Learning Sample-Aware Threshold for Semi-Supervised Learning

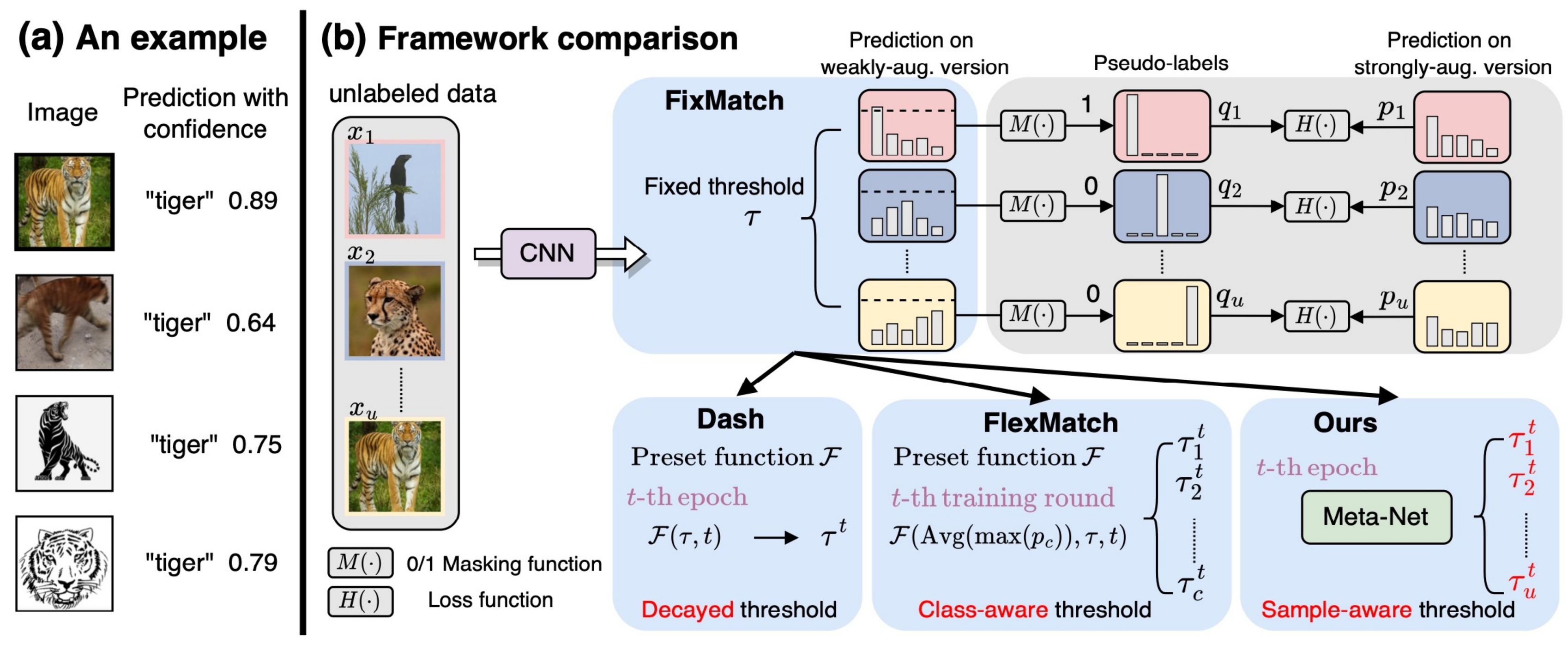
Qi Wei, Lei Feng, Haoliang Sun, Ren Wang, Rundong He, Yilong Yin
Shandong University, Nanyang Technological University
Contact: 1998v7@gmail.com

Machine Learning Journal
ACML 2023

Contributions

- A simple yet effective training framework called Meta-Threshold (Meta-T), which
  - does not leverage prior knowledge to preset adjust function for thresholds
  - contains one hyperparameter, thus does not require complex cross-validation.
- Theoretically provide the convergence of Meta-T which enjoys a rate of $\mathcal{O}(1/_{\epsilon^2})$.
- Meta-T be applied to solve both the conventional and imbalanced SSL tasks.

Motivation and Framework



(a) Motivation: deep models have different learning capabilities for different examples in class *tiger*. Intuitively, setting instance-level thresholds is more logical and beneficial to generate more accurate pseudo-labels for unlabeled instances, further facilitating deep model's learning.

(a) Review of the pseudo-labeling training framework: Meta-T designs a meta-net which dynamically generates a refined confidence threshold for unlabeled example.

Method

☐ Confidence Thresholds in Semi-Supervised Learning

Given an unlabeled data x_m , the training objective is

$$\ell_{x_m} = \mathbb{1}(\max(f(A^{\omega}(x_m); \boldsymbol{w})) > \tau) \cdot H(\hat{y}_m, f(A^s(x_m); \boldsymbol{w}))$$

A^{ω} weak augmentation
A^s strong augmentation
H loss function
τ confidence threshold

☐ Meta-Threshold

The training objective in Meta-T is

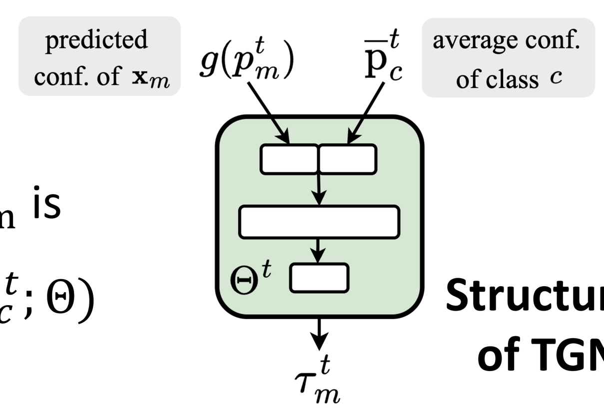
$$\ell_{x_m} = \mathbb{1}(\max(f(A^{\omega}(x_m); \boldsymbol{w})) > \tau_m) \cdot H(\hat{y}_m, f(A^s(x_m); \boldsymbol{w}))$$

Sample-level threshold is produced by a meta-net $\tau_m = V_m(\boldsymbol{w}, \Theta)$

Threshold Generated Network (TGN)

At epoch t, the generated threshold for x_m is

$$\tau_m^t = V(g(f(x_m; \boldsymbol{w})), \bar{p}_c^t; \Theta)$$



Structure of TGN

Bi-level optimization

The **optimal parameters** of two networks can be obtained by minimizing the loss:

$$\mathbf{w}^*(\Theta) = \arg\min_{\mathbf{w}} L_u = \frac{1}{M} \sum\nolimits_{x_m \in D^u} \ell_{x_m}(\mathbf{w}, \Theta)$$

$$\Theta^* = \arg\min_{\Theta} L_{\text{meta}}(\mathbf{w}^*(\Theta)) = \frac{1}{N} \sum\nolimits_{i=1}^{N} H_i(\mathbf{w}^*(\Theta))$$

Solving the meta-optimization problem contains **three steps**:

(1) Formulating learning manner of classifier network

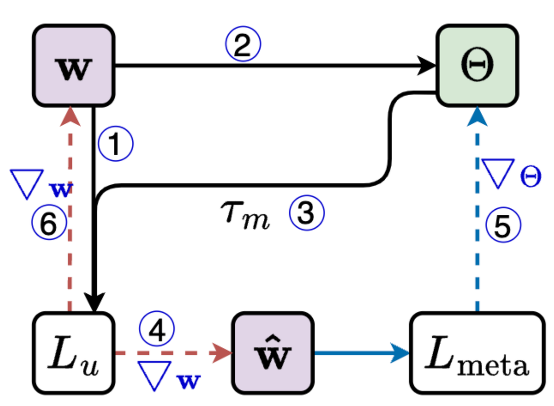
$$\hat{\mathbf{w}}^{(t)}(\Theta) = \mathbf{w}^{(t)} - \alpha \frac{1}{n\mu} \sum\nolimits_{i=1}^{n\mu} \nabla_{\mathbf{w}} \ell_{\mathbf{x}_i}(\mathbf{w}^{(t)}, \Theta^{(t)})$$

(2) Updating parameters Θ of TGN

$$\Theta^{(t+1)} = \Theta^{(t)} - \psi \frac{1}{n} \sum\nolimits_{i=1}^{n} \nabla_{\Theta} H_i(\hat{\mathbf{w}}^{(t)}(\Theta))$$

(3) Updating parameters $\boldsymbol{w}$ of classifier network

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \alpha \frac{1}{n\mu} \sum\nolimits_{i=1}^{n\mu} \nabla_{\mathbf{w}} \ell_{\mathbf{x}_i}(\mathbf{w}^{(t)}, \Theta^{(t+1)})$$



Flowchart of Meta-T

Learning algorithm

Algorithm 1 Learning algorithm of **Meta-T**.

Require: Unlabeled/labeled data D^u/D^l, batch size n, a coefficient μ, max iterations T.
Ensure: Classifier network parameter $\mathbf{w}^{(T)}$.
1: Initialize $\mathbf{w}^{(0)}$ for classifier network and $\Theta^{(0)}$ for TGN.
2: **for** $t = 0$ **to** $T-1$ **do**
3: Random sample $\{(\mathbf{x}_1^l, \mathbf{y}_1^l), ..., (\mathbf{x}_n^l, \mathbf{y}_n^l)\}$ from D^l and $\{\mathbf{x}_1, ..., \mathbf{x}_{(\mu \times n)}\}$ from D^u.
4: Calculate $\hat{\mathbf{w}}^{(t)}(\Theta)$. ▷ Eq. (6)
5: Update $\Theta^{(t+1)}$. ▷ Eq. (7)
6: Update $\mathbf{w}^{(t+1)}$. ▷ Eq. (8)
7: **end for**

Experiments

☐ SOTA performance on eight test benchmarks (typical SSL)

Methods	CIFAR-10 (Wide ResNet-28-2) 40 labels	CIFAR-10 250 labels	CIFAR-10 4000 labels	CIFAR-100 (Wide ResNet-28-8) 400 labels	CIFAR-100 2500 labels	CIFAR-100 10000 labels
Π-Model	-	54.26±3.97	14.01±0.38	-	57.25±0.48	37.88±0.11
VAT	74.66±2.12	41.03±1.79	10.51±0.12	85.20±1.40	46.84±0.79	32.14±0.19
MixMatch	47.54±11.50	11.05±0.86	6.42±0.10	67.61±1.32	39.94±0.37	28.31±0.33
UDA	29.05±5.93	8.82±1.08	4.88±0.18	59.28±0.88	33.13±0.22	24.50±0.25
CoMatch	6.91±1.39	4.91±0.33	-	-	-	-
SimMatch	5.60±1.37	4.84±0.39	**3.96±0.01**	37.81±2.21	**25.07±0.32**	**20.58±0.11**
Pseudo-labeling	-	49.78±0.43	16.09±0.28	-	57.38±0.46	36.21±0.19
FixMatch	11.39±3.37	5.07±0.65	4.26±0.05	48.85±1.75	28.29±0.11	22.60±0.12
Dash	9.16±4.31	4.78±0.12	4.13±0.06	44.83±1.36	27.18±0.21	21.97±0.14
FlexMatch	4.97±0.06	4.98±0.09	4.19±0.01	39.94±1.62	26.49±0.20	21.90±0.15
Meta-T (ours)	**4.39±0.28**	**4.10±0.20**	4.01±0.09	**36.17±1.40**	25.81±0.72	20.74±0.23

Error rates (%) ↓

Methods	SVHN 40 labels	SVHN 250 labels	STL-10 1000 labels
Π-Model	-	18.96±1.92	26.23±0.82
VAT	74.75±3.38	4.33±0.12	37.95±1.12
MixMatch	42.55±14.53	3.98±0.23	10.41±0.61
UDA	52.63±20.51	5.69±2.76	7.66±0.56
ReMixMatch	3.34±0.20	2.92±0.48	5.23±0.45
PL	-	20.21±1.09	27.99±0.83
FixMatch	3.14±1.60	2.64±0.64	5.17±0.63
Dash	3.03±1.59	**2.17±0.10**	3.96±0.25
FlexMatch	8.19±3.20	-	5.77±0.18
Meta-T (ours)	**2.89±0.92**	2.29±0.51	**3.51±0.34**

Top-1 / Top-5 accuracy (%) ↑

	ImageNet 1%	ImageNet 10%	ImageNet 100%
Sup. baseline	25.4 / 48.4	56.4 / 80.4	
FixMatch	53.4 / 74.4	70.8 / 89.0	
CoMatch	66.0 / 86.4	73.6 / 91.6	80.4 / 94.6
SimMatch	67.2 / 87.1	74.4 / 91.6	
Meta-T (ours)	**67.7 / 87.9**	**75.0 / 91.7**	

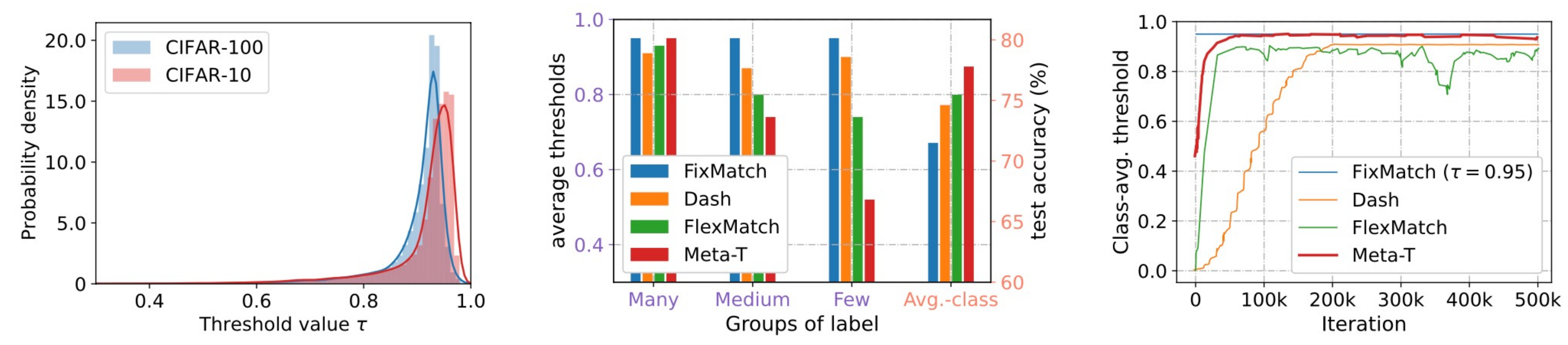
Error rates (%) ↓

	IMDb	Amazon-5	Yelp-5
UAD	18.33±0.61	50.29±4.6	47.49±6.83
FixMatch	7.59±0.28	42.70±0.53	39.56±0.70
FlexMatch	7.80±0.23	42.34±0.62	39.01±0.17
SoftMatch	7.48±0.12	**42.14±0.92**	39.31±0.45
Meta-T(ours)	**7.20±0.20**	42.60±0.41	**38.44±0.37**

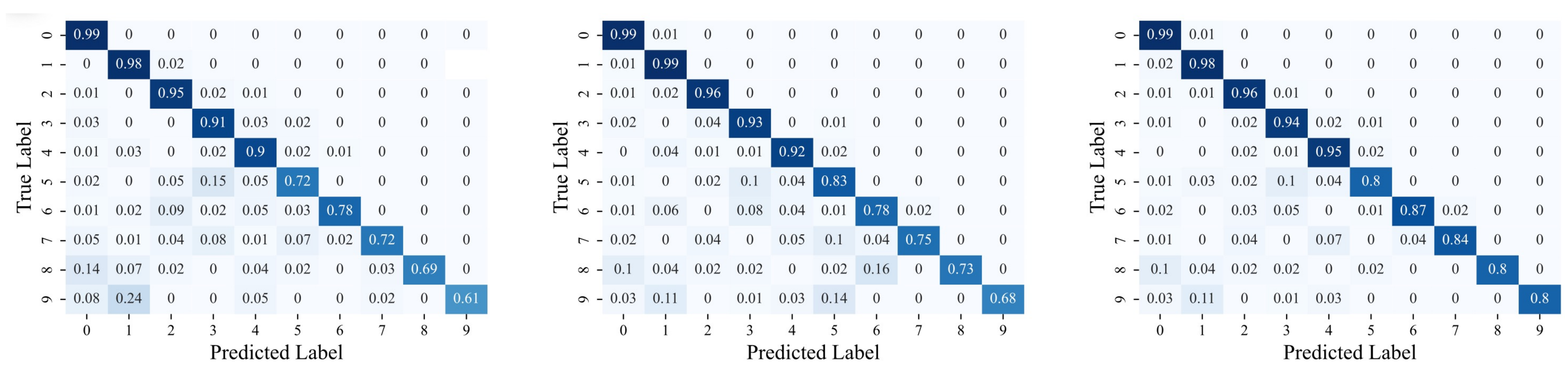
☐ SOTA performance on imbalanced SSL task

Methods	$N_1 = 1500, M_1 = 3000$ $\gamma = 50$	$\gamma = 100$	$\gamma = 150$	$N_1 = 500, M_1 = 4000$ $\gamma = 50$	$\gamma = 100$	$\gamma = 150$
Supervised	65.23±0.05	58.94±0.13	55.63±0.38	51.31±0.34	45.82±0.41	40.90±0.39
cRT	67.82±0.14	63.43±0.45	59.56±0.44	56.28±1.45	48.11±0.79	45.02±1.08
LDAM	68.91±0.10	63.15±0.24	58.68±0.30	56.41±0.92	49.27±0.88	45.10±0.75
MixMatch	73.59±0.46	65.03±0.26	62.71±0.29	65.32±1.20	56.41±1.96	52.38±1.88
ReMixMatch	78.96±0.29	72.88±0.12	68.61±0.40	76.83±0.98	70.12±1.23	59.58±1.30
DARP	81.60±0.31	75.23±0.14	69.31±0.26	76.72±0.46	69.41±0.50	61.23±0.31
CReST	82.03±0.26	75.08±0.41	69.84±0.39	76.18±0.36	69.50±0.70	60.81±0.55
Adsh	83.38±0.06	76.52±0.35	71.49±0.30	**79.27±0.38**	70.97±0.46	62.04±0.51
FixMatch	79.10±0.14	71.50±0.31	68.47±0.15	77.34±0.96	68.45±0.94	60.10±0.82
Dash	81.93±0.10	74.62±0.26	72.29±0.42	77.90±0.39	70.41±0.27	62.11±0.32
FlexMatch	82.86±0.25	75.47±0.41	70.62±0.30	78.69±0.50	71.80±0.29	62.85±0.39
Meta-T (ours)	**83.94±0.12**	**77.80±0.39**	**73.07±0.58**	78.41±0.22	**72.40±0.42**	**64.46±0.60**

☐ Effectiveness analysis

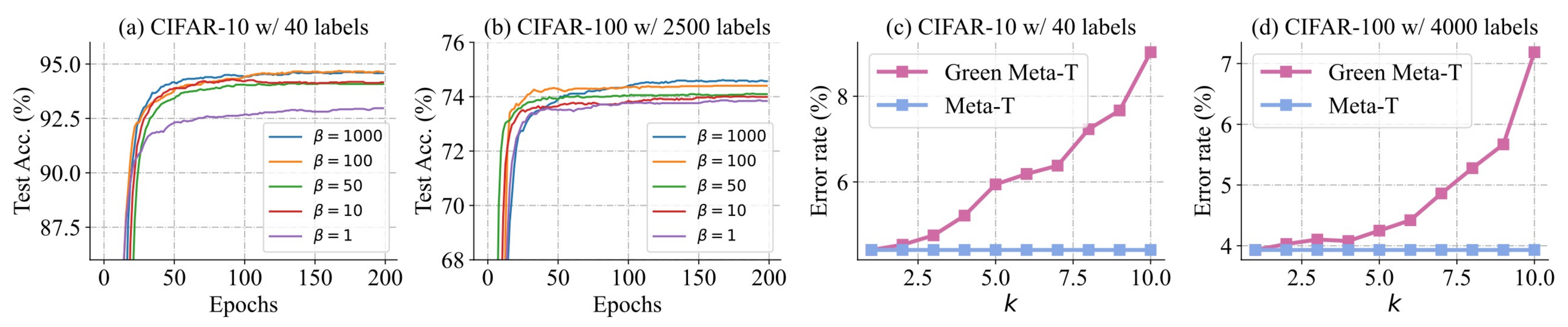


(a) Density of generated sample-level thresholds (b) Threshold generation under imbalanced SSL setting (c) Dynamic curves of generated thresholds



(a) FixMatch (b) FlexMatch (c) Meta-T

☐ Sensitivity analysis



Reference

[1] Zhang *et al.* Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. NIPS 2021
[2] Xu *et al.* Dash: Semi-supervised learning with dynamic thresholding. ICML 2021