted form of transfer learning (TL), and several recent works (Li et al., 2018; 2019b; You et al., 2020; Ximei et al., 2021) like Self-Tuning (Ximei et al., 2021) combining TL with SSL methods. Self-Tuning (Ximei et al., 2021) and HCR (Tan et al., 2022) introduce CL pre-trained models as the regularization to mitigate confirmation bias in TL.

C.4 Adversarial Training for SSL

In the realm of SSL, innovative approaches have emerged that utilize adversarial training. One approach involves generating synthetic data (Odena, 2016; Dai et al., 2017) using a generator network and assigning it to a new ”generated” class. The goal is to make the discriminator network provide class labels for these synthetic samples. Another line of research creates adversarial examples through techniques like VAT (Miyato et al., 2018), which adds noise to input data; VAdD (Park et al., 2018), introducing an adversarial exit layer into the model's architecture; and RAT (Suzuki & Sato, 2020), extending the concept of noise to input transformations. These methods aim to impose local smoothness constraints on the model's learned representations without relying on pseudo-labels during training. These advancements enhance model robustness and generalization, particularly in data-scarce scenarios, by utilizing latent data distribution structures for more effective learning. This research contributes significantly to improving SSL algorithms, addressing challenges in leveraging unlabeled data to enhance the applicability and performance of machine learning models in real-world applications. These innovative adversarial training approaches are poised to advance SSL.