Debiased Pseudo Labeling in Self-Training

Baixu Chen [*1] **Junguang Jiang** [*1] **Ximei Wang** [1] **Jianmin Wang** [1] **Mingsheng Long** [1]

Abstract

Deep neural networks achieve remarkable performances on a wide range of tasks with the aid of large-scale labeled datasets. However, large-scale annotations are time-consuming and labor-exhaustive to obtain on realistic tasks. To mitigate the requirement for labeled data, self-training is widely used in both academia and industry by pseudo labeling on readily-available unlabeled data. Despite its popularity, pseudo labeling is well-believed to be unreliable and often leads to training instability. Our experimental studies further reveal that the performance of self-training is biased due to data sampling, pre-trained models, and training strategies, especially the inappropriate utilization of pseudo labels. To this end, we propose Debiased, in which the generation and utilization of pseudo labels are decoupled by two independent heads. To further improve the quality of pseudo labels, we introduce a worst-case estimation of pseudo labeling and seamlessly optimize the representations to avoid the worst-case. Extensive experiments justify that the proposed Debiased not only yields an average improvement of 14.4% against state-of-the-art algorithms on 11 tasks (covering generic object recognition, fine-grained object recognition, texture classification, and scene classification) but also helps stabilize training and balance performance across classes.

1. Introduction

Deep learning has achieved great success in many machine learning problems in the past decades, especially where large-scale labeled datasets are present. In real-world applications, however, manually labeling sufficient data is time-consuming and labor-exhaustive. To reduce the need for labeled data, great effort (Grandvalet & Bengio, 2005; Lee, 2013; Tarvainen & Valpola, 2017; Chen et al., 2020a) has been paid to improve the data efficiency of deep neural networks by learning from a few labeled samples and a large number of unlabeled samples. Among them, self-training is an effective approach to deal with the lack of labeled data. Typical self-training methods (Lee, 2013; Sohn et al., 2020) assign pseudo labels to unlabeled samples with the model's own predictions and then train the model with these pseudo labeled samples as if they were labeled examples, whose effectiveness has been proved both theoretically (Wei et al., 2021) and empirically (Sohn et al., 2020).

Although self-training has achieved great advances in benchmark datasets, they still exhibit large training instability and extreme performance imbalance across classes. Figure 1 shows that the accuracy of FixMatch (Sohn et al., 2020), one of the state-of-the-art self-training methods, fluctuates greatly when trained *from scratch*. Though its performance will gradually recover after a sudden sharp drop, this is still not expected, since in most real-world applications, *pre-trained* models are more often adopted, and the performance of pre-trained models is difficult to recover after a drastic decline. Besides, although FixMatch improves the average accuracy, it also leads to the *Matthew effect*, *i.e.*, "the rich get richer and the poor get poorer". As shown in Figure 2, the accuracy of well-behaved categories is further increased while that of bad-behaved ones is decreased to nearly zero, which is also not expected, since most machine learning models prefer performance balance across categories, even when the class imbalance exists in the training data (Zhou et al., 2018). The above phenomena are caused by the *bias* between the pseudo labeling function with the unknown target labeling function. Training with biased and unreliable pseudo labels has the chance to accumulate errors and ultimately lead to performance fluctuations. And for those poor-behaved categories, the bias of the pseudo labels is worse and will be further enhanced as self-training progresses, ultimately leading to the Matthew effect.

To escape from the dilemma, we first delved into bias issues arising from the pseudo labeling process and found that they can be briefly grouped into three kinds: (1) *Data bias* which is caused by the scarcity of labeled data and the random quality of the sampled data points; (2) *Model bias* which means the model preference for different categories due to different pre-training methods; (3) *Training bias* which

[*]Equal contribution [1]School of Software, BNRist, Tsinghua University, China. Correspondence to: Mingsheng Long <mingsheng@tsinghua.edu.cn>.

Preprint.

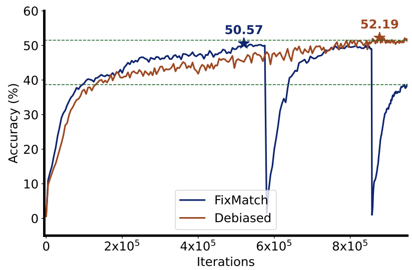


Figure 1. Top-1 accuracy when training from scratch on *CIFAR-100*. FixMatch causes the accuracy to fluctuate wildly while our Debiased method effectively improves the training stability and test accuracy.

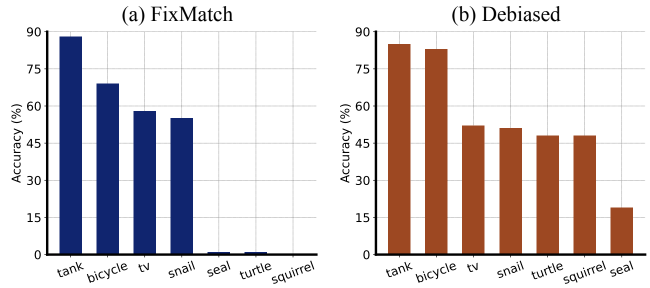


Figure 2. Top-1 accuracy of 7 randomly selected categories on *CIFAR-100*. FixMatch increases the accuracy of well-behaved categories while decreasing the accuracy of poor-behaved categories. In contrast, our Debiased method effectively balances the performance between different categories.

refers to the bias increment brought by the inappropriate utilization of pseudo labels for training. Note that the data bias and model bias are inherent in the problem. When they are larger, the training bias will also be larger and when they are zero, *i.e.*, the pseudo labeling is almost correct, training bias will also no longer exist.

Further, we present *Debiased*, a novel approach to decrease the above bias in the pseudo labeling process. Specifically, to eliminate the *training bias*, the classifier head is only trained with clean labeled samples and no longer trained with the unreliable pseudo-labeled samples, which ensures that pseudo labeling biases will not scale up with training. To optimize the feature generator with unlabeled data for better representation, we introduce a parameter-independent agent classifier head, through which the gradients on pseudo labeled samples are backpropagated to the feature generator. By decoupling the generation and utilization of pseudo labels with two independent heads, this mechanism can effectively prevent bias accumulation and thus greatly boost the model's tolerance to biased pseudo labels. Further, to decrease the *data bias* and *model bias* which cannot be calculated directly, we turn to estimate the worst case of training bias that implicitly reflects the data bias and model bias, and optimize the representations to decrease the worst-case bias and thereby improve the quality of pseudo labels.

The contributions of this work are summarized as three-fold:

- We systematically analyze the problem and the causes of pseudo labeling bias in self-training.
- We propose *Debiased*, a novel approach to mitigate the bias of pseudo labeling and boost the stability and performance balance across classes, as well as a universal add-on for different self-training methods.
- We conduct extensive experiments and validate that *Debiased* outperforms state-of-the-art algorithms on 11 self-training tasks by large margins, 14.4% on average.

2. Related Work

Self-training (Yarowsky, 1995; Rosenberg et al., 2005; Grandvalet & Bengio, 2005; Lee, 2013) is a widely-used approach to utilize unlabeled data. Among self-training techniques, Pseudo Label (Lee, 2013) is one of the most popular forms by leveraging the model itself to obtain proxy labels for unlabeled data. However, this paradigm still suffers from the problem of confirmation bias (Arazo et al., 2020), where the learner struggles to correct its own mistakes when learning from inaccurate pseudo labels. Recent works mainly tackle this issue from the following two aspects.

Generate higher-quality pseudo labels. Existing methods select samples with confidence threshold (Lee, 2013; Xu et al., 2021) or prediction uncertainty (Rizve et al., 2021), assign pseudo labels with the density of local neighborhood (Shi et al., 2018; Iscen et al., 2019), meta-learning (Pham et al., 2021), or simply use the predictions from a weaker augmented images (Sohn et al., 2020) or the average predictions from multiple augmentations (Berthelot et al., 2019) to generate more reliable pseudo labels. Rather than designing pseudo labeling manually, our method optimizes the representation to avoid the worst-case of pseudo labeling, and thereby improves its quality in a *data-driven* manner.

Improve tolerance with inaccurate pseudo labels. To mitigate the error accumulations, existing methods maintain a mismatch between the generation and utilization of pseudo labels, such as generating pseudo labels from the average of previous predictions (Laine & Aila, 2017), from an exponential moving average of the model (Tarvainen & Valpola, 2017), from a fixed teacher of the previous round (Xie et al., 2020b), or from another model in an online mutual-teaching manner (Ge et al., 2020). However, the generated pseudo labels still have an impact on the pseudo labeling itself, and the bias is still easily increased when self-training with unreliable pseudo labels. In contrast, our method decouples the generation and utilization of pseudo labels with completely independent classifier heads.

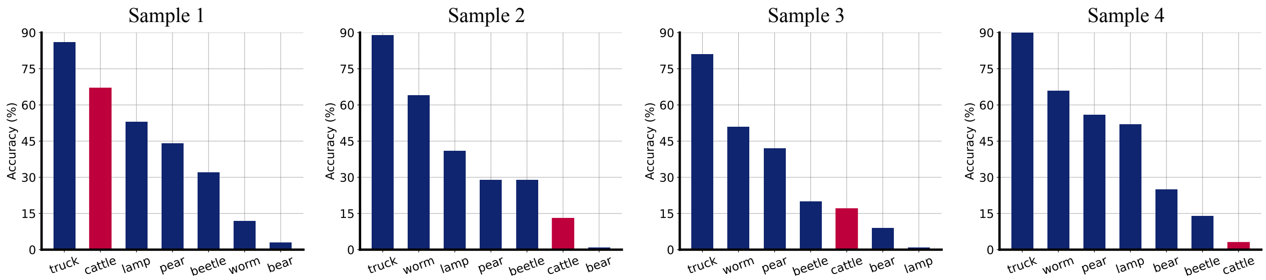


Figure 3. Data bias. Top-1 accuracy of 7 randomly selected categories when trained with different labeled data sampled from *CIFAR-100* (400 images). The same category (such as cattle) may have completely different accuracy in different samples. Following Sohn et al. (2020), 4 labeled data are sampled for each category by default in our analysis.

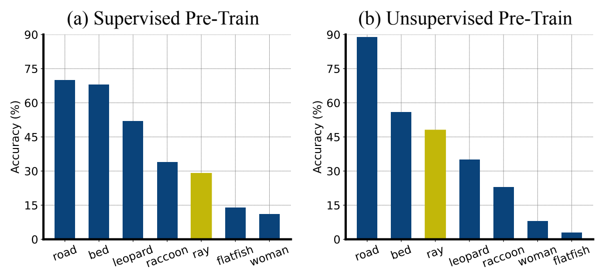


Figure 4. Model bias. Top-1 accuracy of 7 randomly selected categories with different pre-trained models on *CIFAR-100*. Different pre-trained models show different category preferences.

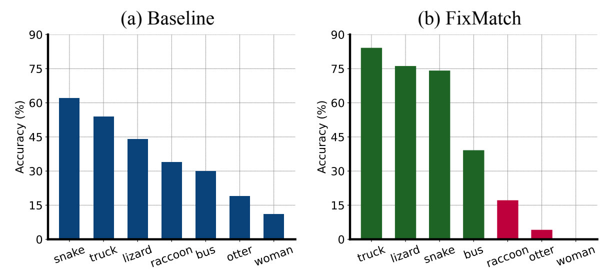


Figure 5. Training bias. Top-1 accuracy of 7 randomly selected categories with different training methods on *CIFAR-100*. FixMatch largely increases the bias of poor-behaved categories.

Recently, self-supervised methods (Devlin et al., 2019; He et al., 2020) are also used on unlabeled data to improve the model performance with few labeled samples, either in the pre-training stage (Chen et al., 2020a), or in the downstream tasks (Wang et al., 2021). However, the training of self-supervision usually relies on big data and heavy computation, which is not feasible in most applications. Besides, although these methods avoid the use of unreliable pseudo labels, it is difficult for them to learn task-specific information from unlabeled data for better performance.

3. Analysis of Bias in Pseudo Labeling

In this section, we provide some analysis on where the bias in the pseudo labeling comes from. The bias in the pseudo labeling refers to *the deviation between the learned decision hyperplanes and the true decision hyperplanes*. It can be measured by the accuracy of each category approximately, since the more biased the decision hyperplanes, the lower the accuracy of related categories. By analyzing the accuracy of different categories under different training conditions, we have the following findings.

Different random seeds will not influence the bias of pseudo labeling. When other conditions remain unchanged, the categories with high accuracy and the categories with low accuracy are determined, *i.e.*, the bias of the pseudo labeling is not random. Thus, our following analysis makes sense.

The sampling of labeled data will largely influence the bias of pseudo labeling. As shown in Figure 3, when the data sampling is different, the accuracy of the same category may be very high or very low. The reason is that the distances between different data points and the true decision hyperplanes are not the same, with some supporting data points closer and others far away. When there are few labeled data, there may be a big difference in the distances between supporting data of each category and the true decision hyperplanes, thus the learned decision hyperplanes will be biased towards some categories, which we call *data bias*.

The pre-training of models also affects the bias of pseudo labeling. Pre-training has become a common paradigm in many applications of deep learning (Devlin et al., 2019), and also one of the most practical solutions to the problem of data scarcity (Chen et al., 2020a; Jiang et al., 2022). Figure 4 shows that different pre-trained models lead to different category bias, even if the pre-trained dataset and the downstream labeled dataset are both identical. One possible reason is that the representations learned by different pre-trained models focus on different aspects of the data (Zhao et al., 2021). Therefore, the same data could also have different distances to the decision hyperplanes in the representation level with different pre-trained models. And we call the bias of learned decision hyperplanes brought by pre-training models *model bias*.

Training with pseudo labels in turn enlarges the bias of pseudo labeling. Figure 5 shows that the model bias and data bias exist but are limited when no pseudo labels are used. After training with pseudo labels (*e.g.*, using FixMatch), the performance gap for different categories greatly enlarges, with the accuracy of some categories increasing from 60% to 80% and that of some categories dropping from 15% to 0%. The reason is that for well-behaved categories, the pseudo labels are almost accurate, thus using them for training could further reduce the bias. Yet for many poorly-behaved categories, the pseudo labels are not reliable, and the common pseudo labeling mechanism will further increase the bias, and fail to correct it back in the follow-up training. The bias increased here is brought by unreasonable training strategies, thus we call it *training bias*.

Training bias is introduced by inappropriate utilization of pseudo labels, thus can be eliminated, while data bias and model bias are inherent in the problem, thus can only be reduced. Next we will mention how to decrease these biases.

4. Debiased Pseudo Labeling

4.1. Background

Assume we have a labeled dataset $\mathcal{L} = \{(\mathbf{x}_i^l, y_i^l\}_{i=1}^{n_l}$ of n_l labeled samples and an unlabeled dataset $\mathcal{U} = \{(\mathbf{x}_j^u)\}_{j=1}^{n_u}$ of n_u unlabeled samples, where the size of the labeled dataset is usually much smaller than that of the unlabeled dataset, *i.e.*, $n_l \ll n_u$. Denote ψ the feature generator, and h the task-specific head. The standard cross-entropy loss on weakly augmented labeled examples is as follows

$$L_{\mathcal{L}}(\psi, h) = \frac{1}{n_l} \sum_{i=1}^{n_l} L_{\text{CE}}\big((h \circ \psi \circ \alpha)(\mathbf{x}_i^l), y_i^l\big), \quad (1)$$

where α is the weak augmentation function. Since there are few labeled samples, the feature generator and the task-specific head will easily over-fit, and typical self-training methods use these pseudo labels on plenty of unlabeled data to decrease the generalization error. Different self-training methods design different pseudo labeling function $\widehat{f}$ (Lee, 2013; Xu et al., 2021; Rizve et al., 2021). Take FixMatch (Sohn et al., 2020) for an instance. FixMatch first generates predictions $\widehat{\mathbf{p}} = (h \circ \psi \circ \alpha)(\mathbf{x})$ on a weakly-augmented version of given unlabeled images, and adopts a confidence threshold τ mechanism to filter out unreliable pseudo labels

$$\widehat{f}_{\psi,h}(\mathbf{x}) = \begin{cases} \arg\max \widehat{\mathbf{p}}, & \max \widehat{\mathbf{p}} \geq \tau, \\ -1, & \text{otherwise}, \end{cases} \quad (2)$$

where $\widehat{f}_{\psi,h}$ refers to the pseudo labeling by model $h \circ \psi$, hyperparameter τ specifies the threshold above which a pseudo label is retained and -1 indicates that this pseudo label is omitted in training. Then FixMatch uses the pseudo labels to train on the strongly-augmented unlabeled images,

$$L_{\mathcal{U}}(\psi, h, \widehat{f}) = \frac{1}{n_u} \sum_{j=1}^{n_u} L_{\text{CE}}\big((h \circ \psi \circ \mathcal{A})(\mathbf{x}_j^u), \widehat{f}(\mathbf{x}_j^u)\big), \quad (3)$$

where $\widehat{f}$ is a notation of general pseudo labeling function and $\mathcal{A}$ is the strong augmentation function. As shown in Figure 6(a), the optimization objective for FixMatch is

$$\min_{\psi,h} L_{\mathcal{L}}(\psi, h) + \lambda L_{\mathcal{U}}(\psi, h, \widehat{f}_{\psi,h}), \quad (4)$$

where λ is the trade-off between the loss on the labeled data and that on the unlabeled data. FixMatch removes low-confidence samples during the pseudo labeling process, yet it still fails to debias the pseudo labeling process.

The issues of typical pseudo labeling methods come from two aspects. (1) The pseudo labels are generated and used by the same model or related models, which leads to the training bias, *i.e.*, the errors of the model might be amplified as the self-training progresses. (2) When trained with few labeled samples, the problem of unreliable pseudo labeling caused by data bias and model bias cannot be ignored anymore, yet existing methods have no explicit designs to reduce these biases. To tackle the above issues, we propose two important designs to debias the utilization and generation of pseudo labeling in Section 4.2 and 4.3 respectively.

4.2. Generate and utilize pseudo labels independently

The confirmation bias of FixMatch stems from the way of training on the pseudo labels generated by itself. To alleviate this bias, some methods turn to generate pseudo labels from a better teacher model, such as the moving average of the original model in Figure 6(b) (Tarvainen & Valpola, 2017) or simply the model obtained from the previous round of training in Figure 6(c) (Xie et al., 2020b), and then utilize these pseudo labels to train both the feature generator ψ and the task-specific h. However, there is still a relationship between the teacher model that generates the pseudo labels and the student model that utilizes the pseudo labels in the above methods, and the decision hyperplanes of the student model $h \circ \psi$ still depend on the biased pseudo labeling $\widehat{f}$. Thus training bias still exists in the self-training process.

To get rid of the training bias when utilizing the pseudo labels, we optimize the task-specific head h, only with the clean labels on $\mathcal{L}$ and without any unreliable pseudo labels from $\mathcal{U}$. To prevent the deep models from over-fitting to the few labeled samples, we still use pseudo labels, but only for learning a better representation. As shown in Figure 6(d), we introduce an agent head h_{agent}, which is connected to

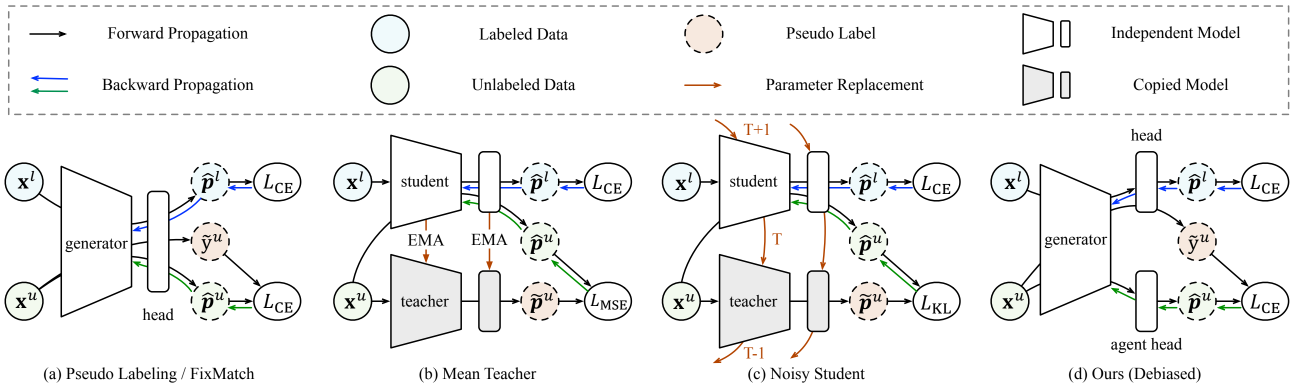


Figure 6. Comparisons on how different self-training methods generate and utilize pseudo labels. **(a)** Pseudo Labeling and FixMatch generate and utilize pseudo labels on the same model. **(b)** Mean Teacher generates pseudo labels from the Exponential Moving Average (EMA) of the current model. **(c)** Noisy Student generates pseudo labels from the teacher model which is obtained from the previous round of training. **(d)** Debiased generates pseudo labels from head h and utilizes pseudo labels on a completely independent agent head h_{agent}.

the feature generator ψ and only optimized with the pseudo labels from $\mathcal{U}$. The full optimization objective is

$$\min_{\psi,h,h_{\text{agent}}} L_{\mathcal{L}}(\psi,h) + \lambda L_{\mathcal{U}}(\psi,h_{\text{agent}},\widehat{f}_{\psi,h}), \tag{5}$$

where the pseudo labels are generated by head h and utilized by a completely *independent* agent head h_{agent}. This mechanism of separation can effectively prevent the bias accumulation brought by self-training on head h in Equation 4. Note that the agent heat h_{agent} is only responsible for gradient backpropagation to the feature generator ψ during training and will be discarded during inference, and thus will introduce no inference cost.

4.3. Avoid generation of erroneous pseudo labels

Section 4.2 presents a solution to eliminating the training bias, yet the data bias and model bias still exist in the pseudo labeling $\widehat{f}$. As shown in Figure 7(a), due to the data bias and model bias, labeled samples of each class have different distances to the decision hyperplanes in the representation space, which leads to a deviation between the learned hyperplanes and the real decision hyperplanes, especially when the size of labeled samples is very small. As a result, pseudo labeling $\widehat{f}$ is very likely to generate incorrect pseudo labels on unlabeled data points that are close to these biased decision hyperplanes. And our objective now is to optimize the feature representations to reduce the data bias and model bias, and finally improve the quality of pseudo labels.

Since we have no labels for $\mathcal{U}$, we cannot directly measure and thereby reduce data bias and model bias. Yet training bias has some correlations with data bias and model bias. Recall in Section 4.2, the task-specific head h is only optimized with clean labeled data, since optimization with incorrect pseudo labels will push the learned hyperplanes in a more biased direction and lead to the training bias. Therefore, training bias can be considered as the accumulation of data bias and model bias with inappropriate utilization of pseudo labels, which is training algorithm dependent. And the worst training bias that can be achieved among all the training methods can better measure the degree of model bias and data bias. By decreasing the worst training bias, we can indirectly decrease the model bias and data bias. Specifically, the worst training bias corresponds to the worst possible head h' learned by pseudo labeling, such that h' predicts correctly on all the labeled samples $\mathcal{L}$ while making as many mistakes as possible on the unlabeled data $\mathcal{U}$,

$$h_{\text{worst}}(\psi) = \arg\max_{h'} L_{\mathcal{U}}(\psi,h',\widehat{f}_{\psi,h}) - L_{\mathcal{L}}(\psi,h'), \tag{6}$$

where the mistakes of h' on unlabeled data are estimated by its discrepancy with the current pseudo labeling function $\widehat{f}$. Equation 6 measures the worst-case of task-specific head h that might be learned in the future when trained with pseudo labeling on the current feature generator ψ. It is also the *worst hyperplanes* as shown in Figure 7(b), which deviates as much as possible from the currently learned hyperplanes while ensuring that all labeled samples are correctly distinguished. Note that h_{worst} depends on the feature representations generated by ψ, thus we can optimize feature generator ψ to decrease the worst-case bias,

$$\min_{\psi} L_{\mathcal{U}}(\psi,h_{\text{worst}}(\psi),\widehat{f}_{\psi,h}) - L_{\mathcal{L}}(\psi,h_{\text{worst}}(\psi)). \tag{7}$$

As shown in Figure 7(c), Equation 7 encourages the feature of unlabeled samples to be distinguished correctly even by the worst hyperplanes, i.e., be generated far away from the current hyperplanes, thereby reducing the model bias and data bias in the feature representations.

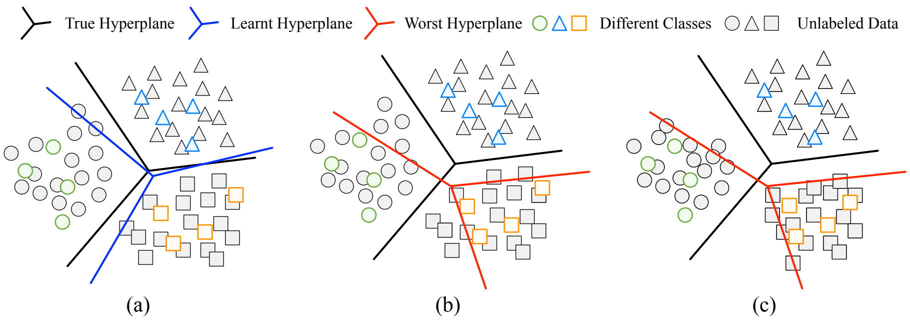


Figure 7. Concept explanations. **(a)** Bias between the hyperplanes learned on limited labeled data and the true hyperplanes. **(b)** The worst hyperplanes are hyperplanes that correctly distinguish the labeled samples while making as many mistakes as possible on the unlabeled samples. **(c)** Feature representations are optimized to improve the performance of the worst hyperplanes.

5. Experiments

We evaluate the proposed Debiased on 11 tasks, covering a wide range of datasets, including **(1)** superordinate-level object classification: *CIFAR-10* (Krizhevsky et al., 2009), *CIFAR-100* (Krizhevsky et al., 2009), *Caltech-101* (Fei-Fei et al., 2004); **(2)** fine-grained object classification: *Food-101* (Bossard et al., 2014), *CUB-200-2011* (Wah et al., 2011), *Stanford Cars* (Krause et al., 2013), *FGVC Aircraft* (Maji et al., 2013), *OxfordIIIT Pets* (Parkhi et al., 2012), *Oxford Flowers* (Nilsback & Zisserman, 2008); **(3)** texture classification: *DTD* (Cimpoi et al., 2014); **(4)** scene classification: *SUN397* (Xiao et al., 2010). The complete training dataset size ranges from $2,040$ to $75,750$ and the number of classes ranges from 10 to 397. Following Kornblith et al. (2019), we report mean accuracy per-class on *Caltech-101*, *FGVC Aircraft*, *OxfordIIIT Pets*, *Oxford Flowers*, and top-1 accuracy for other datasets. Following Sohn et al. (2020), we construct a labeled subset with 4 or 10 labels per category. To make a fair comparison, we keep the labeled subset for each dataset the same throughout our experiments.

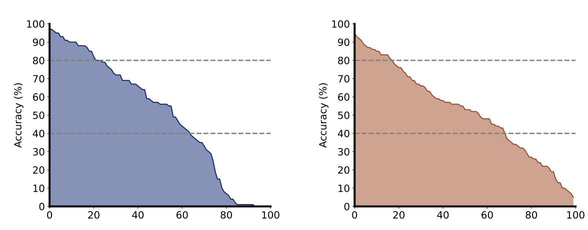
For experiments *without* pre-trained models, we follow Sohn et al. (2020) and adopt Wide ResNet-28-8 (Zagoruyko & Komodakis, 2016) with an input size of 32×32. For experiments *with* pre-trained models, we adopt ResNet50 (He et al., 2016) with an input size of 224×224 and pre-trained on ImageNet (Deng et al., 2009). We adopt MoCo v2 (Chen et al., 2020b) as unsupervised pre-trained models. We compare our method with many state-of-the-art self-training methods, including Pseudo Label (Lee, 2013), Π-Model (Laine & Aila, 2017), Mean Teacher (Tarvainen & Valpola, 2017), UDA (Xie et al., 2020a), FixMatch (Sohn et al., 2020), and Self-Tuning (Wang et al., 2021).

When training from scratch, we adopt the same hyper-parameters as FixMatch (Sohn et al., 2020), with learning rate of 0.03, mini-batch size of 512, weight-decay of 0.001. For other experiments, we use SGD with momentum 0.9 and weight-decay in $\{0.0005, 0.001\}$, learning rates in $\{0.001, 0.003, 0.01, 0.03\}$. The mini-batch size is set to 64 following Su et al. (2021). For each image, we first apply random-resize-crop and then use RandAugment (Cubuk et al., 2020) for strong augmentation $\mathcal{A}$ and random-horizontal-flip for weak augmentation α. The trade-off hyperparameter λ is set to 1 for all datasets. More details on hyperparameter selection can be found in Appendix A.2. Each experiment is repeated three times.

As suggested by Oliver et al. (2018), we reimplement all baselines and perform all experiments using the same codebase. **We will release the codebase for all the methods.**

5.1. Main results

Train from scratch. Figure 1 compares the validation error of FixMatch and Debiased during the training procedure on *CIFAR-100*. We observe that the performance of FixMatch suffers from significant fluctuations during training. In contrast, the accuracy of Debiased increases steadily and surpasses the best accuracy of FixMatch by 3.2%, relatively. Figure 8 compares the per-category accuracy of FixMatch and Debiased on *CIFAR-100*. FixMatch leads to severely imbalanced accuracy across different categories, where the accuracy almost drops to 0 for over 15% categories. The proposed Debiased successfully alleviates these issues and improves the accuracy of those poor-behaved categories.



(a) FixMatch (b) Debiased

Figure 8. The accuracy for each category in descending order on *CIFAR-100* (Wide ResNet-28-8, train from scratch).

Table 1. Comparison between Debiased and various baselines (ResNet50, supervised and unsupervised pre-trained). Δ indicates the absolute *improvement over state-of-the-art* on each task. ↓ indicates a performance degradation compared with the baseline.

		Caltech101	CIFAR10	CIFAR100	SUN397	DTD	Aircraft	CUB	Flowers	Pets	Cars	Food101	Average
Supervised	Baseline	81.4	65.2	48.2	39.9	47.7	25.4	46.5	85.2	78.1	33.3	33.8	53.2
	Pseudo Label (2013)	86.3	83.3	54.7	41.0	50.2	27.2	54.3	92.3	87.8	41.4	38.0	59.7
	Π-Model (2017)	83.5	73.1	49.2	39.7↓	50.3	24.3↓	47.1	90.7	82.2	30.9	33.9	55.0
	Mean Teacher (2017)	83.7	82.1	56.0	37.9↓	51.6	30.7	49.6	91.0	82.8	39.1	40.3	58.6
	UDA (2020a)	85.8	83.6	54.7	41.3	49.0	27.1	52.1	92.0	83.1	45.6	41.7	59.6
	FixMatch (2020)	86.3	84.6	53.1	41.3	48.6	25.2↓	52.3	93.2	83.7	46.4	37.1	59.3
	Self-Tuning (2021)	87.2	76.0	57.1	41.8	50.7	35.2	58.9	92.6	86.6	58.3	41.9	62.4
	Debiased (Ours)	**89.6**	**94.9**	**70.4**	**47.3**	**53.5**	**43.2**	**68.7**	**94.8**	**89.8**	**71.0**	**58.5**	**71.1**
	Δ	2.4	10.3	13.3	5.5	1.9	8.0	9.8	1.6	2.0	12.7	16.6	8.7
Unsupervised	Baseline	79.5	66.6	46.5	38.1	47.9	28.7	37.5	87.7	60.0	38.1	32.9	51.2
	Pseudo Label (2013)	86.2	70.8	49.8	38.6	50.0	26.6↓	41.8	93.0	68.4	37.3↓	32.8↓	54.1
	Π-Model (2017)	80.1	76.2	44.8	37.8↓	50.0	23.5↓	31.6↓	93.1	62.8	25.6↓	30.4↓	50.5
	Mean Teacher (2017)	80.4	80.8	51.3	34.2↓	48.8	33.8	41.6	92.9	67.0	50.5	39.1	56.4
	UDA (2020a)	85.0	87.4	53.6	42.3	46.2↓	35.7	41.4	94.1	69.3	51.5	39.3	58.7
	FixMatch (2020)	83.1	82.2	49.3	39.2	43.9↓	30.1	36.8↓	94.3	65.7	48.6	36.8	55.5
	Self-Tuning (2021)	81.6	51.5↓	44.7↓	35.1↓	45.5↓	31.4	41.6	91.0	66.9	52.0	31.9↓	52.1
	Debiased (Ours)	**89.2**	**95.2**	**66.0**	**46.6**	**50.9**	**39.9**	**52.4**	**94.4**	**75.4**	**64.2**	**54.8**	**66.3**
	Δ	3.0	7.8	12.4	4.3	0.9	4.2	10.6	0.1	6.1	12.2	15.5	7.6

Supervisied pre-train. Table 1 reveals that current self-training methods lead to relatively mild improvements with supervised pre-trained models, which is consistent with previous findings (Su et al., 2021; Wang et al., 2021). In contrast, the proposed Debiased significantly boosts the performance and outperforms the state-of-the-art method by relatively **14.5%** on all 11 datasets on average. Besides, for datasets that have larger domain shift with *ImageNet*, e.g., *SUN397*, the pseudo labels are more error-prone. Thus, existing self-training methods only increase the accuracy within 2%, while Debiased yields a decent accuracy gain of 7.4%. With a pre-trained model, self-training has better training stability. Yet once the performance degradation occurs, the process is also irreversible (Appendix B.1), partly due to the catastrophic forgetting of pre-trained representation caused by the biased pseudo labels. Also, self-training suffers from more severe performance imbalance across classes (Appendix B.2). Debiased effectively tackles these issues, indicating the importance of reducing bias.

Unsupervised pre-train. Table 1 shows that with unsupervised pre-trained models, previous methods suffer from performance degradation after self-training on the unlabeled data. (Each method suffers on about 3 datasets on average.) The difficulty comes from that unsupervised pre-training task has larger task discrepancy with the downstream classification tasks than supervised pre-training task. Thus, the representations learned by unsupervised pre-trained models usually exhibit stronger model bias, and inappropriate usage of pseudo labels will lead to rapid accumulation errors and increase the training bias. By eliminating training bias and reducing model bias, Debiased brings improvement on all datasets and outperforms the state-of-the-art method by **14.3%** on all 11 datasets on average, relatively.

5.2. Ablation studies

We examine the design of our method on *CIFAR-100* in Table 2 and have the following findings. **(1)** Compared with *Mutual Learning* (Zhang et al., 2018; Ge et al., 2020), where two heads provide pseudo labels to each other, the independent mechanism in our method where one head is only responsible for generating pseudo labels and the other head only uses them can better reduce the training bias. **(2)** A nonlinear agent head is always better than a linear agent head. We conjecture that nonlinear projection can reduce the degeneration of representation with biased pseudo labels. **(3)** The worst-case estimation of pseudo labeling improves the performance by large margins, especially when the labeled samples are extremely scarce and the data bias is severe.

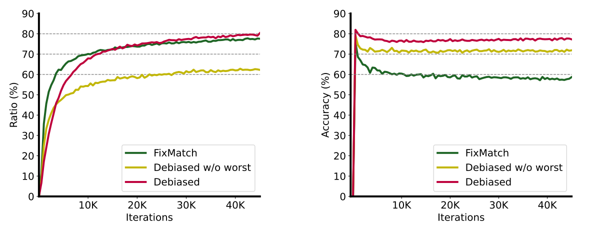
5.3. Analysis

To further investigate how Debiased improves pseudo labeling and self-training performance, we conduct some analysis on *CIFAR-100*. For simplicity, we only give the results with supervised pre-trained models. Results with other models can be found in Appendix B.4.

Debiased improves both the quantity and quality of

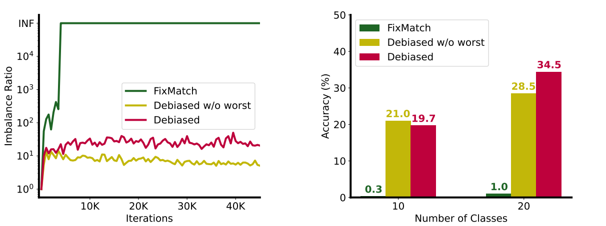
Table 2. Ablation study on *CIFAR-100* with different label proportions and different pre-trained models (ResNet50).

Method	Multiple Heads	Linear Agent Head	Nonlinear Agent Head	Worst Case Estimation	Supervised Pre-Train 400 labels	Supervised Pre-Train 1000 labels	Unsupervised Pre-Train 400 labels	Unsupervised Pre-Train 1000 labels
FixMatch					53.1	67.8	49.3	64.2
Mutual Learning	✔				53.4	68.5	50.5	65.9
Debiased w/o worst	✔	✔			56.7	67.4	57.0	68.4
Debiased w/o worst	✔		✔		58.0	67.8	58.5	69.3
Debiased	✔		✔	✔	**70.4**	**76.1**	**66.0**	**73.4**



(a) Quantity of pseudo labels (b) Quality of pseudo labels

Figure 9. The quantity and quality of pseudo labels during training on *CIFAR-100* (ResNet50, supervised pre-trained).



(a) Quantity of pseudo labels (b) Quality of pseudo labels

Figure 10. Analysis of pseudo labels for poor-behaved categories on *CIFAR-100* (ResNet50, supervised pre-trained). **(a)** The change in class imbalance ratio during self-training. **(b)** The average accuracy of five or ten worst-behaved categories.

pseudo labels. As shown in Figure 9(a) and 9(b), FixMatch exploits unlabeled data *aggressively*, on average producing more than 70% pseudo labels during training. But the cost is that the accuracy of pseudo labels continues to drop, eventually falling below 60%, which is consistent with our motivation in Section 3 that inappropriate utilization of pseudo labels will in turn enlarges the training bias. On the contrary, the accuracy of pseudo labels in Debiased suffers from a smaller drop. Rather, it keeps rising afterward and exceeds 70% throughout the training. Besides, Debiased generates more pseudo labels in the later stages of training.

Debiased generates better pseudo labels for poor-behaved categories. To measure the quantity of pseudo labels on poor-behaved categories, we calculate the class imbalance ratio I on a class-balanced validation set, $I = \max_c N(c)/\min_{c'} N(c')$, where $N(c)$ denotes the number of predictions that fall into category c. As shown in Figure 10(a), the class imbalance ratio of FixMatch rises rapidly and reaches infinity after 5000 iterations, indicating that the model completely ignores those poorly-learned categories.

Table 3. Debiased as a general add-on to three mainstream self-training methods on *CIFAR-100* (ResNet50).

Pre-training		Supervised		Unsupervised	
Label Amount		400	1000	400	1000
Mean Teacher	Biased	56.0	67.0	51.3	63.5
	Debiased	**62.7**	**70.7**	**60.7**	**69.3**
Noisy Student	Biased	50.3	63.4	52.4	62.8
	Debiased	**68.9**	**74.8**	**66.6**	**75.2**
FixMatch	Biased	53.1	67.8	49.3	64.2
	Debiased	**70.4**	**76.1**	**66.0**	**73.4**

To measure the quality of pseudo labels on poor-behaved categories, we calculate the average accuracy of 10 or 20 worst-behaved categories in Figure 10(b). The average accuracy on the worst 20 categories of FixMatch is only **1.0%**. By reducing training bias with the agent head, data bias and model bias with the worst-case estimation, the average accuracy balloons to **28.5%** and **34.5%**, respectively.

5.4. Debiased as a general add-on

To explore incorporating Debiased into different state-of-the-art self-training methods, we consider three mainstream paradigms of self-training shown in Figure 6, including FixMatch (Sohn et al., 2020), Mean Teacher (Tarvainen & Valpola, 2017) and Noisy Student (Xie et al., 2020b). Implementation details of Debiased versions of these methods can be found in Appendix A.3. Table 3 compares the original (Biased) and Debiased versions of these methods on *CIFAR-100* with both supervised pre-trained model and unsupervised pre-trained models. Results show that the proposed Debiased yields large improvement on all these self-training methods, indicating that training bias widely exists in existing self-training methods and Debiased can serve as a universal add-on to reduce pseudo labeling bias.

6. Conclusion

To mitigate the requirement for labeled data, pseudo labels are widely used on the unlabeled data, yet they suffer from severe confirmation bias. In this paper, we systematically delved into the bias issues and present *Debiased*, a novel approach to decrease bias in pseudo labeling. Experimentally, *Debiased* increases state-of-the-art algorithms by 14.4% on

11 tasks and can serve as a universal add-on.

References

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

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

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

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020a.

Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

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

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR*, 2020.

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

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

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

Ge, Y., Chen, D., and Li, H. Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. In *ICLR*, 2020.

Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. In *NeurIPS*, 2005.

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

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Iscen, A., Tolias, G., Avrithis, Y., and Chum, O. Label propagation for deep semi-supervised learning. In *CVPR*, 2019.

Jiang, J., Shu, Y., Wang, J., and Long, M. Transferability in deep learning: A survey, 2022.

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *CVPR*, 2019.

Krause, J., Deng, J., Stark, M., and Fei-Fei, L. Collecting a large-scale dataset of fine-grained cars. In *FGVC*, 2013.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

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

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

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

Oliver, A., Odena, A., Raffel, C., Cubuk, E. D., and Goodfellow, I. J. Realistic evaluation of deep semi-supervised learning algorithms. In *NeurIPS*, 2018.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. Cats and dogs. In *CVPR*, 2012.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

Pham, H., Dai, Z., Xie, Q., and Le, Q. V. Meta pseudo labels. In *CVPR*, 2021.

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

Rosenberg, C., Hebert, M., and Schneiderman, H. Semi-supervised self-training of object detection models. In *WACV*, 2005.

Shi, W., Gong, Y., Ding, C., Tao, Z. M., and Zheng, N. Transductive semi-supervised deep learning using min-max features. In *ECCV*, 2018.

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. In *ICML*, 2014.

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

Tarvainen, A. and Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017.

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

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

Wei, C., Shen, K., Chen, Y., and Ma, T. Theoretical analysis of self-training with deep networks on unlabeled data. In *ICLR*, 2021.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V. Unsupervised data augmentation for consistency training. In *NeurIPS*, 2020a.

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

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

Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL*, 1995.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *BMVC*, 2016.

Zhang, Y., Xiang, T., Hospedales, T. M., and Lu, H. Deep mutual learning. In *CVPR*, 2018.

Zhao, N., Wu, Z., Lau, R. W. H., and Lin, S. What makes instance discrimination good for transfer learning? In *ICLR*, 2021.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. In *PAMI*, 2018.

A. Implementation Details

Our code is based on PyTorch (Paszke et al., 2019). **We will release both the code for our method and that for all the baselines**. The following are the implementation details of our experiments.

A.1. Architecture

The architectures of different classifier heads are as follows. For nonlinear heads, we adopt Dropout (Srivastava et al., 2014) to alleviate over-fitting.

- Linear agent head: `Linear-Softmax`;
- Nonlinear agent head: `Linear-ReLU-Dropout-Linear-Softmax`;
- Worst-case head: `Linear-ReLU-Dropout-Linear-Softmax`.

A.2. Hyperparameters

For experiments *without* pre-trained models, we use the same hyperparameters as FixMatch (Sohn et al., 2020). Specifically, we use learning rate of 0.03, mini-batch size of 512 (64 for labeled data, 448 for unlabeled data), weight decay of 0.001, confidence threshold of 0.95, unlabeled loss weight $\lambda = 1.0$. For our method, we set the projection dimension of the agent head and worst head to 1024.

For experiments *with* pre-trained models, we use SGD with momentum of 0.9. We choose weight decay in $\{0.0005, 0.001\}$, learning rates in $\{0.001, 0.003, 0.01, 0.03\}$. We train for $40k$ iterations and use the cosine learning rate schedule. The mini-batch size is set to 64. Besides, we tune the following algorithm-specific hyperparameters.

Π-Model. We search unlabeled loss weight λ in $\{0.1, 0.3, 1.0, 3.0\}$, warm-up iterations of unlabeled loss in $\{5 \times 10^3, 10^4\}$.

Mean Teacher. We fix the exponential moving average hyperparameter α to 0.999. We search unlabeled loss weight λ in $\{0.1, 0.3, 1.0, 3.0\}$, warm-up iterations of unlabeled loss in $\{5 \times 10^3, 10^4\}$.

Pseudo Label. We search confidence threshold τ in $\{0.7, 0.8, 0.9, 0.95\}$, unlabeled loss weight λ in $\{0.1, 0.3, 1.0, 3.0\}$.

FixMatch. The same as Pseudo Label.

UDA. We search temperature in $\{0.1, 0.5, 1.0, 2.0\}$, unlabeled loss weight λ in $\{0.1, 0.3, 1.0, 3.0\}$.

Noisy Student. We search temperature in $\{0.1, 0.5, 1.0, 2.0\}$, unlabeled loss weight λ in $\{0.1, 0.3, 1.0, 3.0\}$. We iterate 3 rounds, excluding the round that trains with only labeled data. The final performance is reported.

Self-Tuning. We try queue size in $\{24, 32\}$ and use temperature 0.07, projection dimension 1024, same as the original paper.

Debiased. We set the confidence threshold to 0.7 by default. We fix the projection dimension of the agent head and the worst head to 2048. The trade-off hyperparameter λ is set to 1.

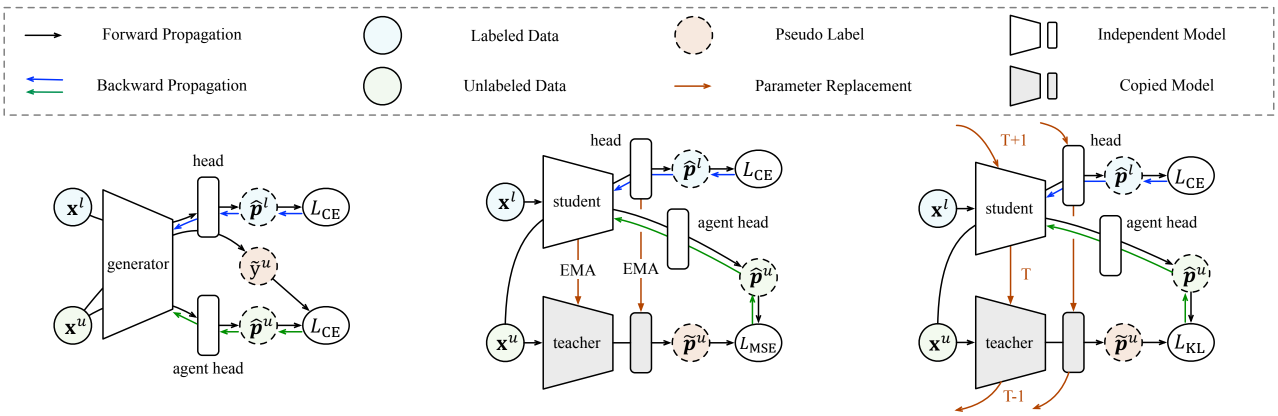
A.3. Debiased as a general add-on to previous self-training methods

In this section, we will illustrate how to incorporate Debiased into 3 typical self-training methods, including FixMatch, Mean Teacher, and Noisy Student. We will mainly focus on the modification to the generation and utilization of pseudo labels, and omit the introduction of the worst heads since they are the same across different self-training methods.

Debiased FixMatch. As shown in Figure 11(a), the pseudo labels on the unlabeled data are generated by the main head h and utilized by the agent head h_{agent}. The main head h is only trained on the clean labeled data.

Debiased Mean Teacher. As shown in Figure 11(b), the pseudo labels on the unlabeled data are generated by the exponential moving average of the main head h and utilized by the agent head h_{agent}. The main head h is only trained on the clean labeled data.

Debiased Noisy Student. As shown in Figure 11(c), the pseudo labels on the unlabeled data are generated by the head h of previous round $T-1$ and utilized by the agent head h_{agent}. The main head h is only trained on the clean labeled data.



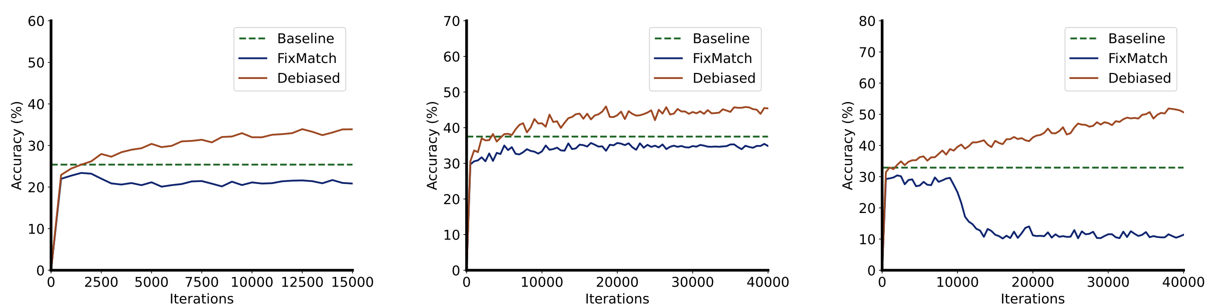
(a) Debiased FixMatch (b) Debiased Mean Teacher (c) Debiased Noisy Student

Figure 11. Illustrations on how different Debiased self-training methods generate and utilize pseudo labels.

B. More Experimental Results

B.1. Experiments on training stability

We further explore the training stability of FixMatch when using pre-trained models on various tasks. Figure 12 illustrates several failure cases of FixMatch.



(a) *Aircraft* (supervised pre-trained) (b) *CUB* (unsupervised pre-trained) (c) *Food-101* (unsupervised pre-trained)

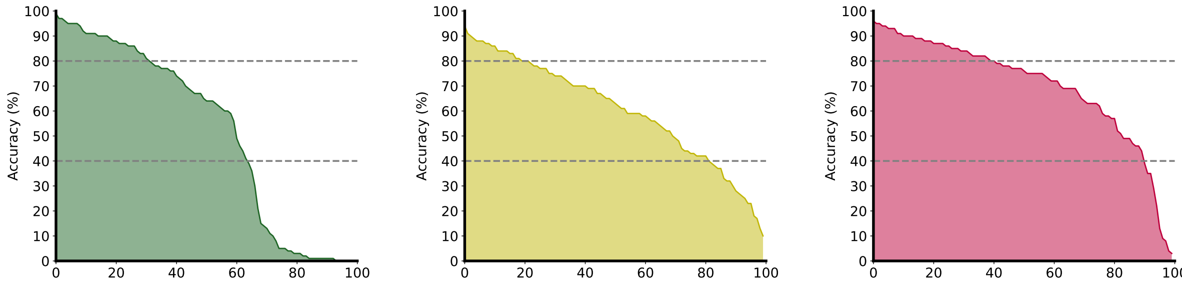
Figure 12. Failure cases of FixMatch with confidence threshold 0.7 (ResNet50).

Figures 12(a) and 12(b) show **when the performance of the pre-trained models declines, it cannot be recovered later**. Note that we try confidence threshold in $\{0.7, 0.8, 0.9, 0.95\}$ and get similar results.

Figure 12(c) demonstrates a complete failure case of FixMatch. With unsupervised pre-trained models and confidence threshold of 0.7, there can be a lot of noise in pseudo labels and thus the performance of FixMatch decline severely. Note this result is not the entry we report in Table 1 (threshold is set to 0.9 for this dataset). Instead, we aim to show **Debiased improves the training stability when there is much noise**.

B.2. Experiments on performance balance between categories

Figure 13 plots the top-1 accuracy of each category on *CIFAR-100* yielded by self-training on 400 labels and unlabeled data with supervised pre-trained models. The results are consistent with our previous analysis (Section 5.3) that **Debiased improves the performance of those poor-behaved categories**.



(a) FixMatch (b) Debiased w/o worst (c) Debiased

Figure 13. Top-1 accuracy of each category on *CIFAR-100* (ResNet50, supervised pre-trained).

B.3. Experiments with more labeled data

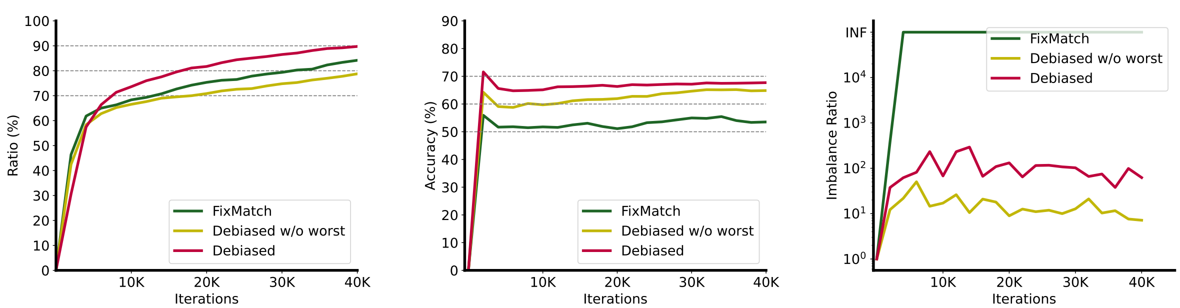
Table 4 reports the performance of Debiased with 1000 labels on *CIFAR-100* with different pre-trained models. Debiased yields over 10% improvements, relatively.

Table 4. Experiments with 10 labels per-class on *CIFAR-100* (ResNet50).

	Supervised Pre-Train	Unsupervised Pre-Train
Baseline	61.5	56.2
Pseudo Label	67.4	57.3
Π-Model	63.3	55.5
Mean Teacher	67.0	63.5
UDA	65.1	67.5
FixMatch	67.8	64.2
Self-Tuning	66.0	60.2
Debiased	**75.6**	**76.8**

B.4. Analysis on the behavior of pseudo labels

In this subsection, we explore the behavior of pseudo labels with unsupervised pre-trained models. Concretely, we focus on the quantity, accuracy as well as class imbalance ratio I of pseudo labels. Recall that $I = \max_c N(c)/\min_{c'} N(c')$, where $N(c)$ denotes the number of predictions that fall into category c. Figure 14 shows the results on *CIFAR-100* with 400 labels. We observe the same phenomenon that **Debiased effectively reduces the bias of pseudo labels, thus improving the self-training process**.



(a) Quantity (b) Quality (c) Quantity of poor-behaved categories

Figure 14. Analysis on the behavior of pseudo labels on *CIFAR-100* (ResNet50, unsupervised pre-trained). **(a)** The quantity of pseudo labels above the confidence threshold. **(b)** The accuracy of pseudo labels. **(c)** The class imbalance ratio I of pseudo labels.

B.5. Ablation study on nonlinear main classifier head

Experiments suggest that using a nonlinear agent head improves performance. We further explore how things are going for the main head. As shown in Table 5, **using nonlinear main head or not results in a similar performance on average**. We conjecture this is because a nonlinear main head is more likely to over-fit with few labeled samples.

Table 5. Ablation on nonlinear main head on *CIFAR-100* (FixMatch, ResNet50, supervised pre-trained).

Head	Supervised Pre-Train		Unsupervised Pre-Train	
	400 labels	1000 labels	400 labels	1000 labels
Linear	53.1	**67.8**	**49.3**	**64.2**
Nonlinear	**54.1**	67.2	48.6	64.0