Boosting Semi-Supervised Learning by bridging high and low-confidence predictions

Khanh-Binh Nguyen
Department of Electrical and Computer Engineering
Sungkyunkwan University, South Korea
binhnk@skku.edu

Joon-Sung Yang
School of Electrical and Electronic Engineering
and Department of Systems Semiconductor Engineering
Yonsei University, South Korea
js.yang@yonsei.ac.kr

Abstract

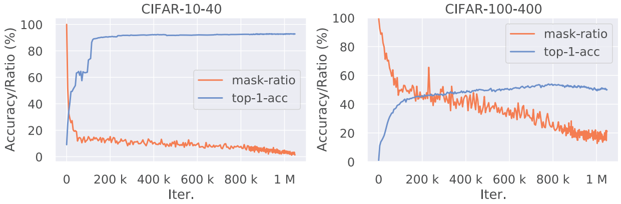
Pseudo-labeling is a crucial technique in semi-supervised learning (SSL), where artificial labels are generated for unlabeled data by a trained model, allowing for the simultaneous training of labeled and unlabeled data in a supervised setting. However, several studies have identified three main issues with pseudo-labeling-based approaches. Firstly, these methods heavily rely on predictions from the trained model, which may not always be accurate, leading to a confirmation bias problem. Secondly, the trained model may be overfitted to easy-to-learn examples, ignoring hard-to-learn ones, resulting in the "Matthew effect" where the already strong become stronger and the weak weaker. Thirdly, most of the low-confidence predictions of unlabeled data are discarded due to the use of a high threshold, leading to an underutilization of unlabeled data during training. To address these issues, we propose a new method called ReFixMatch, which aims to utilize all of the unlabeled data during training, thus improving the generalizability of the model and performance on SSL benchmarks. Notably, ReFixMatch achieves 41.05% top-1 accuracy with 100k labeled examples on ImageNet, outperforming the baseline FixMatch and current state-of-the-art methods.

1. Introduction

The strengths of Deep Neural Networks (DNNs) have been proven through numerous successes in a wide range of tasks, such as image classification [14], speech recognition [1], and natural language processing [42]. Despite the high performance and state-of-the-art benchmarks, the superior performance of DNNs heavily relies on training with a large amount of labeled data [15, 17, 28, 34, 35]. In addition, there are also challenges in using large labeled datasets, such as the availability of the datasets, the cost of collecting and labeling data, etc. To alleviate the dependence on labeled data, semi-supervised learning (SSL) has been proposed. With the advantages of using a large volume of unlabeled data, SSL has become a powerful method for training models. Furthermore, using SSL not only reduces the cost of collecting data but also produces equivalent results to supervised learning approaches. This success has led to the development of many SSL methods [5, 6, 22, 23, 45, 48]. A popular approach of SSL methods is to produce an artificial label for unlabeled data and train the model using the artificial label as ground truth. For example, the pseudo-labeling [23] (categorized as self-training [37, 48] method) uses the model's class prediction as a pseudo-label to train. It is a well-established technique for semi-supervised learning [26, 43], domain adaptation [18, 31], and transfer learning [3]. Unlike pseudo-labeling, consistency regularization uses loss functions such as mean squared error (MSE) or Kullback-Leibler divergence (KL divergence) to minimize the difference between model predictions for different augmented inputs.

Recent work from [43] suggests using a high threshold to filter out only reliable pseudo-labels for training and masking out the rest. FlexMatch [52] improves the performance of FixMatch by applying the Curriculum Pseudo Labeling (CPL) method to let the model learn equally among classes with class-wise dynamic thresholds. CoMatch [25] uses Contrastive Graph Regularization to improve performance by learning jointly-evolved class probabilities and image representations. SimMatch [54] simultaneously considers semantic similarity and instance similarity of the data. While achieving state-of-the-art performance, FixMatch and its variants [21, 25, 52, 54] are still encountering the confirmation bias problem [2]. To eliminate the effects of learning on biased pseudo-label, a number of works have been proposed [7, 16, 49, 50, 54]. However, because of the high threshold setting, a large proportion of unlabeled data with prediction scores below the threshold is discarded during training and never used, especially for hard-to-learn classes. This leads to another major issue that the unlabeled data is not fully exploited for FixMatch and many studies based on it. Furthermore, to tackle the confirmation bias issue, the previous studies introduced additional modules and extra computational overhead.

We visualize the correlation between a top-1 accuracy and a mask ratio on CIFAR-10/100 in Figure 1. It can be seen that while the number of qualified pseudo-labels is increasing by iterations, the accuracy just slowly increases and starts to decrease after 800k iterations. This problem is clearly noticeable for large datasets such as CIFAR-100 in Figure 1b. Furthermore, the number of qualified pseudo-labels that are used during training only ranges from 60% to 80% of total unlabeled data in CIFAR-100.



(a) CIFAR-10 (b) CIFAR-100

Figure 1. Top-1 accuracy vs mask ratio of unlabeled data from FixMatch. **(a)** CIFAR-10 40-label split. **(b)** CIFAR-100 400-label split.

In this work, we propose a simple SSL pipeline, ReFixMatch, which is shown in Figure 2. Conventionally, UDA [47], MixMatch [6], and ReMixMatch [5] train models with ”soft” pseudo-labels for the whole unlabeled dataset. Later, FixMatch [43] simplifies them by using only ”hard” pseudo-labels from the high-confidence predictions. FixMatch also shows that with the high-confidence threshold, sharpening the predictions into ”soft” pseudo-labels does not lead to a significant difference in performance. Hence, they discard the low-confidence predictions during training. Unlike previous approaches, ReFixMatch aims to maximize the utilization of the whole unlabeled dataset to improve generalization during training. Specifically, we bridge the usage of ”hard” pseudo-labels from high-confidence predictions and ”soft” pseudo-labels from low-confidence predictions. Thus, the low-confidence predictions would be considered guesses, and the information from them could be transferred to the model to improve its performance and representation. In this manner, we leverage the advantages of both ”hard” and ”soft” pseudo-labels as well as the whole unlabeled dataset. The use of low-confidence samples has already been well studied in many other related tasks. This usage, however, is still being studied for semi-supervised learning tasks. There are also research that leverages low-confidence predictions, such as [12, 53]. However, in order to enhance the learning process, they either use multiple models or introduce a complicated pipeline. This work presents an efficient yet straightforward approach based on FixMatch, the most widely used SSL pipeline. The novelty of ReFixMatch lies in the simplicity, which helps it outperform SOTA methods, which are much more complex. ReFixMatch adds no overhead to the conventional pipeline since it only uses an extra loss term. Because of this simplicity, many SSL frameworks, including semi-supervised semantic segmentation and object detection can be benefitted from this study. The benefit of introducing ReFixMatch is particularly remarkable on all datasets, especially imbalanced datasets. ReFixMatch achieves 28.60%, 8.39%, and 6.11% error rates when the number of labels is 40, 250, and 1000, respectively, on the STL-10 dataset. Furthermore, on the SVHN dataset, ReFixMatch achieves 2.15% and 1.89%hl error rate; ReFixMatch with CPL gives 2.63% and 2.01% error rates when the label amount is 40 and 1000, respectively, while FlexMatch fails with a large margin. ReFixMatch also improves the convergence speed and the generalization of the model.

To sum up, this paper makes the following contributions:

- We systematically investigate and analyze the importance of low-confidence predictions for unlabeled data in the training of SSL methods.
- We propose a simple yet effective method, ReFixMatch, to leverage the whole unlabeled data set, including high and low-confidence predictions.
- ReFixMatch introduces no additional modules or extra computational overhead, and it can be used with any SSL method to improve performance.
- ReFixMatch establishes a new state-of-the-art performance for semi-supervised learning. ReFixMatch achieves a 41.05% error rate on ImageNet with 100k labeled images and outperforms prior methods.

2. Analysis of high-confidence and low-confidence pseudo-label

In order to examine the importance of low-confidence predictions in the training process, we train FixMatch separately with ”hard” and ”soft” pseudo-labels. The

”hard” pseudo-label training is the conventional FixMatch using high-confidence predictions, while for the ”soft” pseudo-label training, the model is trained only on low-confidence predictions. Specifically, instead of choosing high-confidence predictions as the pseudo-label, we take the low-confidence predictions from weakly-augmented examples, sharpen them by temperature $\mathbf{T} = \mathbf{0.5}$ and compute the KL divergence with the predictions from strongly-augmented.

Table 1. Error rate of FixMatch using high-confidence vs low-confidence predictions on CIFAR-10 with 40, 250, and 1000-label split.

DATASET	HIGH-CONFIDENCE	LOW-CONFIDENCE
CIFAR-10-40	7.47	28.88
CIFAR-10-250	4.86	8.07
CIFAR-10-4000	4.21	8.04

The experiment results from Table 1 show that using only low-confidence predictions to train the model can still achieve a competitive performance with the one using high-confidence predictions on the CIFAR-10 dataset. This shows that the conventional approach of using a high threshold and discarding a large proportion of unlabeled data during training is inefficient and does not fully leverage the unlabeled data. Thus, instead of using only high-confidence predictions, in this work, we bridge the strengths of both high-confidence and low-confidence predictions.

3. ReFixMatch

We propose ReFixMatch, a simple SSL pipeline that considers the information from the whole unlabeled dataset. The main novelty of ReFixMatch is the utilization of unlabeled examples that have a prediction probability lower than the threshold τ. In the following section, we explain the whole process of ReFixMatch for semi-supervised image classification problems.

3.1. ReFixMatch pipeline

Our proposed ReFixMatch pipeline consists of two phases, as shown in Figure 2. In the training phase, we perform supervised training for the model with labeled data and evaluate the standard cross-entropy loss. During the inference phase, two perturbed versions of unlabeled images, which are either weakly or strongly augmented, are created. Then, for the unlabeled data, pseudo-labels are generated from the high-confidence predictions of the weakly-augmented unlabeled version. Next, these pseudo-labels are used to supervise the model prediction of the strongly-augmented version on the next iteration, together with labeled data. Last, we sharpen the low-confidence predictions of the weakly-augmented unlabeled version. A KL divergence loss function is used for the sharpened low-confidence predictions and the predictions from the strongly-augmented version.

While minimizing the cross entropy loss between model logits and hard one-hot targets remains the go-to recipe for supervised classification training, learning from soft tar-get emerges in many lines of research. Label Smoothing [36, 43] is a straightforward method that applies a fixed smoothing (softening) factor α to the hard one-hot classification target. The motivation is that label smoothing prevents the model from becoming over-confident.

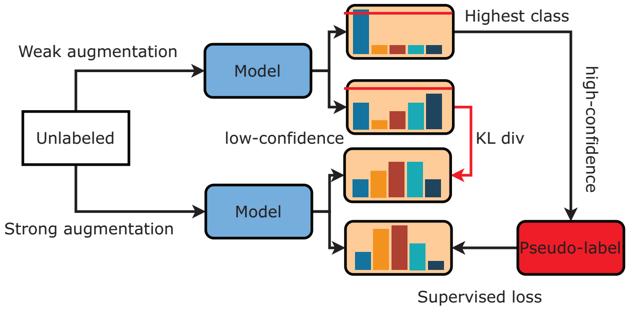


Figure 2. Diagram of the proposed ReFixMatch. For weakly-augmented predictions, the high-confidence predictions are converted to one-hot pseudo-label, while the low-confidence predictions are sharpened with temperature $\mathbf{T}$. We measure the Kullback-Leibler divergence loss for the sharpened low-confidence predictions with the strongly-augmented predictions from the same input and the cross-entropy loss for the ”hard” pseudo-label.

3.2. Preliminaries

In SSL, the training data consists of labeled and unlabeled data. Let $\mathcal{X} = \{(x_b, y_b) : b \in (1, \ldots, B)\}$ be a batch of B labeled examples, where x_b is training examples and y_b is one-hot labels, and $\mathcal{U} = \{u_b : b \in (1, \ldots, \mu B)\}$ be a batch of μB unlabeled examples where μ is a hyperparameter determining the relative sizes of $\mathcal{X}$ and $\mathcal{U}$.

We construct the loss function of our proposed ReFixMatch with the supervised loss, which is a standard cross-entropy loss ($\mathcal{L}_s^{\mathrm{CE}}$) for the labeled data, and the unsupervised loss, including the KL divergence loss ($\mathcal{L}_{\mathrm{KL}}$) for the low-confidence predictions and the standard cross-entropy loss ($\mathcal{L}_u^{\mathrm{CE}}$) for high-confidence predictions.

$$\mathcal{L}_{\mathrm{SSL}} = \mathcal{L}_s^{\mathrm{CE}} + \lambda_u \mathcal{L}_u \tag{1}$$

where λ_u is the fixed weight for the unlabeled data loss.

Specifically, $\mathcal{L}_s^{\mathrm{CE}}$ is a standard cross-entropy loss on weakly-augmented labeled data:

$$\mathcal{L}_s^{\mathrm{CE}} = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}\left(y_b, p_m\left(y \mid \mathcal{A}_w\left(x_b\right)\right)\right) \tag{2}$$

where $p_m(y|x)$ is the predicted class distribution of the model for input x, and $H(p,q)$ denotes the "hard" label cross-entropy between two probability distributions, p and q. Then, let $\mathcal{A}_w$ be the weakly (i.e., random crop and flip) augmentation and $\mathcal{A}_s$ be the strongly (i.e., RandAugment [10]) augmentation for unlabeled data, respectively. $\mathcal{L}_u$ is defined as a total of the standard cross-entropy loss ($\mathcal{L}_u^{\mathrm{CE}}$) and the KL divergence loss ($\mathcal{L}_{\mathrm{KL}}$). $\mathcal{L}_u^{\mathrm{CE}}$ is the loss between the high-confidence pseudo-label of weakly-augmented unlabeled data and the predictions of the model for strongly-augmented unlabeled data. $\mathcal{L}_{\mathrm{KL}}$ is the KL divergence loss between the sharpened low-confidence predictions of weakly-augmented examples $\mathcal{A}_w$ and the predictions of strongly-augmented examples $\mathcal{A}_s$, defined as:

$$\mathcal{L}_u = \mathcal{L}_u^{\mathrm{CE}} + \mathcal{L}_{\mathrm{KL}} \tag{3}$$

$$\mathcal{L}_u^{\mathrm{CE}} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\left(\max\left(q_b\right) \geq \tau\right) \mathrm{H}\left(\hat{q}_b, p_m\left(y \mid \mathcal{A}_s\left(u_b\right)\right)\right) \tag{4}$$

$$\mathcal{L}_{\mathrm{KL}} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\left(\max\left(q_b\right) < \tau\right) D_{\mathrm{KL}}\left(p_s^w \mid p_m\left(y \mid \mathcal{A}_s\left(u_b\right)\right)\right) \tag{5}$$

$$p_s^w\left(y \mid \mathcal{A}_w\left(u_b\right)\right) = \frac{\exp\left(z_b/\mathbf{T}\right)}{\sum_k \exp\left(z_k/\mathbf{T}\right)} \tag{6}$$

where $\hat{q}_b = \arg\max\left(q_b\right)$ is the pseudo-label with $q_b = p_m\left(y \mid \mathcal{A}_w\left(u_b\right)\right)$ for input $\mathcal{A}_w\left(u_b\right)$, τ is the threshold for choosing pseudo-label, D_{KL} denotes the KL divergence function, z_b is the logits for example, $\mathcal{A}_w\left(u_b\right)$ and $\mathbf{T}$ is the temperature for sharpening.

3.3. Algorithm

The algorithm for ReFixMatch is presented in Algorithm 1. Compared to FlexMatch, the ReFixMatch algorithm is much simpler, as it does not require computation of the threshold for each iteration. The algorithm of ReFixMatch is as simple as FixMatch, with only additional loss for low-confidence predictions. Therefore, ReFixMatch does not require any additional budget compared to prior methods.

Algorithm 1: ReFixMatch algorithm

```
Input: Labeled batch
       $\mathcal{X} = (x_b, y_b) : b \in (1, \ldots, B)$, unlabeled
       batch $\mathcal{U} = u_b : b \in (1, \ldots, \mu B)$, confidence
       threshold $\tau$, unlabeled data ratio $\mu$,
       unlabeled loss weight $\lambda_u$, temperature $\mathbf{T}$
$\mathcal{L}_s^{\mathrm{CE}} = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}\left(y_b, p_m\left(y \mid \mathcal{A}_w\left(x_b\right)\right)\right)$
  // Cross-entropy loss for labeled data
for $b = 1$ to $\mu B$ do
    /* Compute prediction after applying
       weak data augmentation of $u_b$                    */
    $q_b = p_m\left(y \mid \mathcal{A}_w\left(u_b\right); \theta\right)$
    /* Sharpen the low-confidence
       predictions                                        */
    $p_s^w\left(y \mid \mathcal{A}_w\left(u_b\right)\right) = \frac{\exp\left(z_b/\mathbf{T}\right)}{\sum_k \exp\left(z_k/\mathbf{T}\right)}$
/* Cross-entropy loss with pseudo-label and
   confidence threshold for high-confidence
   unlabeled data                                         */
$\mathcal{L}_u^{CE} =$
  $\frac{1}{\mu B}\left(\sum_{b=1}^{\mu B} \mathbb{1}\left(\max\left(q_b\right) \geq \tau\right) \mathrm{H}\left(\hat{q}_b, p_m\left(y \mid \mathcal{A}_s\left(u_b\right)\right)\right)\right)$
/* Kullback-Leibler divergence loss with
   sharpened pseudo-label and confidence
   threshold for low-confidence unlabeled
   data                                                   */
$\mathcal{L}_{KL} = \frac{1}{\mu B}\left(\sum_{b=1}^{\mu B} \mathbb{1}\left(\max\left(q_b\right) < \tau\right)\right.$
  $\left.D_{\mathrm{KL}}\left(p_s^w \mid p_m\left(y \mid \mathcal{A}_s\left(u_b\right)\right)\right)\right)$
$\mathcal{L}_u = \mathcal{L}_u^{CE} + \mathcal{L}_{KL}$
return $\mathcal{L}_s^{CE} + \lambda_u \mathcal{L}_u$
```

4. Experiments

We evaluate ReFixMatch on common datasets such as CIFAR-10/100 [20], SVHN [32], STL-10 [9], and ImageNet [11] under various labeled data amounts. We mainly compare our proposed method with recent state-of-the-art methods such as UDA [47], FixMatch [43], FlexMatch [52], CoMatch [25], SimMatch [54], and AdaMatch [7]. We also include a fully-supervised experiment for each dataset. The implementation and evaluation of all methods are based on TorchSSL[1].

We use the same training hyperparameters for a fair comparison of UDA, FixMatch, and FlexMatch methods. There are only minor differences for some hyperparameters regarding each method algorithm settings. Standard stochastic gradient descent (SGD) with a momentum of 0.9 is used as an optimizer in all experiments [33, 44]. An initial learning rate of 0.03 with a cosine annealing learning rate scheduler [27] is used for a total of 2^{20} training iterations. We also conducted an exponential moving average with a momentum of 0.999. The batch size of labeled data is 64 for all datasets except ImageNet. μ is set to 7 for CIFAR-10/100, SVHN and STL-10, and it is set to 1 for ImageNet. τ is set to 0.8 for UDA and is set to 0.95 for FixMatch, FlexMatch, and ReFixMatch. These configurations follow the original papers [43, 47, 52]. We set $\mathbf{T}$ to 0.4 for UDA and 0.5 for ReFixMatch. The strong augmentation in our experiments is RandAugment [10]. For the ImageNet dataset, we use ResNet-50 [20] and for other datasets, we use variants of Wide-ResNet (WRN).

[1] https://github.com/TorchSSL/TorchSSL

Table 2. Error rates on CIFAR-10/100, SVHN, and STL-10 datasets on 5 different folds. All models are tested using the same code base from TorchSSL. **Bold** indicates best result and Underline indicates second-best result.

DATASET	CIFAR-10			CIFAR-100			STL-10			SVHN	
LABEL AMOUNT	40	250	4000	400	2500	10000	40	250	1000	40	1000
UDA [47]	10.62$_{\pm 3.75}$	5.16$_{\pm 0.06}$	4.29$_{\pm 0.07}$	46.39$_{\pm 1.59}$	27.73$_{\pm 0.21}$	22.49$_{\pm 0.23}$	37.42$_{\pm 8.44}$	9.72$_{\pm 1.15}$	6.64$_{\pm 0.17}$	5.12$_{\pm 4.27}$	1.89$_{\pm 0.01}$
FIXMATCH [43]	7.47$_{\pm 0.28}$	4.86$_{\pm 0.05}$	4.21$_{\pm 0.08}$	46.42$_{\pm 0.82}$	28.03$_{\pm 0.16}$	22.20$_{\pm 0.12}$	35.97$_{\pm 4.14}$	9.81$_{\pm 1.04}$	6.25$_{\pm 0.33}$	3.81$_{\pm 1.18}$	1.96$_{\pm 0.03}$
FLEXMATCH [52]	4.97$_{\pm 0.06}$	4.98$_{\pm 0.09}$	4.19$_{\pm 0.01}$	39.94$_{\pm 1.62}$	26.49$_{\pm 0.20}$	21.90$_{\pm 0.15}$	29.15$_{\pm 4.16}$	8.23$_{\pm 0.39}$	5.77$_{\pm 0.18}$	8.19$_{\pm 3.20}$	6.72$_{\pm 0.30}$
COMATCH [25]	6.51$_{\pm 1.18}$	5.35$_{\pm 0.14}$	4.27$_{\pm 0.12}$	53.41$_{\pm 2.36}$	29.78$_{\pm 0.11}$	22.11$_{\pm 0.22}$	**13.74**$_{\pm 4.20}$	**7.63**$_{\pm 0.94}$	**5.71**$_{\pm 0.08}$	8.20$_{\pm 5.32}$	2.01$_{\pm 0.04}$
SIMMATCH [54]	5.38$_{\pm 0.01}$	5.36$_{\pm 0.08}$	4.41$_{\pm 0.07}$	39.32$_{\pm 0.72}$	**26.21**$_{\pm 0.37}$	**21.50**$_{\pm 0.11}$	16.98$_{\pm 4.24}$	8.27$_{\pm 0.04}$	5.74$_{\pm 0.31}$	7.60$_{\pm 2.11}$	2.05$_{\pm 0.05}$
ADAMATCH [7]	5.09$_{\pm 0.21}$	5.13$_{\pm 0.05}$	4.36$_{\pm 0.05}$	**38.08**$_{\pm 1.35}$	26.66$_{\pm 0.33}$	21.99$_{\pm 0.15}$	19.95$_{\pm 5.17}$	8.59$_{\pm 0.43}$	6.01$_{\pm 0.02}$	6.14$_{\pm 5.35}$	2.02$_{\pm 0.05}$
REFIXMATCH	**4.94**$_{\pm 0.01}$	**4.83**$_{\pm 0.05}$	4.18$_{\pm 0.05}$	46.12$_{\pm 1.07}$	27.28$_{\pm 0.22}$	21.60$_{\pm 0.04}$	28.60$_{\pm 4.21}$	8.21$_{\pm 0.30}$	5.74$_{\pm 0.30}$	**2.15**$_{\pm 1.23}$	**1.89**$_{\pm 0.03}$
REFIXMATCH + CPL	4.95$_{\pm 0.05}$	4.85$_{\pm 0.06}$	**4.13**$_{\pm 0.02}$	46.73$_{\pm 1.37}$	27.25$_{\pm 0.25}$	21.78$_{\pm 0.04}$	28.66$_{\pm 4.40}$	8.23$_{\pm 0.31}$	5.76$_{\pm 0.42}$	2.63$_{\pm 1.46}$	2.01$_{\pm 0.05}$
FULLY-SUPERVISED	4.62$_{\pm 0.05}$			19.30$_{\pm 0.09}$			NONE			2.13$_{\pm 0.02}$	

4.1. CIFAR-10/100, STL-10, SVHN

We evaluate the best error rate by averaging the results of five runs with different random seeds for each method. The classification error rates on the CIFAR-10/100, STL-10, and SVHN datasets are given in Table 2.

We employ Wide-ResNet [51] as a backbone model for experiments. Detailed model selection is reported in Appendix **??**. ReFixMatch achieves the best performance on most of the datasets with different amounts of labels, as shown in Table 2. ReFixMatch not only achieves high performance across all datasets but also performs well on the SVHN dataset, while FlexMatch performs less favorably on imbalanced datasets such as the SVHN [52]. Especially, ReFixMatch using CPL improves the results of FlexMatch on the SVHN dataset. This proves that our proposed method with the strategy of leveraging the whole unlabeled dataset can mitigate the overfitting issue when training on small and imbalanced datasets. However, since ReFixMatch and FlexMatch have the same approach in common that helps the model utilize more data, using CPL with our proposed ReFixMatch results in a degradation of performance on balanced datasets. Moreover, CPL improves the number of ”hard” pseudo-labels, thus reducing the effect of ReFixMatch. It should be noted that ReFixMatch adds no overhead, while CoMatch and SimMatch use additional complex modules. They also use the distribution alignment technique, which provides much better results.

4.2. ImageNet

We further evaluate ReFixMatch on large and complex datasets such as ImageNet [11]. We train the models with 100k of labeled data. Furthermore, because the ImageNet dataset is large and complex, we set the τ threshold value to 0.7 to improve the capture of samples with the correct pseudo-label. The batch size is 128 and the weight decay is 0.0003 for the 100k labels experiment. For 10% experiments, we follow the settings in [25, 43, 54].

As reported in Table 3, ReFixMatch outperforms FixMatch, FlexMatch, and CoMatch with 41.05% and 19.01% for the top-1 and top-5 error rates, respectively. This result clearly indicates that our proposed ReFixMatch can help boost performance for large and complex datasets such as ImageNet, especially when they are imbalanced (the number of images per class in the ImageNet dataset ranges between 732 and 1300). Besides, when applying CPL from FlexMatch [52] to ReFixMatch, the results drop to 41.75% and 19.36% for the top-1 and top-5 error rates, respectively, as we explained in Section 4.1. In addition, ReFixMatch also surpasses the best performance of CoMatch and SimMatch by a large margin with 10% labels; details are in Appendix **??**.

Table 3. Error rate results on ImageNet.

METHOD	TOP-1	TOP-5	TOP-1	TOP-5
	100K		10%	
FIXMATCH [43]	43.66	21.80	28.50	10.90
FLEXMATCH [52]	41.85	19.48	-	-
COMATCH [25]	42.17	19.64	26.30	8.60
SIMMATCH [54]	-	-	25.60	8.40
REFIXMATCH	**41.05**	**19.01**	**24.80**	**8.10**
REFIXMATCH+CPL	41.75	19.36	-	-

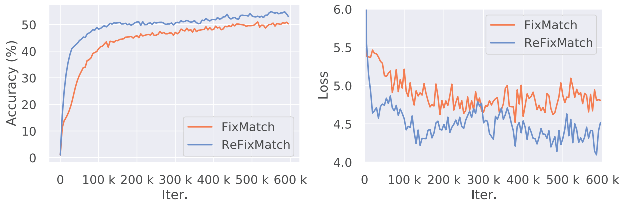
4.3. Ablation Study

4.3.1 Training Efficiency

The convergence speed of our proposed ReFixMatch is extremely noticeable through our extensive experiments. As we can see in Figure 3, on CIFAR-100, ReFixMatch achieves over 40% of accuracy within the first few iterations, while FixMatch nearly hits 20%. After 200k iterations, ReFixMatch achieves over 50% accuracy, while FixMatch only achieves around 45% of accuracy. Moreover, the loss landscape of our proposed ReFixMatch also decreases faster than that of FixMatch. In Figure 3, we visualize the validation loss and top-1 accuracy of both FixMatch and ReFixMatch on the CIFAR-100 dataset with a 400-label split over 600k iterations for a better view of the difference.

Table 4. Class-wise accuracy comparison on CIFAR-10 40-label split.

CLASS NUMBER	0	1	2	3	4	5	6	7	8	9
FIXMATCH	0.964	0.982	0.697	0.852	0.974	**0.890**	**0.987**	0.970	0.982	**0.981**
REFIXMATCH	**0.971**	**0.984**	**0.905**	**0.881**	**0.977**	0.872	0.984	**0.974**	**0.984**	0.98
FLEXMATCH	0.967	0.980	**0.921**	0.866	0.957	0.883	**0.988**	**0.975**	0.982	0.968
REFIXMATCH + CPL [52]	**0.967**	**0.983**	0.915	**0.876**	**0.969**	**0.889**	0.971	0.974	**0.985**	**0.973**



(a) Top-1 accuracy (b) Eval loss

Figure 3. Convergence analysis of ReFixMatch and FixMatch. **(a)**, **(b)** depict top-1 accuracy and loss on CIFAR-100 with 400 labels.

4.3.2 Class-wise accuracy on CIFAR-10 40-label split

In Table 4, we present a thorough comparison of class-wise accuracy. Our proposed ReFixMatch maintains high accuracy in easy-to-learn classes while simultaneously improving the accuracy in hard-to-learn classes. ReFixMatch's final class-wise accuracy is balanced across classes, including hard-to-learn classes. This demonstrates that employing both high and low-confidence predictions enhances not just the overall performance of the trained model but also the performance of each class. ReFixMatch clearly outperforms FixMatch in class-wise accuracy in the evaluation phase for hard-to-learn classes.

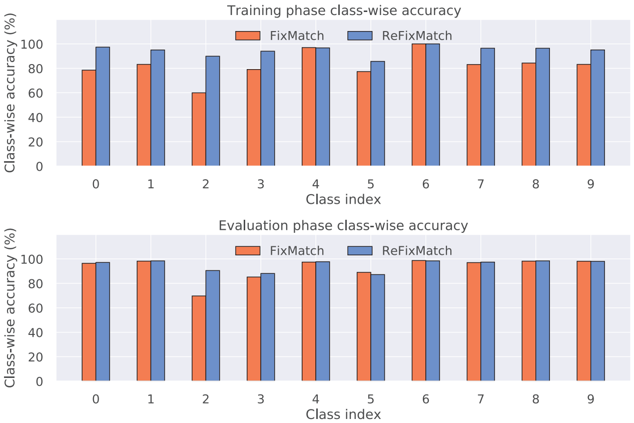


Figure 4. Class-wise accuracy comparison on CIFAR-10 40-label split at the best iteration of FixMatch and ReFixMatch.

The class-wise accuracy from the training phase, as shown in Figure 4, indicates that leveraging the whole unlabeled dataset can improve the generalization of the model. The results show that our ReFixMatch class-wise accuracy is much higher than FixMatch, and it also is balanced between easy-to-learn and hard-to-learn classes.

Figure 5 shows the accuracy of the pseudo-label during training on the CIFAR-10 40-label split. We can see that ReFixMatch can improve the accuracy of the pseudo-label over both FixMatch and FlexMatch.

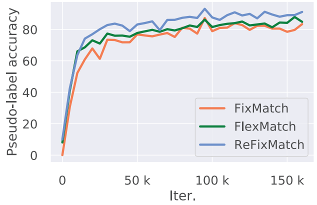
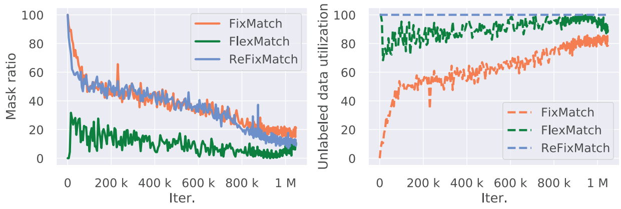


Figure 5. Pseudo-label accuracy.

4.3.3 Data utilization and mask ratio

We present the unlabeled data utilization and mask ratio of FixMatch and ReFixMatch on the CIFAR-100 dataset with a 400-label split in Figures 6a, 6b. ReFixMatch helps reduce the mask-out data ratio and always uses the whole unlabeled dataset during training. It also can be seen that the mask ratio of ReFixMatch less fluctuates than FixMatch. It should be noted that FlexMatch has a lower mask ratio since it uses a lower threshold for each class, which allows the more low-confidence prediction to be used as pseudo-label but also introduces more noise to the model.



(a) Mask ratio (b) Unlabeled data utilization

Figure 6. Unlabeled data utilization and mask ratio on CIFAR-100 dataset with 400-label split.

4.3.4 CIFAR-10 Confusion Matrix

Figure 7 shows the confusion matrix of FixMatch, FlexMatch, and ReFixMatch on the CIFAR-10 dataset with a

Table 5. Precision, recall, F1-score and AUC results on SVHN and STL-10.

Dataset	SVHN-40				STL-10-40			
Criteria	Precision	Recall	F1 Score	AUC	Precision	Recall	F1 Score	AUC
UDA [47]	**0.9783**	0.9776	0.9777	0.9977	0.6385	0.5319	0.4765	0.8581
FixMatch [43]	0.9731	0.9706	0.9716	0.9962	0.6590	0.5830	0.5405	0.8862
FlexMatch [52]	0.9566	0.9691	0.9625	0.9975	0.6403	0.6755	0.6518	0.9249
ReFixMatch	0.9779	**0.9777**	**0.9778**	**0.9978**	**0.8518**	**0.7140**	**0.6908**	**0.9571**

40-label split.

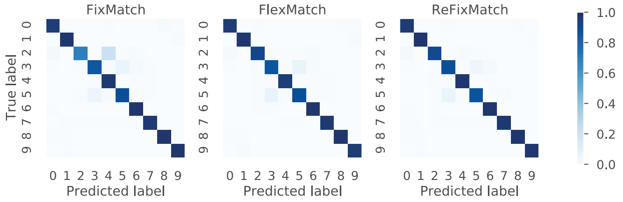
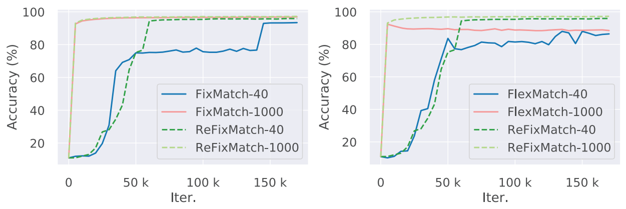


Figure 7. Confusion matrix of FixMatch, FlexMatch, and ReFixMatch features on CIFAR-10 with the 40-label split.

Precision, Recall, F1 and AUC

We also report precision, recall, F1-score, and AUC (area under curve) results on SVHN and STL-10 datasets with 40 labels to completely evaluate the performance of all methods in a classification setting. As demonstrated in Table 5, ReFixMatch has the best performance in accuracy, recall, F1-score, and AUC, while also having lower error rates. These measurements, along with error rates (accuracy), demonstrate the robust performance of our proposed method. Especially on STL-10, simple ReFixMatch improves precision and recall by a large margin compared with prior methods.

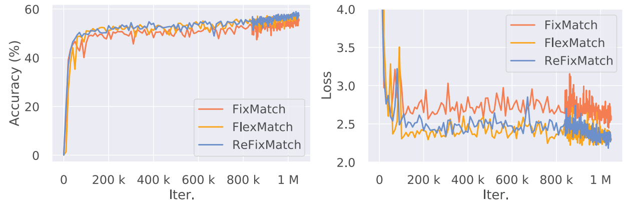
4.3.5 Imbalance dataset problem



(a) FixMatch vs ReFixMatch (b) FlexMatch vs ReFixMatch+CPL

Figure 8. Accuracy comparison of Figure 8a: FixMatch vs ReFixMatch and Figure 8b: FlexMatch vs ReFixMatch+CPL for first 150k iterations on SVHN dataset with 40-label and 1000-label split.

For example, when dealing with imbalanced datasets such as the SVHN and ImageNet datasets, ReFixMatch outperforms both FixMatch and FlexMatch. FlexMatch fails on the SVHN dataset since CPL may yield low final thresholds for the tail classes, allowing noisy pseudo-labeled samples to be accepted and trained. In contrast, ReFixMatch preserves the high fixed threshold of FixMatch, and the final results on the SVHN dataset are improved. In addition, ReFixMatch outperforms both FixMatch and FlexMatch by a large margin without additional modules on the ImageNet.



(a) Top-1 accuracy (b) Eval loss

Figure 9. Accuracy and loss comparison of FixMatch, FlexMatch, and ReFixMatch on ImageNet dataset.

4.3.6 Long-tailed issue

To further prove the effectiveness of ReFixMatch, we evaluate ReFixMatch on the imbalanced SSL setting. We conduct experiments on CIFAR-10-LT, SVHN-LT, and CIFAR-100-LT with different imbalance ratios. Following [24, 36, 46], we use WRN-28-2 as the backbone. We consider long-tailed (LT) imbalance, where the number of data points exponentially decreases from the first class to the last, i.e., $N_k = N_1 \times \lambda^{-\frac{k-1}{L-1}}$, where $\lambda = \frac{N_1}{N_k}$. For CIFAR-10, we set $\lambda = 100, N_1 = 1000$, and $\beta = 10\%, 20\%$, and 30%, respectively. Similarly, we set $\lambda = 100, N_1 = 1000$, and $\beta = 20\%$ for SVHN. And for CIFAR-100, we set $\lambda = 20, N_1 = 200$, and $\beta = 40$. The results are recorded in Table 6 with an average of three different runs.

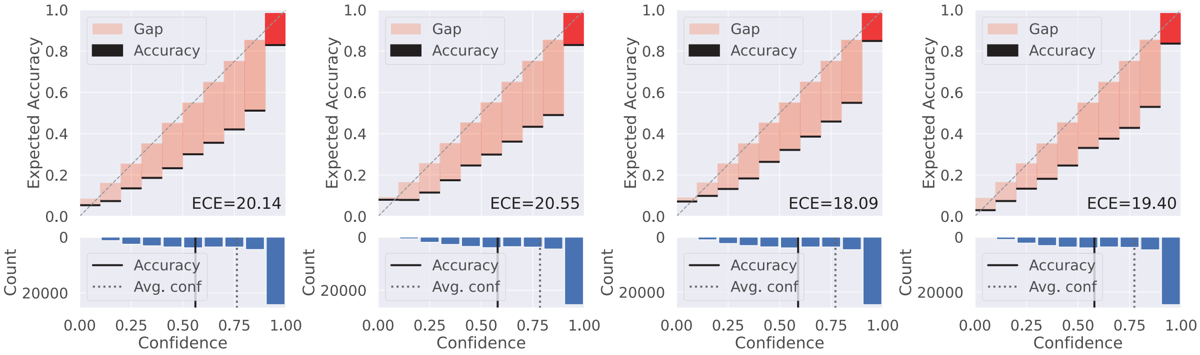
Surprisingly, ReFixMatch boosts the performance by a large margin when used with ABC [24]. With an accuracy of 85.42%, ReFixMatch outperforms ABC with an 8.2% improvement when β equals 10%.

4.4. Calibration of SSL

[8] suggests addressing confirmation bias from the calibration perspective. We measure the calibration of FixMatch, FlexMatch, ReFixMatch, and ReFixMatch+CPL

Table 6. Overall accuracy under the long-tailed setting

	CIFAR-10-LT			SVHN-LT	CIFAR-100-LT
Algorithm		$\lambda = 100$		$\lambda = 100$	$\lambda = 20$
	$\beta = 10\%$	$\beta = 20\%$	$\beta = 30\%$	$\beta = 20\%$	$\beta = 40\%$
Vanilla	-	55.3±1.30	-	77.0±0.67	40.1±1.15
VAT [30]	-	55.3±0.88	-	81.3±0.47	40.4±0.34
BALMS [36]	-	70.7±0.59	-	87.6±0.53	50.2±0.54
FixMatch [43]	70.0±0.59	72.3±0.33	74.9±0.63	88.0±0.30	51.0±0.20
w/ CReST+PDA [46]	73.9±0.40	76.6±0.46	74.9±0.63	89.1±0.69	51.6±0.29
w/ DARP [19]	-	73.7±0.98	-	88.6±0.19	51.4±0.37
w/ DARP+cRT [19]	74.6±0.98	78.1±0.895	77.6±0.73	89.9±0.44	54.7±0.46
w/ ABC [24]	77.2±1.60	81.1±0.82	81.5±0.29	92.0±0.38	56.3±0.19
w/ ABC + ReFixMatch	**85.4±0.01**	**81.3±0.75**	**82.1±0.25**	**92.1±0.06**	**57.0±0.09**



(a) FixMatch (b) FlexMatch (c) ReFixMatch (d) ReFixMatch+CPL

Figure 10. Reliability diagrams (top) and confidence histograms (bottom) for ImageNet dataset.

trained on the ImageNet dataset with 100k labels [2]. Several common calibration indicators are used: Expected Calibration Error (ECE), confidence histogram, and reliability diagram. As shown in Fig. 10, even though FlexMatch has higher accuracy than FixMatch, its ECE value of 20.55 is larger than that of FixMatch, at 20.14, indicating poorer probability estimation. On the other hand, ReFixMatch achieves both higher accuracy and a lower ECE value of 18.09, which proves that it can reduce the confirmation bias and produce a better calibrated model. Furthermore, despite having a lower performance than FlexMatch, ReFixMatch+CPL still achieves an ECE value of 19.40.

5. Related Work

In SSL, self-training is extensively used [29, 41]. The model's output probabilities are treated as ”soft” labels for unlabeled data. Pseudo-labeling is a self-training variation that converts the probability to ”hard” labels [23]. To alleviate the confirmation bias problem, pseudo-labeling is used together with confidence-based thresholding, which keeps unlabeled samples only when predictions are sufficiently confident [38,43,47,52]. Consistency regularization is used to make predictions on perturbed versions of unlabeled data match the pseudo-label [4, 22, 40]. There are many techniques to generate perturbed versions of unlabeled data such as data augmentation [13], stochastic regularization [22,39], and adversarial perturbations [30].

FixMatch [43] presents a hybrid approach for SSL that combines pseudo-labeling and consistency regularization. The qualified pseudo-labeling in FixMatch creates a sharpening-like effect that promotes the ability of the model to give high-confidence predictions. FlexMatch proposes a Curriculum Pseudo Labeling (CPL) approach, which allows standard SSL to train with a dynamic threshold for each class. CPL is a dynamic thresholding strategy since it dynamically adjusts the threshold for each class after each iteration, allowing better performance for each class.

[25] propose CoMatch, which combines the ideas of consistency regularization and contrastive learning, in which the target similarity of two instances is measured by the similarity of two class probability distributions, and it achieves the current state-of-the-art semi-supervised learning performance. However, the hyperparameters are extremely sensitive, and the optimal temperature and threshold vary for different datasets and settings.

SimMatch [54] proposes a novel semi-supervised learning framework that simultaneously considers semantic similarity and instance similarity. It shows that by considering consistency regularization on both the semantic level and instance level, SimMatch improves its performance and

[2] https://github.com/hollance/reliability-diagrams

achieves state-of-the-art semi-supervised learning.

6. Conclusions

In this paper, we present ReFixMatch, a new semi-supervised learning pipeline that improves upon the conventional FixMatch algorithm by utilizing both high-confidence and low-confidence predictions. Despite its simplicity, ReFixMatch can significantly improve the generalization of the model and boost performance without any additional computational overheads. ReFixMatch outperforms the conventional state-of-the-art methods by a large margin across datasets without introducing additional modules or computational overheads.

7. Acknowledgement

This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korean Government through the Ministry of Science and ICT (MSIT) under Grant 2021-0-00106 and Grant 2022-0-00971.

References

[1] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR, 2016. 1

[2] 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)*, pages 1–8. IEEE, 2020. 2

[3] Andrew Arnold, Ramesh Nallapati, and William W Cohen. A comparative study of methods for transductive transfer learning. In *Seventh IEEE international conference on data mining workshops (ICDMW 2007)*, pages 77–82. IEEE, 2007. 1

[4] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *NIPS*, 2014. 8

[5] David Berthelot, Nicholas Carlini, Ekin Dogus Cubuk, Alexey Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *ICLR*, 2020. 1, 2

[6] 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, 2019. 1, 2

[7] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alex Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. *arXiv preprint arXiv:2106.04732*, 2021. 2, 4, 5

[8] Mingcai Chen, Yuntao Du, Yi Zhang, Shuwei Qian, and Chongjun Wang. Semi-supervised learning with multi-head co-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6278–6286, 2022. 7

[9] 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*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 4

[10] Ekin Dogus Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3008–3017, 2020. 4

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4, 5

[12] Zhengyang Feng, Qianyu Zhou, Qiqi Gu, Xin Tan, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. Dmt: Dynamic mutual training for semi-supervised learning. *Pattern Recognition*, page 108777, 2022. 2

[13] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. *arXiv preprint arXiv:1706.05208*, 2017. 8

[14] 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*, pages 770–778, 2016. 1

[15] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically, 2017. 1

[16] Zijian Hu, Zhengyu Yang, Xuefeng Hu, and Ram Nevatia. Simple: Similar pseudo label exploitation for semi-supervised classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15099–15108, 2021. 2

[17] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling, 2016. 1

[18] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019. 1

[19] 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. 8

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4

[21] Chia-Wen Kuo, Chih-Yao Ma, Jia-Bin Huang, and Zsolt Kira. Featmatch: Feature-based augmentation for semi-supervised learning. In *European Conference on Computer Vision*, pages 479–495. Springer, 2020. 2

[22] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *ArXiv*, abs/1610.02242, 2017. 1, 8

[23] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. 2013. 1, 8

[24] 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. 7, 8

[25] Junnan Li, Caiming Xiong, and Steven CH Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9475–9484, 2021. 1, 2, 4, 5, 8

[26] Bin Liu, Zhirong Wu, Han Hu, and Stephen Lin. Deep metric transfer for label propagation with limited annotated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 1

[27] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv: Learning*, 2017. 4

[28] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining, 2018. 1

[29] Geoffrey J McLachlan. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, 70(350):365–369, 1975. 8

[30] 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*, 41:1979–1993, 2019. 8

[31] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1094–1103, 2021. 1

[32] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 4

[33] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964. 4

[34] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 1

[35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020. 1

[36] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020. 7, 8

[37] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *WACV/MOTION*, pages 29–36, 2005. 1

[38] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, 1:29–36, 2005. 8

[39] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems*, 29:1163–1171, 2016. 8

[40] Mehdi S. M. Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *NIPS*, 2016. 8

[41] Henry Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965. 8

[42] Richard Socher, Yoshua Bengio, and Christopher D Manning. Deep learning for nlp (without magic). In *Tutorial Abstracts of ACL 2012*, pages 5–5. 2012. 1

[43] 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. 1, 2, 4, 5, 7, 8

[44] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013. 4

[45] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*, 2017. 1

[46] 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*, pages 10857–10866, 2021. 7, 8

[47] Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation for consistency training. *arXiv: Learning*, 2020. 2, 4, 5, 7, 8

[48] Qizhe Xie, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. Self-training with noisy student improves imagenet classification. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2020. 1

[49] Yi Xu, Jiandong Ding, Lu Zhang, and Shuigeng Zhou. Dpssl: Towards robust semi-supervised learning with a few labeled samples. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[50] Yao Yao, Junyi Shen, Jin Xu, Bin Zhong, and Li Xiao. Cls: Cross labeling supervision for semi-supervised learning. *arXiv preprint arXiv:2202.08502*, 2022. 2

[51] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *ArXiv*, abs/1605.07146, 2016. 5

[52] 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 Pro-*

cessing Systems, 34:18408–18419, 2021. 1, 2, 4, 5, 6, 7, 8

[53] Zhen Zhao, Luping Zhou, Lei Wang, Yinghuan Shi, and Yang Gao. Lassl: Label-guided self-training for semi-supervised learning. 2022. 2

[54] 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. 2, 4, 5, 8