模式分析与机器智能
工业和信息化部重点实验室
MIIT Key Laboratory of
Pattern Analysis & Machine Intelligence

ParNeC 模式识别与神经计算研究组
PAttern Recognition and NEural Computing

CHMATCH: Contrastive Hierarchical Matching and Robust AdaptiveThreshold Boosted Semi-Supervised Learning

Jianlong Wu[1], Haozhe Yang[2], Tian Gan[2]*, Ning Ding[3], Feijun Jiang[3], Liqiang Nie[1]*
[1] School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen)
[2] School of Computer Science and Technology, Shandong University [3] Alibaba Group
jlwu1992@pku.edu.cn, sailist@outlook.com, gantian@sdu.edu.cn,
{yuji.dn, feijun.jiangfj}@alibaba-inc.com, nieliqiang@gmail.com

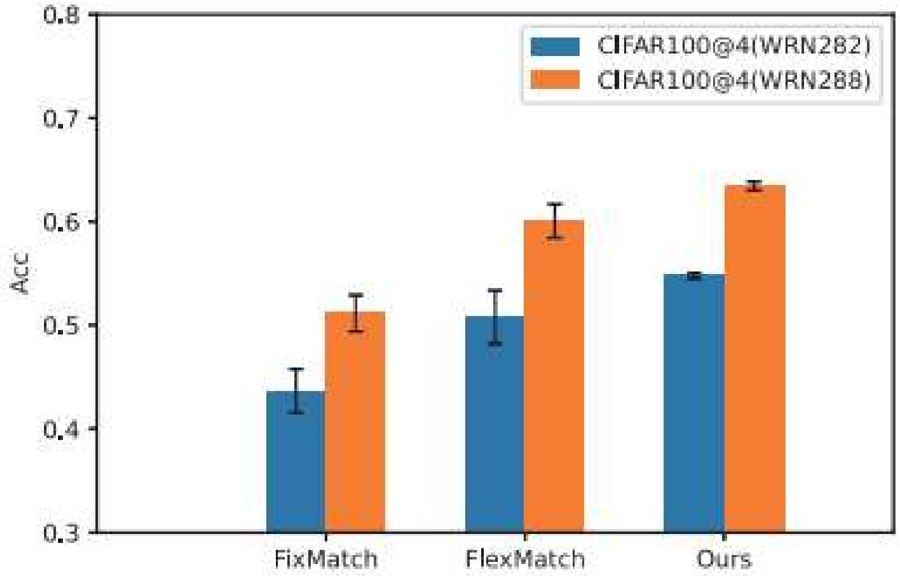
CVPR 2023

Related work

ParNeC 模式识别与神经计算研究组 PAttern Recognition and NEural Computing

FixMa

FlexM



(a) Results on CIFAR-100

(1) The results of both FixMatch and FlexMatch are unstable and of large variances, which is shown in Figure 1(a), especially when there are only a small amount of labeled samples;

(2) Only instance-level consistency is investigated, which neglects inter-class relationship and may make the learned feature indiscriminative.

CoMatch(2021)

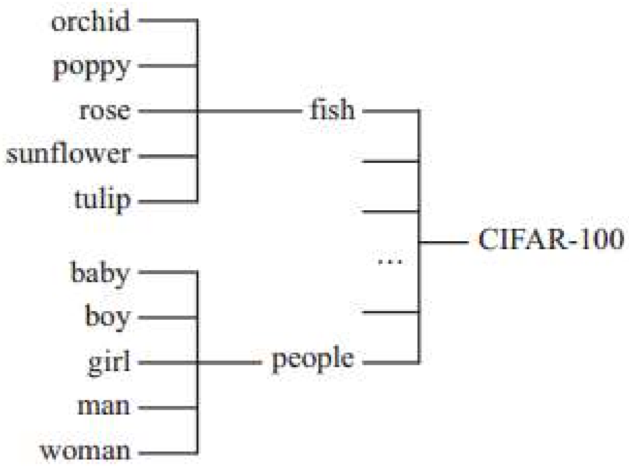
CoMatch effectively utilizes both labeled and unlabeled data to enhance the accuracy and robustness of the model by integrating dual representation learning, memory-smoothed pseudo-labeling, and graph-based contrastive learning.

Method

ParNeC 模式识别与神经计算研究组 PAttern Recognition and NEural Computing

How to solve the problem of high variance in a model?

Introduce additional supervisory information.



(c) Hierarchical label structure

Dual Classification Heads: CHMatch concurrently learns a fine-grained classification head hf (.) and a coarse-grained classification head
hc (.).

The coarse-grained classification head can provide certain guidance to the fine-grained classification head.

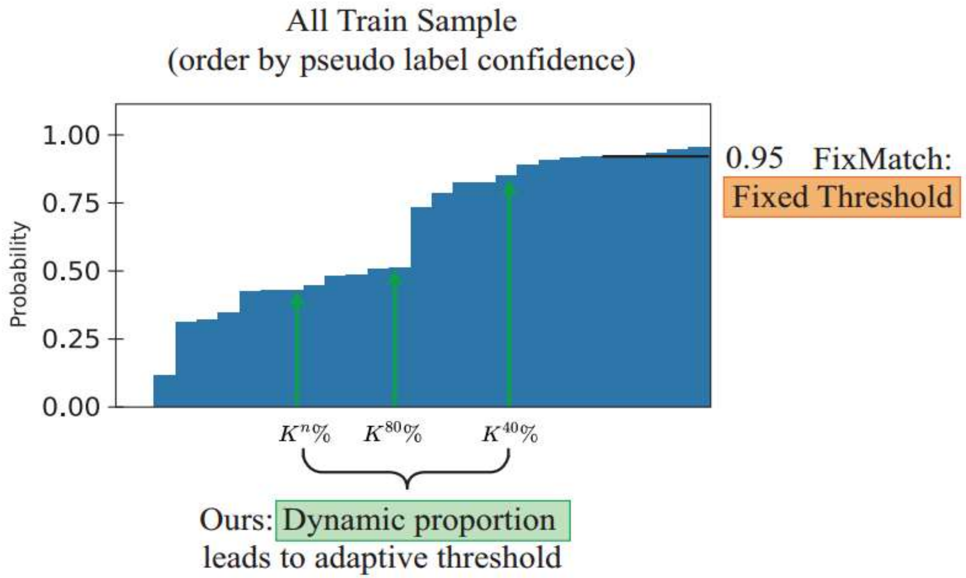
In practice, categories have a hierarchical structure,which is often neglected by existing methods. But it contains extra supervision signals for network training.

Method

ParNeC 模式识别与神经计算研究组 PAttern Recognition and NEural Computing

$$\mathcal{L}_u^f + \mathcal{L}_u^c = \frac{1}{\mu B}\sum_{b=1}^{\mu B}\Big(\mathbb{1}(\max(q_b^f) \geq \tau_f)H(\hat{q}_b^f, p(y|\mathcal{A}(u_b))) + \mathbb{1}(\max(q_b^c) \geq \tau_c)H(\hat{q}_b^c, p(y|\mathcal{A}(u_b)))\Big)$$

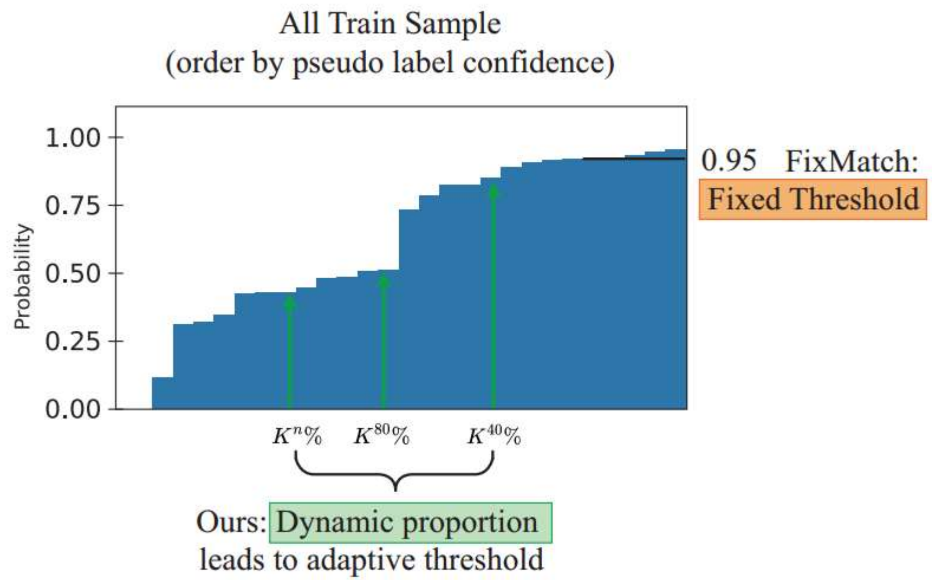
At each epoch,we hope that a certain percentage K% of the samples are chosen for pseudo-label learning. For example, at the earlystage of the training, K% should be small to ensure that theselected samples are of high confidence to guide the network training. In contrast, at the end of the training, K% should be large enough to guarantee that most samples can join the training.



(b) Different threshold learning strategy

Adaptive Threshold Learning: For each classification head, CHMatch develops robust adaptive thresholds τ_c and τ_f to select highly confident samples.

Method



(b) Different threshold learning strategy

We construct a memory-bank to save the maximum probability of the previous N weakly augmented samples as Q_{MB}^{f} = $\{q_1^f, q_2^f, ...q_N^f\}$, where N is much larger than the mini-batch size. At each epoch, we hope that a certain percentage K% of the samples are chosen for pseudo-label learning.

Compute τ_f = percentile(Q_{MB}^{f}; K)

$$|Q_{MB}^{f} > \tau_f|/N = K\%,$$

```
type_map = {
    'linear': params.SCHE.Linear,
    'cos': params.SCHE.Cos,
    'exp': params.SCHE.Exp,
    'log': params.SCHE.Log,
}
if params.k_type != 'fixed':
    sche_fn = type_map[params.k_type]
    self.k_sche = sche_fn(params.mlen * params.min_proportion,
                          params.mlen * params.max_proportion,
                          left=0, right=params.k_inc_epoch)
```

Method

Consider the inter-class relationships.

In a semi-supervised scenario, how can we accurately identify samples of the same class?

Consistency regularization only leads to instance-level correspondence in the feature space, resulting in unclear margins between classes. Using contrastive learning can effectively solve this problem.

$$W_{bj}^{f} = \begin{cases} 1 & \text{if } \hat{q}_b^f = \hat{q}_j^f, \\ 0 & \text{otherwise,} \end{cases}$$

$$l_b^{simclr} = -\log \frac{\exp(f(\alpha(u_b)) \cdot f(\alpha(u_b))/t)}{\sum_{j=1, j\neq b}^{\mu B} \exp(f(\alpha(u_b)) \cdot f(\alpha(u_j))/t)}$$

$$W_{bj} = \begin{cases} 1 & \text{if } W_{bj}^{f} = 1 \text{ and } W_{bj}^{c} = 1, \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathcal{L}_u^{ctl} = -\frac{1}{\mu B} \sum_{b=1}^{\mu B} \left(\frac{1}{\sum_j W_{bj}} \log \frac{\sum_{j=1}^{\mu B} W_{bj} \exp((z_b \cdot z_j')/t)}{\sum_{j=1}^{\mu B} (1 - W_{bj}) \exp(z_b \cdot z_j')/t)} \right)$$

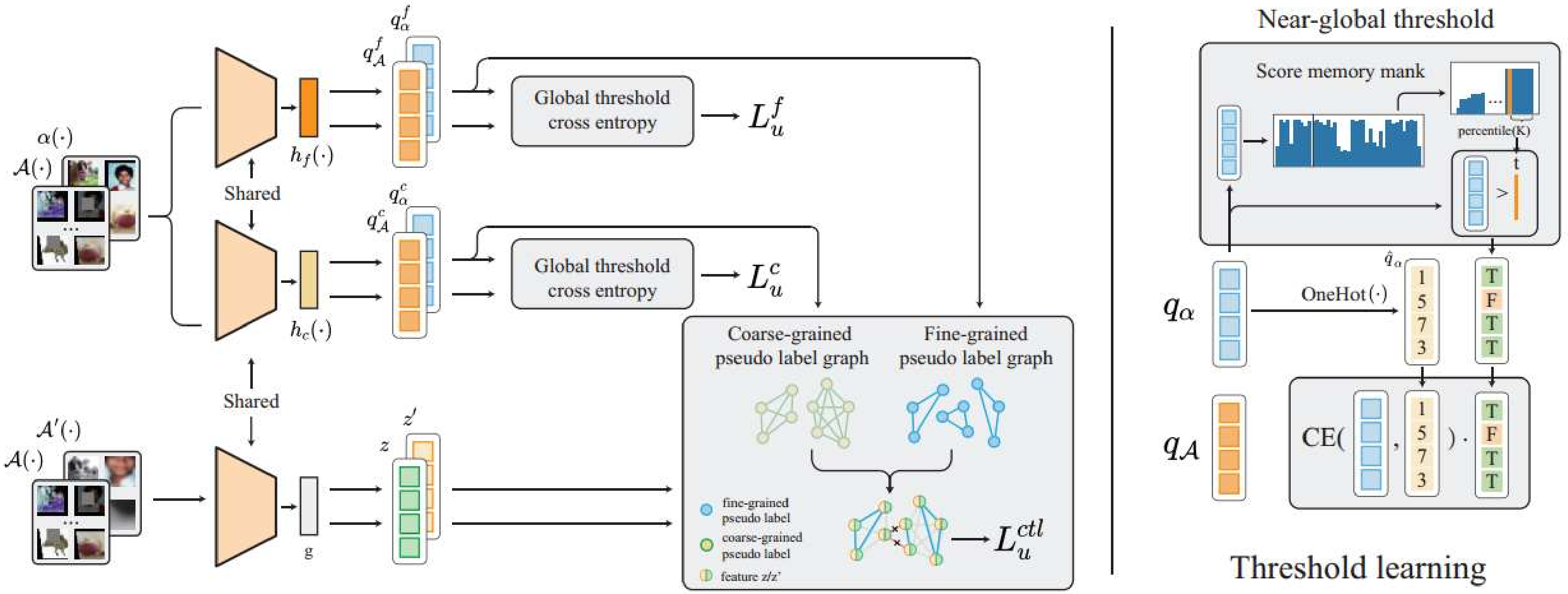


Figure 2. Framework of our CHMatch. (Left) Besides the general classification head $\mathcal{L}_u^f$ in FixMatch, we add a coarse-grained classification head $\mathcal{L}_u^c$ and a projection head $g(\cdot)$. We utilize the hierarchical label information, performing graph matching between fine-grained and coarse-grained pseudo label graphs to guide contrastive feature learning. (Right) A memory-back based strategy is proposed to learn robust adaptive threshold to guide instance-level prediction matching.

Experiment

Table 1. Error rates comparison on CIFAR-10, CIFAR-100, and STL-10.

Methods	CIFAR-10			CIFAR-100@WRN-28-2			CIFAR-100@WRN-28-8			STL-10
Label Amount	40	250	1000	400	2500	10000	400	2500	10000	1000
MixMatch (NeurIPS'19)	36.19± 6.48	13.63± 0.59	6.66± 0.26	-	-	-	67.61± 1.32	39.94± 0.37	28.31± 0.33	61.98± 8.29
FixMatch (NeurIPS'20)	13.91± 3.37	5.07± 0.65	4.26± 0.05	56.34± 2.12	34.53± 0.31	27.89± 0.10	48.85± 1.75	28.29± 0.11	22.60± 0.12	34.62± 0.42
CoMatch (ICCV'21)	6.91± 1.39	4.91± 0.33	4.56± 0.20	58.46± 2.31	36.84± 0.43	31.6± 0.14	41.89± 2.34	28.37± 0.35	**20.86± 0.36**	**20.20± 0.38**
FlexMatch (NeurIPS'21)	**4.97± 0.06**	4.98± 0.09	**4.19± 0.01**	**49.23± 2.58**	**32.51± 0.20**	**26.58± 0.11**	**39.94± 1.62**	**26.49± 0.20**	21.90± 0.15	-
DP-SSL (NeurIPS'21)	6.54± 0.98	**4.78± 0.26**	**4.23± 0.20**	-	-	-	43.17± 1.29	28.00± 0.79	22.24± 0.31	-
CHMatch (ours)	**5.98± 0.19**	**4.91± 0.13**	4.48± 0.10	**45.23± 0.28**	**31.32± 0.47**	**24.84± 0.27**	**36.57± 0.41**	**24.10± 0.10**	**19.92± 0.29**	**10.36± 0.31**

Experiment

Table 2. Error rates results on the ImageNet subset.

Method	Top1		Top5	
Label fraction	1%	10%	1%	10%
MixMatch	-	-	-	-
FixMatch	60.81	34.33	35.84	14.83
CoMatch	42.88	26.48	17.99	9.23
FlexMatch	54.37	29.82	31.61	12.27
DP-SSL	-	-	-	-
CHMatch(ours)	**34.18**	**24.17**	**12.33**	**7.64**

Table 3. Effect of each module.

Modules					Error rate
Graph matching	Coarse label	Fixed threshold	Fixed proportion	Dynamic proportion	
✓	✓	✓			50.14
✓	✓		✓		49.56
				✓	48.83
	✓			✓	47.14
✓	✓			✓	**45.23**

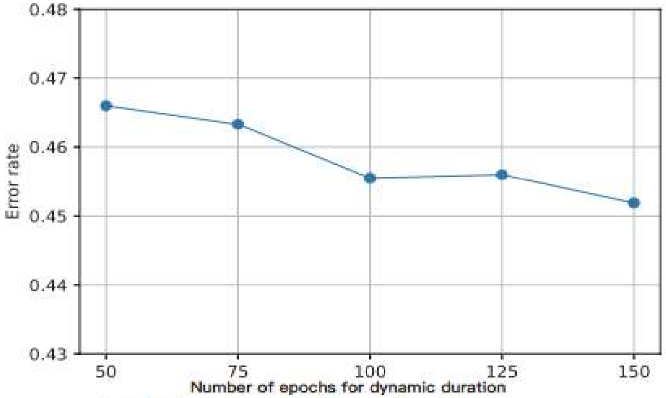
Table 4. Error rate on CIFAR100 under WRN-28-2 with different weights. We vary each parameter and fix other weights as 1.

	0.75	1	1.5	2
α	46.37	45.23	46.51	44.53
β	46.60	45.23	46.15	44.96
γ	45.61	45.23	44.99	45.05

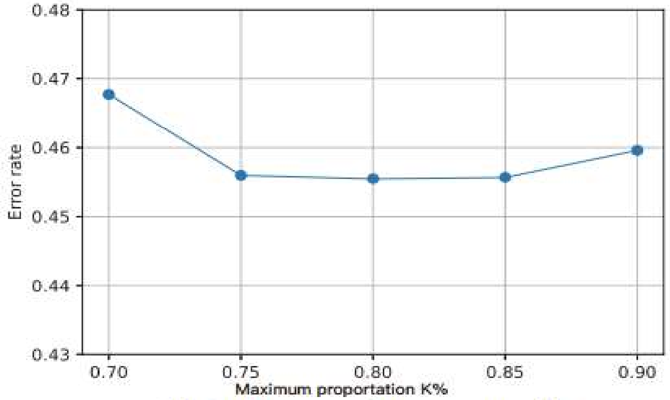
Table 5. Inconsistent rate of coarse labels on CIFAR100.

Method	Backbones	Label Amount		
		400	2500	10000
Supervised	WRN-28-2	78.63%	48.79%	29.99%
CHMatch w/o Graph Matching	WRN-28-2	30.94%	19.58%	16.15%
CHMatch	WRN-28-2	27.91%	17.52%	14.56%
CHMatch	WRN-28-8	23.75%	12.68%	8.56%

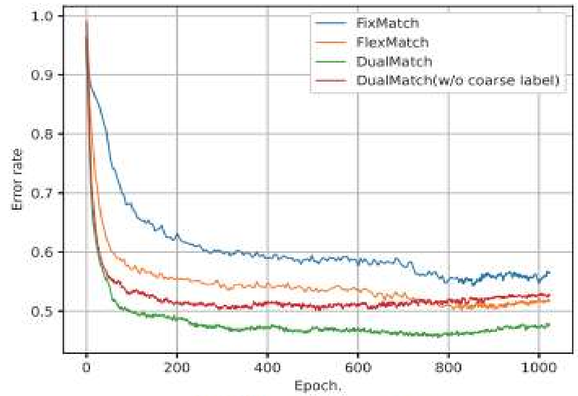
Experiment



(a) Influence of dynamic duration

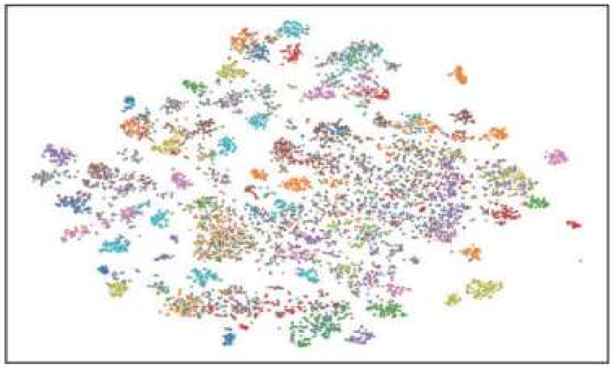


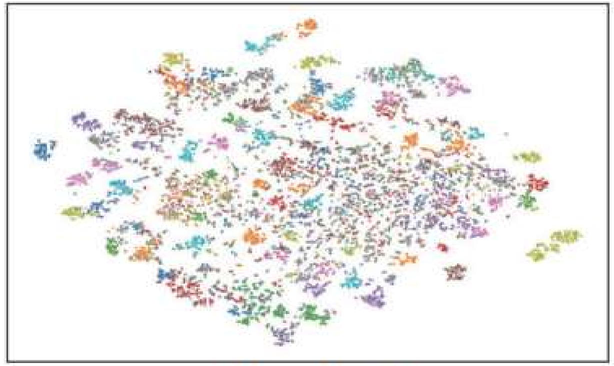
(b) Influence of max proportion



(c) Top-1 error rate

Figure 3. Parameter sensitivity and convergence analysis on CIFAR-100.


(a) FixMatch


(b) FlexMatch

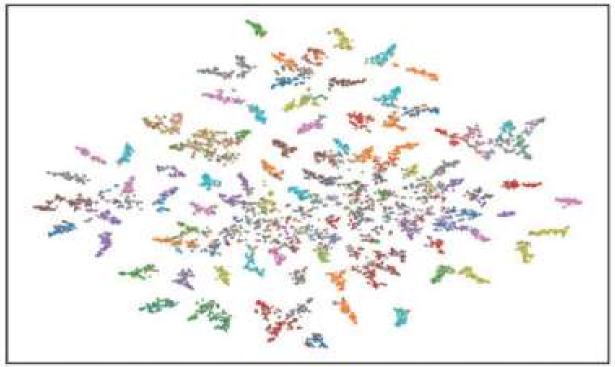

(c) DualMatch

Figure 4. t-SNE visualization for feature representation on CIFAR-100.

Thanks