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

Hyperparameter setting

We show the detailed training hyperparameter settings for each method in Table 1. We also report the detailed hyperparameter settings with a specific model for each dataset in Table 2.

Detailed results

Following the suggestion from [4], we also report the median error rates of the last 20 checkpoints in Table 3. The results show that our proposed ReFixMatch improves performance and surpasses previous methods by a large margin. Furthermore, the results also show that the model trained using ReFixMatch keeps improving until the end of the training process, while FlexMatch is overfit to the data.

Qualitative Analysis

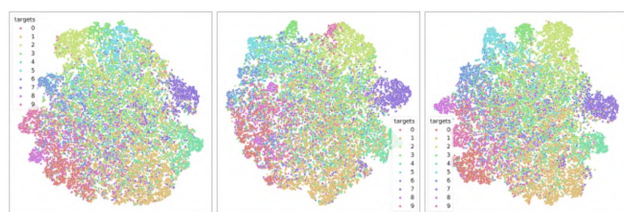
We present the T-SNE visualization of features on STL-10 test dataset with 40-label split in Figure 1a,1b,1c. The visualization is using trained model from FixMatch, FlexMatch and ReFixMatch.

(a) FixMatch (b) FlexMatch (c) ReFixMatch

Figure 1. T-SNE visualization on STL-10 dataset with 40 labels.

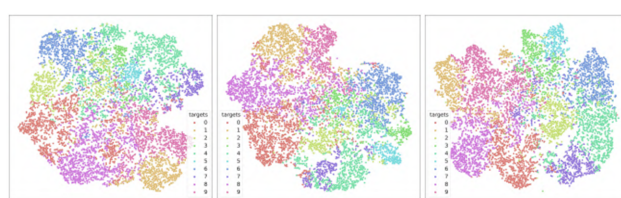
Figures 2a, 2b, and 3c show the T-SNE visualization of features on the SVHN test dataset and the CIFAR-10 test dataset with a 40-label split.

As we can see, ReFixMatch produces a much clearer boundary for each class. This clearly shows that ReFixMatch improves the generalization of the model. In addition, we could see that although FlexMatch gives high performance, its border for class separation is not clear, this is due to the use of low threshold.



(a) FixMatch (b) FlexMatch (c) ReFixMatch

Figure 2. T-SNE visualization on SVHN dataset with 40 labels.



(a) FixMatch (b) FlexMatch (c) ReFixMatch

Figure 3. T-SNE visualization on CIFAR-10 dataset with 40 labels.

Table 1. Training hyperparameters

Algorithm	UDA	ReFixMatch	FixMatch (FlexMatch)
Unlabeled Data to Labeled Data Ratio (CIFAR-10/100, STL-10, SVHN)	7	7	7
Unlabeled Data to Labeled Data Ratio (ImageNet)	-	1	1
Pre-defined Threshold (CIFAR-10/100, STL-10, SVHN)	0.8	0.95	0.95
Pre-defined Threshold (ImageNet)	-	0.7	0.7
Temperature	0.4	0.5	-

Table 2. Dataset-wise hyperparameters

Dataset	CIFAR-10	CIFAR-100	STL-10	SVHN	ImageNet
Model	WRN-28-2	WRN-28-8	WRN-37-2	WRN-28-2	ResNet-50
Weight Decay	5e-4	1e-3	5e-4	5e-4	3e-4
Batch Size	64				128
Learning Rate	0.03				
SGD Momentum	0.9				
EMA Momentum	0.999				
Unsupervised Loss Weight	1				

Table 3. Mean error rates of last 20 checkpoints of all methods. There are 1000 iterations between every two checkpoints

Dataset	CIFAR-10			CIFAR-100			SVHN			STL-10	
# Label	40	250	4000	400	2500	10000	40	250	1000	40	1000
Π Model	$78.78_{\pm 2.24}$	$55.79_{\pm 2.61}$	$13.63_{\pm 0.60}$	$89.27_{\pm 0.73}$	$60.58_{\pm 0.66}$	$38.49_{\pm 0.09}$	$76.23_{\pm 4.60}$	$18.44_{\pm 2.79}$	$7.77_{\pm 0.03}$	$77.80_{\pm 0.63}$	$35.63_{\pm 0.25}$
Pseudo Label	$77.42_{\pm 1.19}$	$48.33_{\pm 2.43}$	$15.64_{\pm 0.29}$	$90.01_{\pm 0.21}$	$58.38_{\pm 0.42}$	$37.64_{\pm 0.16}$	$69.05_{\pm 6.77}$	$16.76_{\pm 1.02}$	$9.99_{\pm 0.35}$	$76.44_{\pm 0.67}$	$33.57_{\pm 0.40}$
VAT	$81.90_{\pm 2.39}$	$42.43_{\pm 1.86}$	$10.83_{\pm 0.07}$	$89.28_{\pm 1.71}$	$47.44_{\pm 0.68}$	$32.66_{\pm 0.33}$	$80.19_{\pm 4.08}$	$4.54_{\pm 0.12}$	$4.31_{\pm 0.20}$	$78.34_{\pm 1.24}$	$48.36_{\pm 0.29}$
Mean Teacher	$77.96_{\pm 2.63}$	$42.47_{\pm 3.79}$	$8.49_{\pm 0.21}$	$81.58_{\pm 1.51}$	$45.61_{\pm 1.12}$	$32.38_{\pm 0.12}$	$47.12_{\pm 2.96}$	$3.56_{\pm 0.04}$	$3.38_{\pm 0.03}$	$76.04_{\pm 2.94}$	$38.94_{\pm 1.14}$
UDA	$10.96_{\pm 3.68}$	$5.46_{\pm 0.07}$	$4.60_{\pm 0.05}$	$51.97_{\pm 1.38}$	$29.92_{\pm 0.35}$	$23.64_{\pm 0.33}$	$5.31_{\pm 4.39}$	$\mathbf{2.01}_{\pm \mathbf{0.03}}$	$1.97_{\pm 0.04}$	$41.11_{\pm 5.21}$	$8.00_{\pm 0.58}$
FixMatch	$7.99_{\pm 0.59}$	$\mathbf{5.12}_{\pm \mathbf{0.03}}$	$4.46_{\pm 0.11}$	$48.95_{\pm 1.19}$	$29.19_{\pm 0.25}$	$23.06_{\pm 0.12}$	$3.92_{\pm 1.18}$	$2.09_{\pm 0.03}$	$2.06_{\pm 0.01}$	$44.70_{\pm 6.58}$	$7.38_{\pm 0.26}$
Dash	$11.02_{\pm 4.05}$	$5.43_{\pm 0.20}$	$4.68_{\pm 0.07}$	$47.88_{\pm 1.31}$	$28.62_{\pm 0.41}$	$22.92_{\pm 0.15}$	$2.28_{\pm 0.18}$	$2.12_{\pm 0.04}$	$2.07_{\pm 0.01}$	$41.21_{\pm 5.25}$	$7.52_{\pm 0.81}$
MPL	$9.65_{\pm 3.02}$	$6.08_{\pm 0.48}$	$4.76_{\pm 0.06}$	$48.45_{\pm 1.61}$	$28.41_{\pm 0.14}$	$\mathbf{22.25}_{\pm \mathbf{0.18}}$	$14.74_{\pm 14.69}$	$2.41_{\pm 0.04}$	$2.39_{\pm 0.01}$	$41.49_{\pm 3.90}$	$7.05_{\pm 0.51}$
FlexMatch	$5.19_{\pm 0.05}$	$5.33_{\pm 0.12}$	$4.47_{\pm 0.09}$	$45.91_{\pm 1.76}$	$28.11_{\pm 0.20}$	$23.04_{\pm 0.28}$	$20.81_{\pm 5.26}$	$17.32_{\pm 2.07}$	$12.90_{\pm 2.68}$	$44.69_{\pm 7.49}$	$\mathbf{6.15}_{\pm \mathbf{0.25}}$
ReFixMatch	$\mathbf{5.03}_{\pm \mathbf{0.11}}$	$5.16_{\pm 0.10}$	$\mathbf{4.43}_{\pm \mathbf{0.02}}$	$\mathbf{44.52}_{\pm \mathbf{1.01}}$	$\mathbf{27.95}_{\pm \mathbf{0.22}}$	$23.01_{\pm 0.18}$	$\mathbf{2.20}_{\pm \mathbf{0.34}}$	$2.03_{\pm 0.03}$	$\mathbf{2.01}_{\pm \mathbf{0.01}}$	$\mathbf{40.21}_{\pm \mathbf{6.11}}$	$6.54_{\pm 0.26}$

ImageNet detailed results

Table 4 shows the detailed results from Table ??. ReFixMatch without using self-supervised pre-trained weights outperforms previous methods such as CoMatch [3] and SimMatch [6]. ReFixMatch achieves 75.2% of top-1 accuracy with the same training duration ($\sim$ 400 epochs) and has fewer parameters of 25.6M during training compared to 30.0M for FixMatch-EMAN, CoMatch, and SimMatch.

A. List of Data Transformations

We report the detailed augmentations used in our method in Table 5. This list of transformations is similar to the original list used in FixMatch [4] and FlexMatch [5].

Precision, Recall, F1 and AUC

We further report precision, recall, F1-score, and AUC (area under curve) results on the CIFAR-10 dataset. As shown in Table 6, ReFixMatch also has the best performance on precision, recall, F1-score, and AUC.

Table 4. Accuracy results on ImageNet with **10%** labeled examples using [3] and [6] source code.

Self-supervised Pre-training	Method	Top-1	Top-5	Params (train/test)	Epochs
None	FixMatch	71.5	89.1	25.6M/25.6M	$\sim$ 300
MoCo-EMAN [1]	FixMatch-EMAN [1]	74.0	90.9	30.0M/25.6M	$\sim$ 1100
None	CoMatch [3]	73.6	91.6	30.0M/25.6M	$\sim$ 400
MoCo V2 [2]	CoMatch [3]	73.7	91.4	30.0M/25.6M	$\sim$ 1200
None	SimMatch [6]	74.4	91.6	30.0M/25.6M	$\sim$ 400
None	**ReFixMatch**	**75.2**	**91.9**	25.6M/25.6M	$\sim$ 400

Table 5. List of transformations used in RandAugment

Transformation	Description	Parameter	Range
Autocontrast	Maximizes the image contrast by setting the darkest (lightest) pixel to black (white).		
Brightness	Adjusts the brightness of the image. $B = 0$ returns a black image, $B = 1$ returns the original image.	B	[0.05, 0.95]
Color	Adjusts the color balance of the image like in a TV. $C = 0$ returns a black & white image, $C = 1$ returns the original image.	C	[0.05, 0.95]
Contrast	Controls the contrast of the image. A $C = 0$ returns a gray image, $C = 1$ returns the original image.	C	[0.05, 0.95]
Equalize	Equalizes the image histogram.		
Identity	Returns the original image.		
Posterize	Reduces each pixel to B bits.	B	[4, 8]
Rotate	Rotates the image by θ degrees.	θ	[-30, 30]
Sharpness	Adjusts the sharpness of the image, where $S = 0$ returns a blurred image, and $S = 1$ returns the original image.	S	[0.05, 0.95]
Shear_x	Shears the image along the horizontal axis with rate R.	R	[-0.3, 0.3]
Shear_y	Shears the image along the vertical axis with rate R.	R	[-0.3, 0.3]
Solarize	Inverts all pixels above a threshold value of T.	T	[0, 1]
Translate_x	Translates the image horizontally by ($\lambda\times$image width) pixels.	λ	[-0.3, 0.3]
Translate_y	Translates the image vertically by ($\lambda\times$image height) pixels.	λ	[-0.3, 0.3]

Table 6. Precision, recall, F1-score and AUC results on CIFAR-10.

Label Amount	40 labels				4000 labels			
Criteria	Precision	Recall	F1-score	AUC	Precision	Recall	F1-score	AUC
FixMatch	0.9333	0.9290	0.9278	0.9910	0.9571	0.9571	0.9569	0.9984
FlexMatch	0.9506	0.9507	0.9506	0.9975	0.9580	0.9581	0.9580	0.9984
ReFixMatch	**0.9513**	**0.9513**	**0.9510**	**0.9976**	**0.9582**	**0.9583**	**0.9582**	**0.9986**

References

[1] Zhaowei Cai, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Zhuowen Tu, and Stefano Soatto. Exponential moving average normalization for self-supervised and semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages

194–203, 2021. 3

[2] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 3

[3] 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. 2, 3

[4] 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

[5] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021. 2

[6] 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, 3