Supplemental Material: Debiased Self-Training for Semi-Supervised Learning

Baixu Chen[*], Junguang Jiang[*], Ximei Wang, Pengfei Wan[§], Jianmin Wang, Mingsheng Long[✉]
School of Software, BNRist, Tsinghua University, China
[§]Y-tech, Kuaishou Technology
{chenbx18,jjg20}@mails.tsinghua.edu.cn, {jimwang,mingsheng}@tsinghua.edu.cn

A Implementation Details

Our code is based on PyTorch [4]. The following are the implementation details of our experiments. We have released the code for our method and that for all the baselines at `https://github.com/thuml/Debiased-Self-Training`.

A.1 Architecture

The architectures of different classifier heads are as follows. For nonlinear heads, we adopt Dropout [8] to alleviate over-fitting.

- Linear main head: `Linear-Softmax`;
- Nonlinear pseudo 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 Wide ResNet-28-2 [14] for *CIFAR-10* and *SVHN*, WRN-28-8 for *CIFAR-100*, WRN-37-2 for *STL-10* and adopt the same hyperparameters as FixMatch [7]. Specifically, we use learning rate of 0.03, mini-batch size of 512 (64 for labeled data, 448 for unlabeled data), weight decay in $\{0.0005, 0.001\}$, unlabeled loss weight $\lambda = 1.0$ and train for $1000k$ iterations. For our method, we set the projection dimension of the pseudo head and worst-case head to $2 \times n_{\text{embedding}}$, where $n_{\text{embedding}}$ denotes the dimension of backbone network output.

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 $\{0.1, 0.3, 1.0, 3.0\}$.

[*]Equal contribution.

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 [10].

FlexMatch. The same as Pseudo Label. Besides, we find it better to turn off threshold warm-up when using pre-trained models.

DebiasMatch. We search confidence threshold τ in $\{0.7, 0.8, 0.9, 0.95\}$, unlabeled loss weight λ in $\{0.1, 0.3, 1.0, 3.0\}$, debias factor in $\{0.1, 0.3, 1.0, 3.0\}$ and fix momentum to 0.999. For a fair comparison, we do not exploit additional supervision from pre-trained CLIP models [5].

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

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

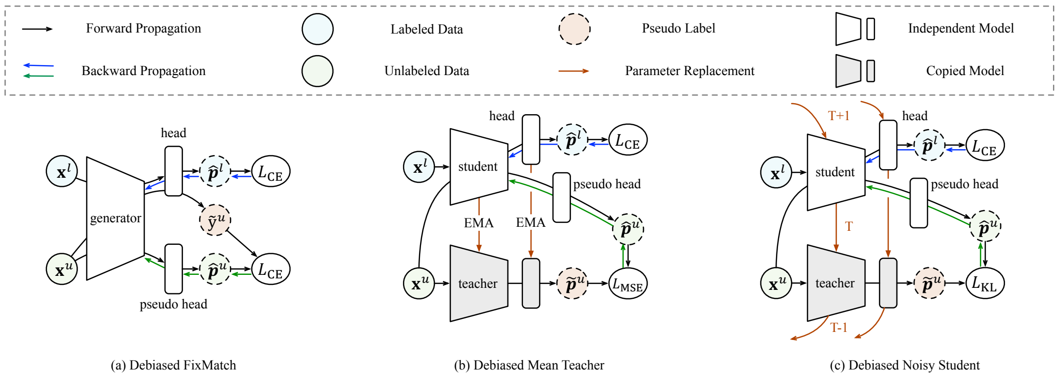


Figure 1: Illustrations on how different Debiased self-training methods generate and utilize pseudo labels.

In this section, we will illustrate how to incorporate DST into 4 typical self-training methods, including FixMatch, FlexMatch, 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-case heads since they are the same across different self-training methods.

Debiased FixMatch. As shown in Figure 1(a), the pseudo labels on the unlabeled data are generated by the main head h and utilized by the pseudo head h_{pseudo}. The main head h is only trained on the clean labeled data.

Debiased FlexMatch. The same as Debiased FixMatch. The learning status of each category is estimated by the main head h.

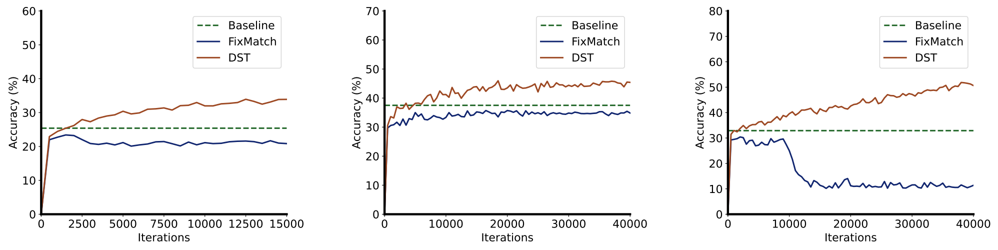
Debiased Mean Teacher. As shown in Figure 1(b), the pseudo labels on the unlabeled data are generated by the exponential moving average of the main head h and utilized by the pseudo head h_{pseudo}. The main head h is only trained on the clean labeled data.

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

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 2 illustrates several failure cases of FixMatch.



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

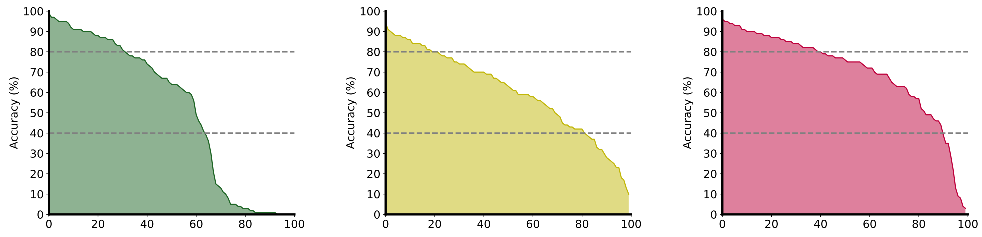
Figure 2: Failure cases of FixMatch with confidence threshold 0.7 (ResNet50).

Figures 2(a) and 2(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 2(c) demonstrates a complete failure case of FixMatch. With unsupervised pre-trained models and a confidence threshold of 0.7, there can be a lot of noise in pseudo labels and thus the performance of FixMatch declines severely. Note this result is not the entry we report in Table 2 (threshold is set to 0.9 for this dataset). Instead, we aim to show that DST improves the training stability when there is much noise.

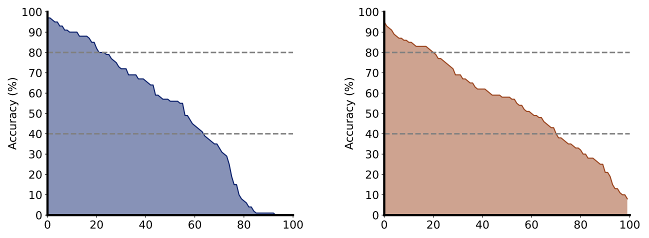
B.2 Experiments on performance balance between categories

Figures 3 and 4 plot the top-1 accuracy of each category on *CIFAR-100* yielded by self-training on 400 labels and unlabeled data when using supervised pre-trained models or training from scratch, respectively. The results are consistent with our previous analysis (Section 5.4) that DST improves the performance of those poorly-behaved categories.



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

Figure 3: Top-1 accuracy of each category on *CIFAR-100* (ResNet50, supervised pre-trained).



(a) FixMatch (b) DST

Figure 4: Top-1 accuracy of each category in on *CIFAR-100* (Wide ResNet-28-8, train from scratch).

B.3 Experiments with varying amounts of labeled data

Table 1 reports the performance of DST with 1000 labels on *CIFAR-100* with different pre-trained models. DST outperforms FlexMatch and DebiasMatch under both settings.

Table 1: Experiments with 10 labels per-class on *CIFAR-100* (ResNet50).

	Supervised Pre-Training	Unsupervised Pre-Training
Baseline	61.5	56.2
Pseudo Label [3]	67.4	57.3
Π-Model [2]	63.3	55.5
Mean Teacher [9]	67.0	63.5
UDA [12]	65.1	67.5
FixMatch [7]	67.8	64.2
Self-Tuning [10]	66.0	60.2
FlexMatch [15]	71.2	71.1
DebiasMatch [11]	73.5	73.9
DST (FixMatch)	**75.6**	**76.8**

Figure 5 plots the accuracy of FixMatch and DST when the number of labeled samples per class varies from 1 to 25 on *CIFAR-100* with supervised pre-trained models. Experiments show that DST is less sensitive to the amount of labeled data than FixMatch and yield consistent improvement.

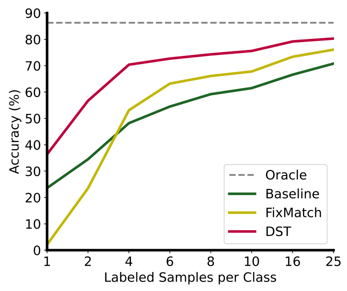


Figure 5: Ablation on the amount of labeled data on *CIFAR-100* (ResNet50, supervised pre-trained).

B.4 Analysis on the behavior of pseudo labels

B.4.1 Using unsupervised pre-trained models

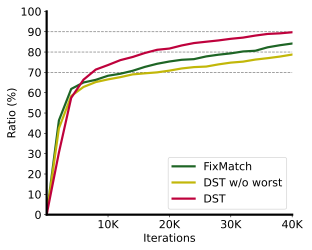
In this subsection, we explore the behavior of pseudo labels with unsupervised pre-trained models. Concretely, we focus on the quantity, accuracy as well as class imbalance ratio I of pseudo labels. Recall that $I = \max_c N(c)/\min_{c'} N(c')$, where $N(c)$ denotes the number of predictions that fall into category c. Figure 6 shows the results on *CIFAR-100* with 400 labels. We observe the same phenomenon that DST effectively reduces the bias of pseudo labels, thereby improving the self-training process.

B.4.2 Comparison with other methods

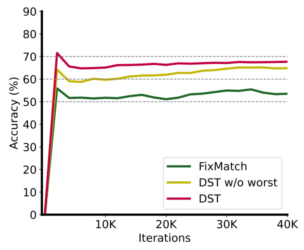
We consider two lines of work that promote the quality of pseudo labels by (1) using dynamic threshold, including Dash [13] and FlexMatch [15]; (2) adopting multi-view training, including Co-training [1] and Multi-task Tri-training [6]. Table 2 shows that DST outperforms baselines by considerable margins. Figure 7 plots the quality of pseudo labels and reveals that our method can better debias pseudo labeling and improve the quality of pseudo labels.

B.5 Ablation study on nonlinear main classifier head

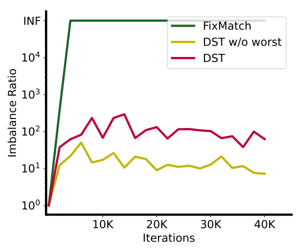
Experiments suggest that using a nonlinear pseudo head improves performance. We further explore how things are going for the main head. As shown in Table 3, using nonlinear main head or not



(a) Quantity



(b) Quality

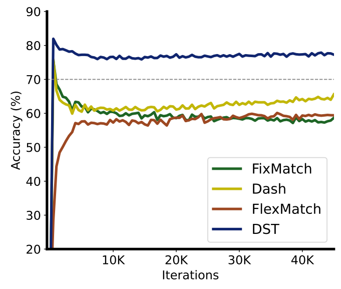


(c) Quantity of poorly-behaved categories

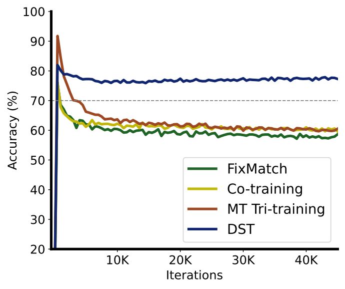
Figure 6: Analysis on the behavior of pseudo labels on *CIFAR-100* (ResNet50, unsupervised pre-trained). **(a)** The quantity of pseudo labels above the confidence threshold. **(b)** The accuracy of pseudo labels. **(c)** The class imbalance ratio I of pseudo labels.

Table 2: Comparison with other methods that improve the quality of pseudo labels (*CIFAR-100*, ResNet50, supervised pre-trained).

		Dynamic Thresholding		Multi-head Training		
Baseline	FixMatch	Dash	FlexMatch	Co-training	MT Tri-training	DST
48.2	53.1	55.4	63.4	54.4	54.4	**70.4**



(a) Comparison with methods that adopt dynamic threshold.



(b) Comparison with methods that adopt multi-view training.

Figure 7: Quality of pseudo labels (*CIFAR-100*, ResNet50, supervised pre-trained).

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

Head	Supervised Pre-Training		Unsupervised Pre-Training	
	400 labels	1000 labels	400 labels	1000 labels
Linear	53.1	**67.8**	**51.4**	**64.2**
Nonlinear	**54.1**	67.2	50.4	64.0

C Broader Impact

First, our research helps improve the performance and training stability of various existing self-training methods, especially when the labeled data is scarce. Second, our proposed method is simple yet effective and thus can potentially reduce the labeling cost of many real-world machine learning applications. Finally, our research helps reduce the bias of self-training models and improves their performance balance.

References

[1] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 1998.

[2] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

[3] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML*, 2013.

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[6] Sebastian Ruder and Barbara Plank. Strong baselines for neural semi-supervised learning under domain shift. In *ACL*, 2018.

[7] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

[8] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. In *ICML*, 2014.

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

[10] Ximei Wang, Jinghan Gao, Mingsheng Long, and Jianmin Wang. Self-tuning for data-efficient deep learning. In *ICML*, 2021.

[11] Xudong Wang, Zhirong Wu, Long Lian, and Stella X Yu. Debiased learning from naturally imbalanced pseudo-labels for zero-shot and semi-supervised learning. In *CVPR*, 2022.

[12] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. In *NeurIPS*, 2020.

[13] Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In *ICML*, 2021.

[14] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

[15] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *NeurIPS*, 2021.