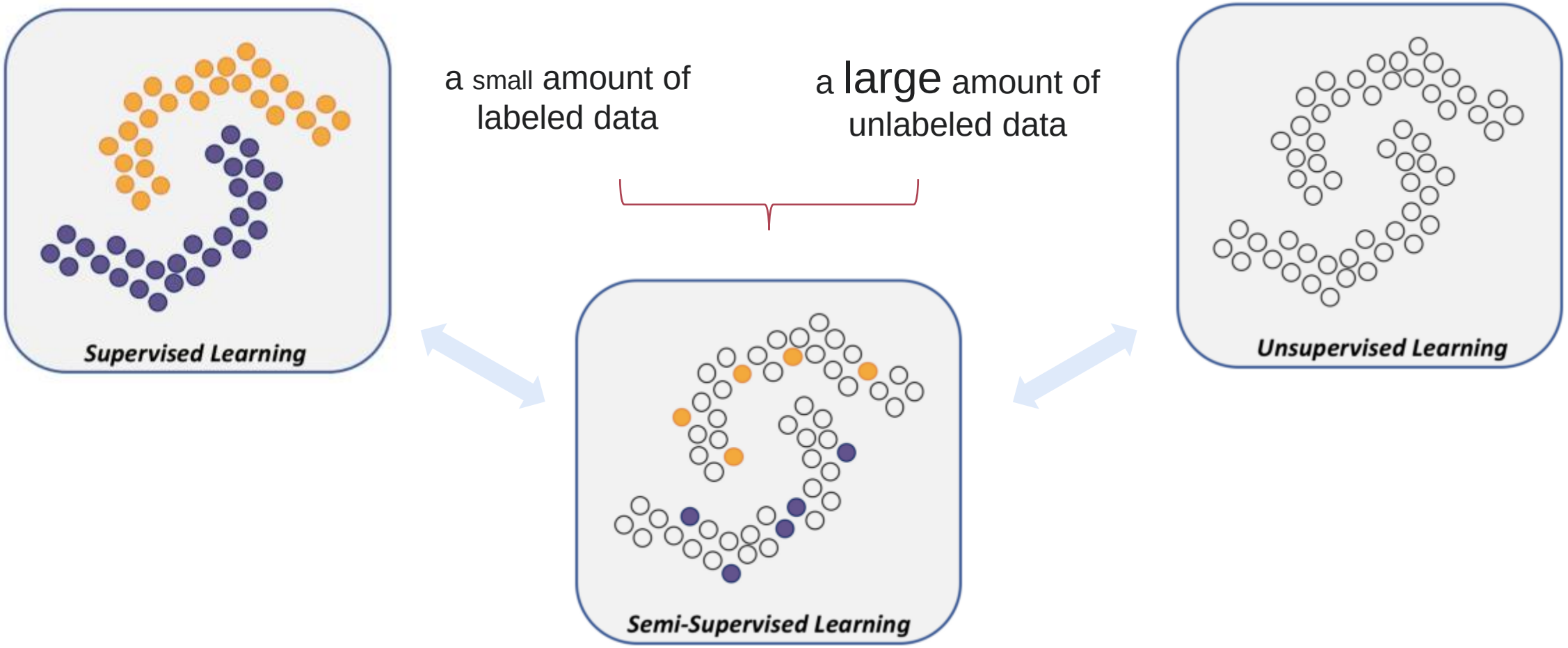
FreeMatch: Self-adaptive Thresholding for Semi-supervised Learning

Yidong Wang[1,2*], **Hao Chen**[3*], **Qiang Heng**[4], **Wenxin Hou**[5], **Yue Fan**[6], **Zhen Wu**[7], **Jindong Wang**[1†], **Marios Savvides**[3], **Takahiro Shinozaki**[2], **Bhiksha Raj**[3,8], **Bernt Schiele**[6], **Xing Xie**[1]

[1]Microsoft Research Asia, [2]Tokyo Institute of Technology, [3]Carnegie Mellon University,
[4]North Carolina State University, [5]Microsoft STCA,
[6]Max Planck Institute for Informatics, Saarland Informatics Campus,
[7]Nanjing University, [8]Mohamed bin Zayed University of AI

ICLR 2023

Semi-supervised Learning (SSL)



Recent methods of SSL

① Consistency regularization —— Mean Teachers (2018) UDA (2020)

② Pseudo labeling —— Noisy Student (2020) UPS (2021)

③ Holistic methods (①+②) ——

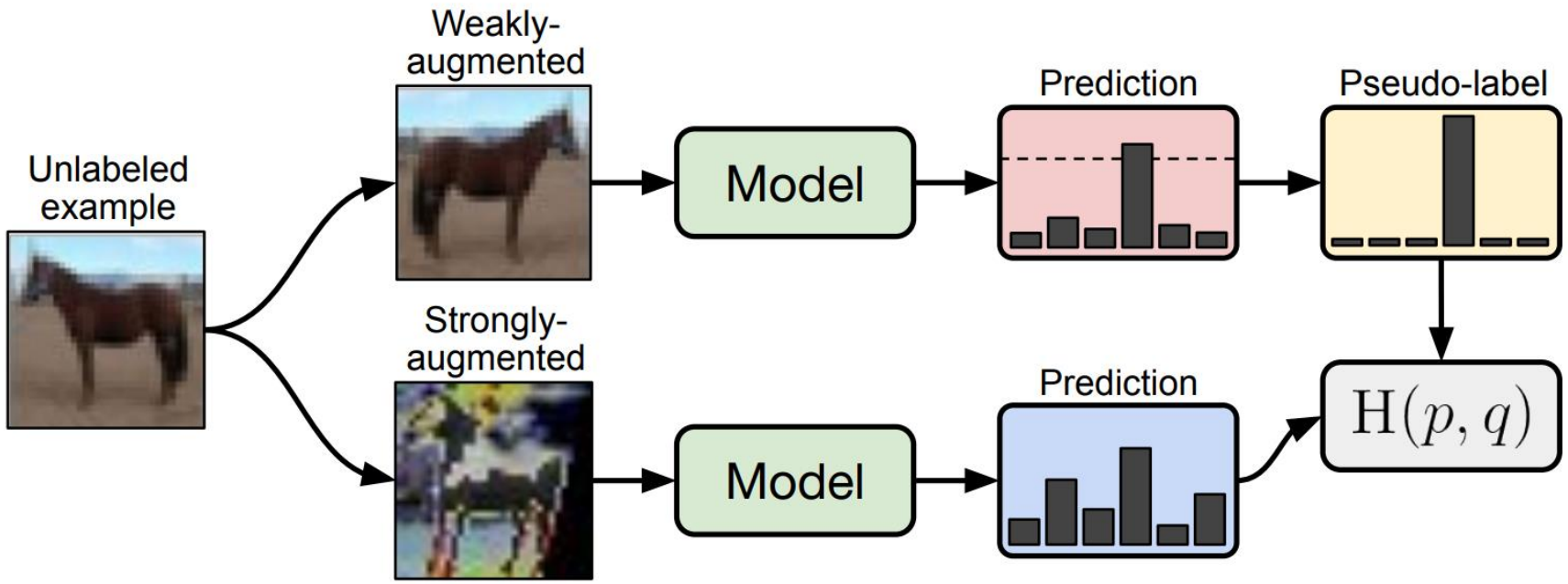
- MixMatch (2019)
- ReMixMatch (2020)
- FixMatch (2020)
- CoMatch (2021)
- Dash (2021)
- FlexMatch (2021)
- AdaMatch (2022)
- Adsh (2022)

– derive from xxMatch

– adaptive threshold

Related work

FixMatch (2020)



$$\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) \geq \tau)\, \mathrm{H}(\hat{q}_b, p_{\mathrm{m}}(y \mid \mathcal{A}(u_b)))$$

τ is fixed high for high-quality unlabeled data, while ignoring a considerable amount of other unlabeled data.

Related work

How to make τ alive ?

Dash (2021)

global threshold

$$\rho_t := C\gamma^{-(t-1)}\widehat{\rho},$$

$$I(\max(\mathbf{h}_i) \geq \tau) \longrightarrow I(f_u(\mathbf{w}; \xi_i^u) \leq \rho_t)$$

AdaMatch (2022)

$$c_\tau = \frac{\tau}{n_{SL}} \sum_{i=1}^{n_{SL}} \max_{j \in [1..k]} (\hat{Y}_{SL,w}^{(i,j)})$$

as the mean confidence of the top-1 prediction multiplied by a user provided threshold τ

FlexMatch (2021)

$$\mathcal{T}_t(c) = \beta_t(c) \cdot \tau. \qquad \beta_t(c) = \frac{\sigma_t(c)}{\max_c \sigma_t}, \qquad \sigma_t(c) = \sum_{n=1}^{N} \mathbb{1}(\max(p_{m,t}(y|u_n)) > \tau) \cdot \mathbb{1}(\arg\max(p_{m,t}(y|u_n) = c)$$

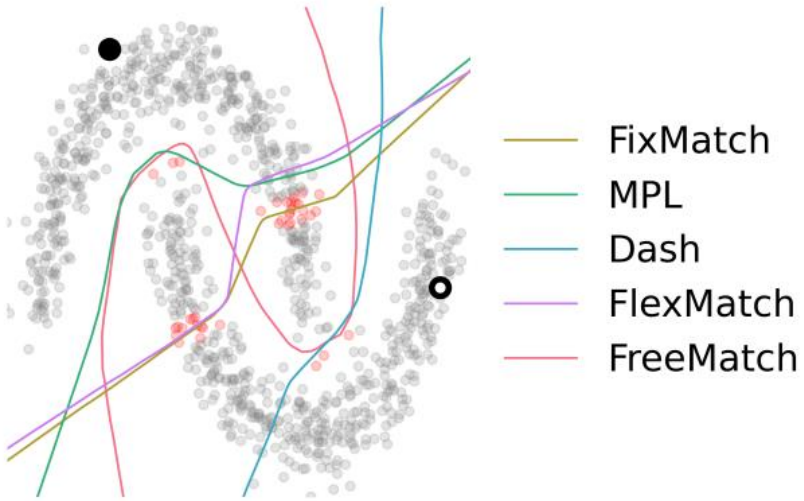
local threshold

Adsh (2022)

τ_1 is for the most majority class, other classes' τ_c fit τ_1 based on percentage of selected pseudo-labels for the most majority class. (for imbalanced SSL)

Limitation

In a nutshell, these methods might be incapable or insufficient in terms of adjusting thresholds according to model's learning progress, thus impeding the training process especially when labeled data is too scarce to provide adequate supervision.



(a) Decision boundary

As shown in Figure (a), on the "two-moon" dataset **with only 1 labeled sample for each class**, the decision boundaries obtained by previous methods fail in the low-density assumption.

Two questions naturally arise:

1) Is it necessary to determine the threshold based on the model learning status?
2) How to adaptively adjust the threshold for best training efficiency?

A motivating example

$$X \mid Y = -1 \sim \mathcal{N}(\mu_1, \sigma_1^2), X \mid Y = +1 \sim \mathcal{N}(\mu_2, \sigma_2^2). \tag{1}$$

We then derive the following theorem to show the necessity of self-adaptive threshold:

Theorem 2.1. *For a binary classification problem as mentioned above, the pseudo label Y_p has the following probability distribution:*

$$\begin{aligned}
P(Y_p = 1) &= \frac{1}{2}\Phi\left(\frac{\frac{\mu_2-\mu_1}{2} - \frac{1}{\beta}\log\left(\frac{\tau}{1-\tau}\right)}{\sigma_2}\right) + \frac{1}{2}\Phi\left(\frac{\frac{\mu_1-\mu_2}{2} - \frac{1}{\beta}\log\left(\frac{\tau}{1-\tau}\right)}{\sigma_1}\right), \\
P(Y_p = -1) &= \frac{1}{2}\Phi\left(\frac{\frac{\mu_2-\mu_1}{2} - \frac{1}{\beta}\log\left(\frac{\tau}{1-\tau}\right)}{\sigma_1}\right) + \frac{1}{2}\Phi\left(\frac{\frac{\mu_1-\mu_2}{2} - \frac{1}{\beta}\log\left(\frac{\tau}{1-\tau}\right)}{\sigma_2}\right), \\
P(Y_p = 0) &= 1 - P(Y_p = 1) - P(Y_p = -1),
\end{aligned} \tag{2}$$

where Φ is the cumulative distribution function of a standard normal distribution. Moreover, $P(Y_p = 0)$ increases as $\mu_2 - \mu_1$ gets smaller.

β is a positive parameter that reflects the model learning status, τ is a threshold.

Method

FreeMatch

$$\mathcal{L}_s = \frac{1}{B}\sum_{b=1}^{B}\mathcal{H}(y_b, p_m(y|\omega(x_b)))$$

– backbone of FixMatch

– novel component

The overall objective: $\mathcal{L} = \mathcal{L}_s + w_u\mathcal{L}_u + w_f\mathcal{L}_f$

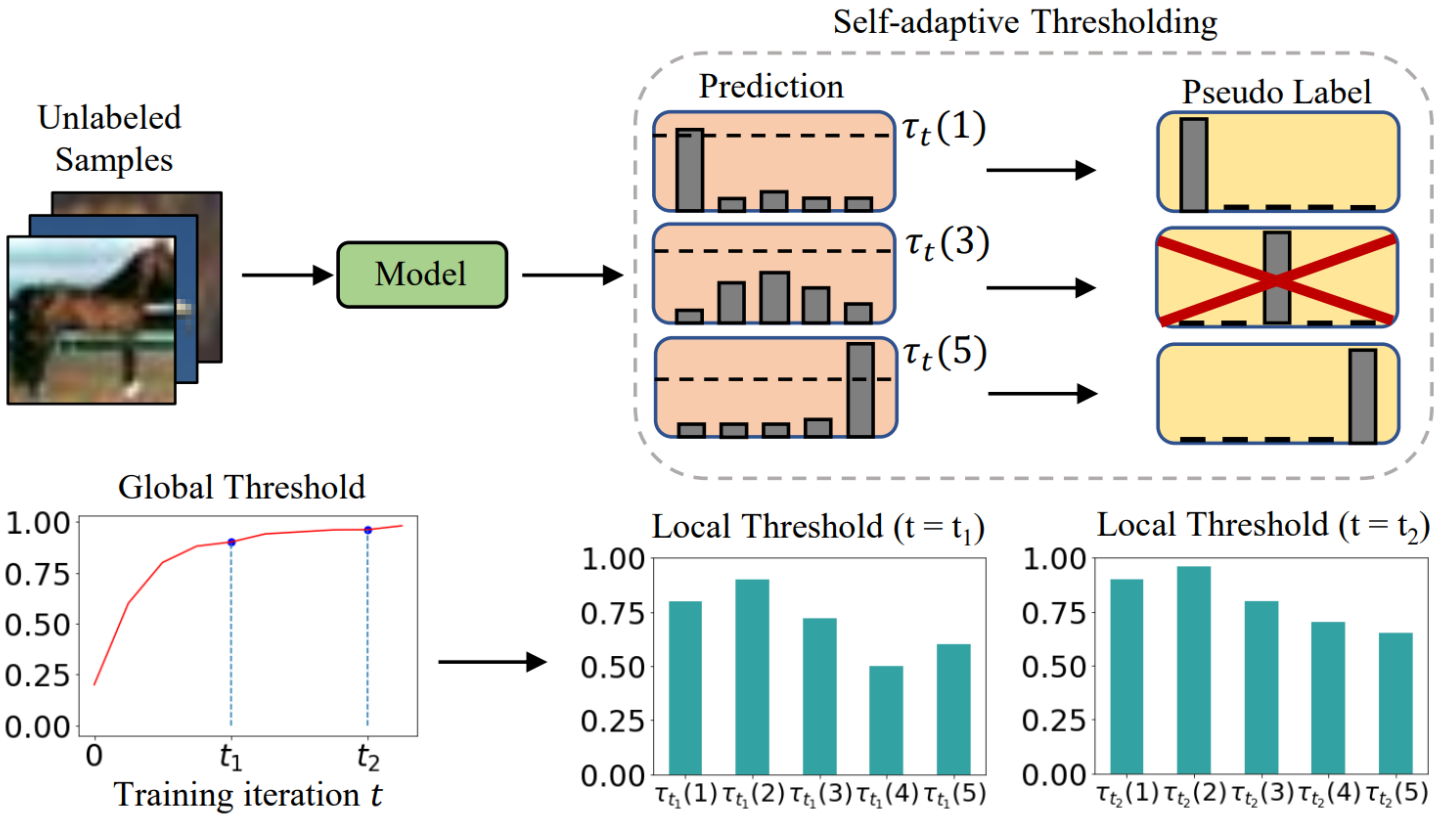
$$\mathcal{L}_u = \frac{1}{\mu B}\sum_{b=1}^{\mu B}\mathbb{1}(\max(q_b) > \tau_t(\arg\max(q_b)) \cdot \mathcal{H}(\hat{q}_b, Q_b)$$

$$\mathcal{L}_f = -\mathcal{H}\left(\text{SumNorm}\left(\frac{\tilde{p}_t}{\tilde{h}_t}\right), \text{SumNorm}\left(\frac{\bar{p}}{\bar{h}}\right)\right)$$

Self-Adaptive Thresholding (SAT)

Self-Adaptive Fairness (SAF)

Self-Adaptive Thresholding (SAT)



global (dataset-specific) and *local* (class-specific) thresholds based on the model's learning status

Self-Adaptive Thresholding (SAT)

global (dataset-specific)

$$\tau_t = \begin{cases} \frac{1}{C}, & \text{if } t = 0, \\ \lambda \tau_{t-1} + (1-\lambda)\frac{1}{\mu B}\sum_{b=1}^{\mu B} \max(q_b), & \text{otherwise,} \end{cases}$$

local (class-specific)

$$\tilde{p}_t(c) = \begin{cases} \frac{1}{C}, & \text{if } t = 0, \\ \lambda \tilde{p}_{t-1}(c) + (1-\lambda)\frac{1}{\mu B}\sum_{b=1}^{\mu B} q_b(c), & \text{otherwise,} \end{cases}$$

where $\lambda \in (0,1)$ is the momentum decay of EMA, q_b is an abbreviation of $p_m\big(y|\omega(u_b)\big)$

Integrating the *global* and *local* thresholds, we obtain the final self-adaptive threshold as:

$$\tau_t(c) = \mathrm{MaxNorm}(\tilde{p}_t(c)) \cdot \tau_t = \frac{\tilde{p}_t(c)}{\max\{\tilde{p}_t(c) : c \in [C]\}} \cdot \tau_t$$

Self-Adaptive Fairness (SAF)

A common fairness objective: $\mathcal{L}_f = \mathbf{U} \log \mathbb{E}_{\mu B} [q_b]$, where $\mathbf{U}$ is a uniform prior distribution.

Considering that the underlying pseudo label distribution may not be uniform, SAF is proposed:

$$\mathcal{L}_f = -\mathcal{H}\left(\text{SumNorm}\left(\frac{\tilde{p}_t}{\tilde{h}_t}\right), \text{SumNorm}\left(\frac{\bar{p}}{\bar{h}}\right)\right)$$

$$\tilde{p}_t = \lambda \tilde{p}_{t-1} + (1-\lambda)\frac{1}{\mu B}\sum_{b=1}^{\mu B} q_b$$

$$\tilde{h}_t = \lambda \tilde{h}_{t-1} + (1-\lambda)\,\text{Hist}_{\mu B}(\hat{q}_b)$$

$$\bar{p} = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{1}\left(\max(q_b) \geq \tau_t(\arg\max(q_b))\right) Q_b,$$

$$\bar{h} = \text{Hist}_{\mu B}\left(\mathbb{1}\left(\max(q_b) \geq \tau_t(\arg\max(q_b))\right) \hat{Q}_b\right)$$

Method

Algorithm 1 FreeMatch algorithm at t-th iteration.

Input: Number of classes C, labeled batch $\mathcal{X} = \{(x_b, y_b) : b \in (1, 2, \ldots, B)\}$, unlabeled batch $\mathcal{U} = \{u_b : b \in (1, 2, \ldots, \mu B)\}$, unsupervised loss weight w_u, fairness loss weight w_f, and EMA decay λ.

L_s

Compute $\mathcal{L}_s$ for labeled data

$\mathcal{L}_s = \frac{1}{B} \sum_{b=1}^{B} \mathcal{H}(y_b, p_m(y|\omega(x_b)))$

...

Update the global threshold

$\tau_t = \lambda \tau_{t-1} + (1 - \lambda) \frac{1}{\mu B} \sum_{b=1}^{\mu B} max(q_b)$ {q_b is an abbreviation of $p_m(y|\omega(u_b))$, shape of τ_t: [1] }

...

Update histogram for $\tilde{p}_t$

$\tilde{h}_t = \lambda \tilde{h}_{t-1} + (1 - \lambda)\, \mathrm{Hist}_{\mu B}\, (\hat{q}_b)$ {Shape of $\tilde{h}_t$: $[C]$}

L_u

for $c = 1$ to C **do**

  $\tau_t(c) = \mathrm{MaxNorm}(\tilde{p}_t(c)) \cdot \tau_t$ {Calculate SAT}

end for

Compute $\mathcal{L}_u$ on unlabeled data

$\mathcal{L}_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\left(\max\left(q_b\right) \geq \tau_t(\arg\max\left(q_b\right))\right) \cdot \mathcal{H}(\hat{q}_b, Q_b)$

L_f

Compute expectation of probability on unlabeled data

$\overline{p} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}\left(\max\left(q_b\right) \geq \tau_t(\arg\max\left(q_b\right)\right) Q_b$ {Q_b is an abbr. of $p_m(y|\Omega(u_b))$, shape of $\overline{p}$: $[C]$}

Compute histogram for $\overline{p}$

$\overline{h} = \mathrm{Hist}_{\mu B}\left(\mathbb{1}\left(\max\left(q_b\right) \geq \tau_t(\arg\max\left(q_b\right)\right) \hat{Q}_b\right)$ {Shape of $\overline{h}$: $[C]$}

Compute $\mathcal{L}_f$ on unlabeled data

$\mathcal{L}_f = -\mathcal{H}\left(\mathrm{SumNorm}(\frac{\tilde{p}_t}{\tilde{h}_t}), \mathrm{SumNorm}(\frac{\overline{p}}{\overline{h}})\right)$

Return: $\mathcal{L}_s + w_u \cdot \mathcal{L}_u + w_f \cdot \mathcal{L}_f$

Quantitative results

Table1: Error rates on CIFAR-10/100, SVHN, and STL-10 datasets.

Dataset	CIFAR-10				CIFAR-100			SVHN			STL-10	
# Label	10	40	250	4000	400	2500	10000	40	250	1000	40	1000
Π Model (Rasmus et al., 2015)	79.18±1.11	74.34±1.76	46.24±1.29	13.13±0.59	86.96±0.80	58.80±0.66	36.65±0.00	67.48±0.95	13.30±1.12	7.16±0.11	74.31±0.85	32.78±0.40
Pseudo Label (Lee et al., 2013)	80.21±0.55	74.61±0.26	46.49±2.20	15.08±0.19	87.45±0.85	57.74±0.28	36.55±0.24	64.61±5.6	15.59±0.95	9.40±0.32	74.68±0.99	32.64±0.71
VAT (Miyato et al., 2018)	79.81±1.17	74.66±2.12	41.03±1.79	10.51±0.12	85.20±1.40	46.84±0.79	32.14±0.19	74.75±3.38	4.33±0.12	4.11±0.20	74.74±0.38	37.95±1.12
MeanTeacher (Tarvainen & Valpola, 2017)	76.37±0.44	70.09±1.60	37.46±3.30	8.10±0.21	81.11±1.44	45.17±1.06	31.75±0.23	36.09±3.98	3.45±0.03	3.27±0.05	71.72±1.45	33.90±1.37
MixMatch (Berthelot et al., 2019b)	65.76±7.06	36.19±6.48	13.63±0.59	6.66±0.26	67.59±0.66	39.76±0.48	27.78±0.29	30.60±8.39	4.56±0.32	3.69±0.37	54.93±0.96	21.70±0.68
ReMixMatch (Berthelot et al., 2019a)	20.77±7.48	9.88±1.03	6.30±0.05	4.84±0.01	42.75±1.05	**26.03**±0.35	**20.02**±0.27	24.04±9.13	6.36±0.22	5.16±0.31	32.12±6.24	6.74±0.14
UDA (Xie et al., 2020a)	34.53±10.69	10.62±3.75	5.16±0.06	4.29±0.07	46.39±1.59	27.73±0.21	22.49±0.23	5.12±4.27	**1.92**±0.05	**1.89**±0.01	37.42±8.44	6.64±0.17
FixMatch (Sohn et al., 2020)	24.79±7.65	7.47±0.28	**4.86**±0.05	4.21±0.08	46.42±0.82	28.03±0.16	22.20±0.12	3.81±1.18	2.02±0.02	1.96±0.03	35.97±4.14	6.25±0.33
Dash (Xu et al., 2021)	27.28±14.09	8.93±3.11	5.16±0.23	4.36±0.11	44.82±0.96	27.15±0.22	21.88±0.07	2.19±0.18	2.04±0.02	1.97±0.01	34.52±4.30	6.39±0.56
MPL (Pham et al., 2021)	23.55±6.01	6.62±0.91	5.76±0.24	4.55±0.04	46.26±1.84	27.71±0.19	21.74±0.09	9.33±8.02	2.29±0.04	2.28±0.02	35.76±4.83	6.66±0.00
FlexMatch (Zhang et al., 2021)	13.85±12.04	4.97±0.06	4.98±0.09	4.19±0.01	39.94±1.62	26.49±0.20	21.90±0.15	8.19±3.20	6.59±2.29	6.72±0.30	29.15±4.16	5.77±0.18
FreeMatch	**8.07**±4.24	**4.90**±0.04	4.88±0.18	**4.10**±0.02	**37.98**±0.42	26.47±0.20	21.68±0.03	**1.97**±0.02	1.97±0.01	1.96±0.03	**15.56**±0.55	**5.63**±0.15
Fully-Supervised	4.62±0.05				19.30±0.09			2.13±0.01			-	

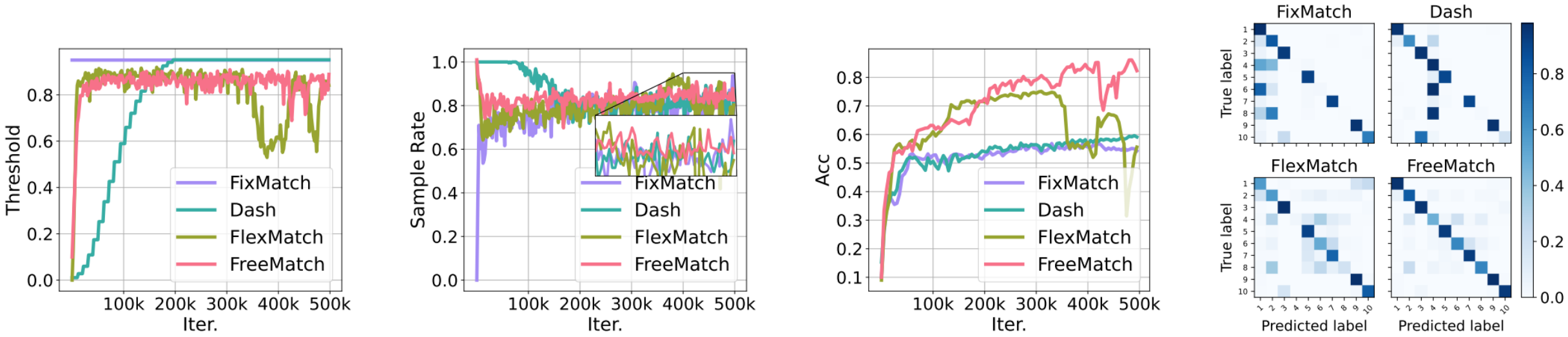
FreeMatch consistently outperforms other methods by a large margin on settings with *extremely limited labeled data*.

On ImageNet with 100k labels, FreeMatch significantly outperforms the latest counterpart FlexMatch by **1.28**%.

	Top-1	Top-5	Runtime (sec./iter.)
FixMatch	43.66	21.80	**0.4**
FlexMatch	41.85	19.48	0.6
FreeMatch	**40.57**	**18.77**	**0.4**

Table2: Error rates and runtime on ImageNet with 100 labels per class

Qualitative results



(a) Confidence threshold (b) Sampling rate (c) Accuracy (d) Confusion matrix

Figure 3: How FreeMatch works in STL-10 with 40 labels, compared to others. (a) Class-average confidence threshold; (b) class-average sampling rate; (c) convergence speed in terms of accuracy; (d) confusion matrix, where fading colors of diagonal elements refer to the disparity of accuracy.

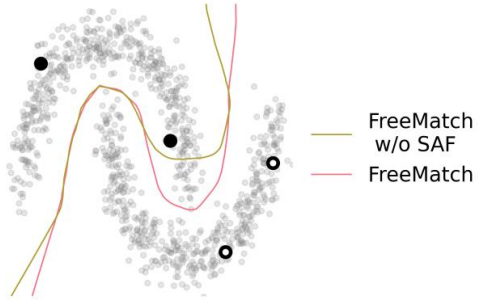
Ablation study

Threshold	CIFAR-10 (40)
τ (FixMatch)	7.47±0.28
$\tau * \mathcal{M}(\beta(c))$ (FlexMatch)	4.97±0.06
$\tau * \mathrm{MaxNorm}(\tilde{p}_t(c))$	5.13±0.03
τ_t (Global)	6.06±0.65
$\tau_t * \mathcal{M}(\beta(c))$	8.40±2.49
CBST	16.65±2.90
RT (AdaMatch)	6.09±0.65
SAT (Global and Local)	**4.92**±0.04

Table 3: Comparison of different thresholding schemes.

Fairness	CIFAR-10 (10)
w/o fairness	10.37±7.70
$U \log \overline{p}$	9.57±6.67
$U \log \mathrm{SumNorm}(\frac{\overline{p}}{h})$	12.07±5.23
DA (AdaMatch)	32.94±1.83
DA (ReMixMatch)	11.06±8.21
SAF	**8.07**±4.24

Table 4: Comparison of different class fairness items.



(b) Self-adaptive fairness

Thanks