PiCO: Contrastive Label Disambiguation for Partial Label Learning

Haobo Wang[1] Ruixuan Xiao[1] Yixuan Li[2] Lei Feng[34]
Gang Niu[4] Gang Chen[1] Junbo Zhao[1*]

[1]Zhejiang University [2]University of Wisconsin-Madison
[3]Chongqing University [4]RIKEN

ICLR 2022 Oral

Motivation

Label Ambiguity in Annotation

A dog image $\boldsymbol{x}_i$ with
$Y_i = \{$Husky, Malamute, Samoyed$\}$

Q: What the dog it is?
A: Siberian Husky?
Emm... Malamute?

Annotation difficulty:
data annotation in the real-world can naturally be subject to inherent label ambiguity and noise.

Solution:

- **Aggressive**:
  random selection, but introduce label noise
- **Passive**:
  leave it being unlabeled, but ignore the fact that some labels are more likely to be true

Motivation

Partial Label Learning

Supervised → Malamute
True Label

Partial Label → Malamute | Husky
Candidate Labels

- **Partial Label Learning (PLL)** [CST11]
  - Each training example is equipped with a set of candidate labels instead of the exact ground-truth label.
- **A Non-trivial Dilemma of PLL**
  - **Representation Learning**: The inherent label uncertainty can undesirably manifest in the representation learning process
  - **Classifier Training**: The quality of representation prevents effective label disambiguation

PiCO

An Overview of Main Contributions

- **Methodology**: A synergistic PLL framework that leverages contrastive learning for enhanced representation and improved label disambiguation
- **Experiments**: Establishes the *SOTA* performance on PLL
- **Theory**: We interpret PiCO from the EM-algorithm perspective

PiCO
Notations

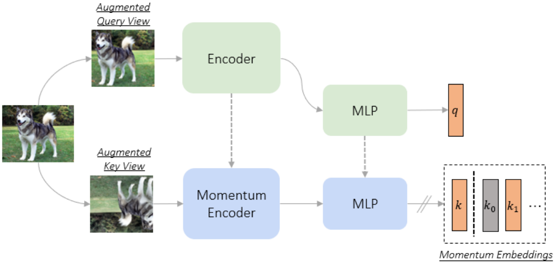
- **Input**: training dataset $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i)\}_{i=1}^{n}$
  - an image $\boldsymbol{x}_i \in \mathcal{X}$
  - a *candidate label set* $Y_i \subset \mathcal{Y} = \{1, 2, ..., C\}$
  - (**assumption**) Y_i contains the true label y_i, i.e., $y_i \in Y_i$
- **Training:** learning with label disambiguation
  - each image $\boldsymbol{x}_i$ is assigned a pseudo target $\boldsymbol{s}_i \in [0, 1]^C$
  - per-sample loss:

$$\mathcal{L}_{\text{cls}}(f; \boldsymbol{x}_i, Y_i) = \sum\nolimits_{j=1}^{C} -s_{i,j} \log(f^j(\boldsymbol{x}_i))$$

$$\text{s.t.} \quad \sum\nolimits_{j \in Y_i} s_{i,j} = 1 \text{ and } s_{i,j} = 0, \forall j \notin Y_i,$$

- **Output**: a predictor $f : \mathcal{X} \rightarrow [0, 1]^C$

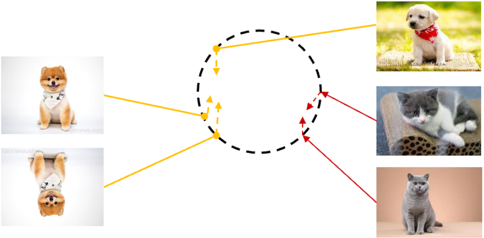
- **MoCo-style [HFW+20] Backbone**
  - Given each sample $(\boldsymbol{x}, Y)$, a query view and a key view are generated by randomized data augmentation $\text{Aug}(\boldsymbol{x})$.
- **Embedding Pool Generation**
  - Get a pair of normalized embeddings $\boldsymbol{q} = g(\text{Aug}_q(\boldsymbol{x}))$ and $\boldsymbol{k} = g'(\text{Aug}_k(\boldsymbol{x}))$.
  - Generate an embdding pool: $A = B_q \cup B_k \cup \text{queue}$



- **Contrastive learning helps representation learning**
  - Pulls examples from the same class close together
  - Push away examples from different classes
- **Per-Sample Contrastive Loss**

$$\mathcal{L}_{\text{cont}}(g; \boldsymbol{x}, \tau, A) = -\frac{1}{|P(\boldsymbol{x})|} \sum_{\boldsymbol{k}_+ \in P(\boldsymbol{x})} \log \frac{\exp(\boldsymbol{q}^\top \boldsymbol{k}_+ / \tau)}{\sum_{\boldsymbol{k}' \in A(\boldsymbol{x})} \exp(\boldsymbol{q}^\top \boldsymbol{k}' / \tau)},$$

- **Challenge**: how to construct the positive set $P(\boldsymbol{x})$.



- **Positive Set Selection**
  - Use the predicted label $\tilde{y} = \arg\max_{j \in Y} f^j(\text{Aug}_q(\boldsymbol{x}))$ from the classifier

$$P(\boldsymbol{x}) = \{\boldsymbol{k}' | \boldsymbol{k}' \in A(\boldsymbol{x}), \tilde{y}' = \tilde{y}\}$$

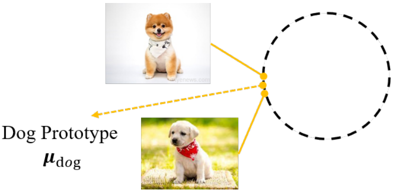
  - Simple but effective; can be theoretically justified (Section 5)
- **The Overall Loss**
  - Jointly train the classifier as well as the contrastive network

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda \mathcal{L}_{\text{cont}}.$$

  - The label ambiguity remains unsolved

- **Prototype-based Disambiguation**
  - Keep a *prototype* embedding vector $\boldsymbol{\mu}_c$ corresponding to each class $c \in \{1, 2, ..., C\}$ as representative embedding vectors
  - If an example is close to the j-th prototype, it also tends to have the j-th label as the ground-truth
- **Pseudo Target Updating by Moving-Average**

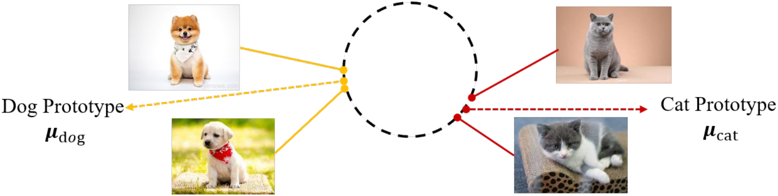
$$\boldsymbol{s} = \phi \boldsymbol{s} + (1 - \phi)\boldsymbol{z}, \quad z_c = \begin{cases} 1 & \text{if } c = \arg\max_{j \in Y} \boldsymbol{q}^\top \boldsymbol{\mu}_j, \\ 0 & \text{else} \end{cases}$$

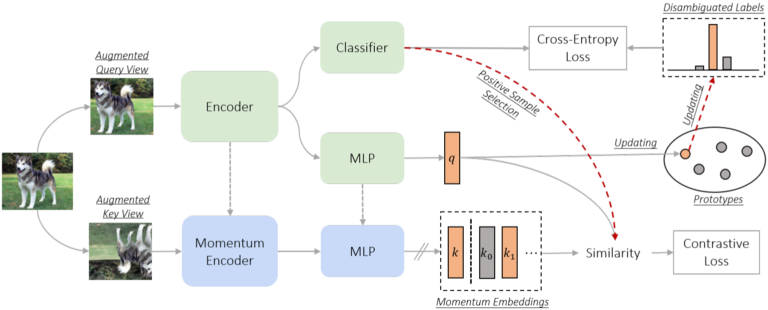


Candidate	dog	cat
Pseudo	**0.65**	0.35
Prototype	**1**	0
Updating	↑	↓

- **Efficient Prototype Updating**
  - Momentum-updating the class prototype vectors

$$\boldsymbol{\mu}_c = \text{Normalize}(\gamma \boldsymbol{\mu}_c + (1-\gamma)\boldsymbol{q}),$$
$$\text{if } c = \arg\max_{j \in Y} f^j(\text{Aug}_q(\boldsymbol{x}))),$$





- Synergy between Contrastive Learning and Label Disambiguation
  - The clustering effect of CL benefits label disambiguation
  - Better label disambiguation reciprocates the CL part by accurate positive set construction

- PiCO achieves comparable results to the fully supervised learning with less label ambiguity

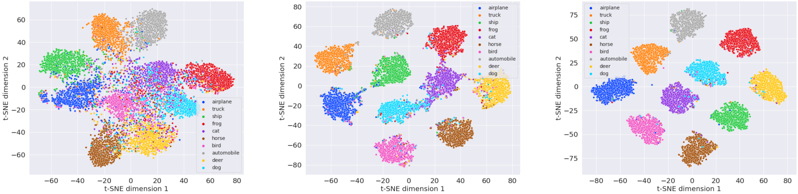
Table 1: Accuracy comparisons on benchmark datasets. Bold indicates superior results. Notably, PiCO achieves comparable results to the fully supervised learning (less than 1% in accuracy with ≈ 1 false candidate).

Dataset	Method	$q = 0.1$	$q = 0.3$	$q = 0.5$
CIFAR-10	PiCO (ours)	**94.39** $\pm$ 0.18%	**94.18** $\pm$ 0.12%	**93.58** $\pm$ 0.06%
	LWS	90.30 $\pm$ 0.60%	88.99 $\pm$ 1.43%	86.16 $\pm$ 0.85%
	PRODEN	90.24 $\pm$ 0.32%	89.38 $\pm$ 0.31%	87.78 $\pm$ 0.07%
	CC	82.30 $\pm$ 0.21%	79.08 $\pm$ 0.07%	74.05 $\pm$ 0.35%
	MSE	79.97 $\pm$ 0.45%	75.64 $\pm$ 0.28%	67.09 $\pm$ 0.66%
	EXP	79.23 $\pm$ 0.10%	75.79 $\pm$ 0.21%	70.34 $\pm$ 1.32%
	Fully Supervised		94.91 $\pm$ 0.07%	
Dataset	**Method**	$q = 0.01$	$q = 0.05$	$q = 0.1$
CIFAR-100	PiCO (ours)	**73.09** $\pm$ 0.34%	**72.74** $\pm$ 0.30%	**69.91** $\pm$ 0.24%
	LWS	65.78 $\pm$ 0.02%	59.56 $\pm$ 0.33%	53.53 $\pm$ 0.08%
	PRODEN	62.60 $\pm$ 0.02%	60.73 $\pm$ 0.03%	56.80 $\pm$ 0.29%
	CC	49.76 $\pm$ 0.45%	47.62 $\pm$ 0.08%	35.72 $\pm$ 0.47%
	MSE	49.17 $\pm$ 0.05%	46.02 $\pm$ 1.82%	43.81 $\pm$ 0.49%
	EXP	44.45 $\pm$ 1.50%	41.05 $\pm$ 1.40%	29.27 $\pm$ 2.81%
	Fully Supervised		73.56 $\pm$ 0.10%	

Experiments

Representation Visualization

- PiCO learns more distinguishable representations



(a) Uniform features (b) PRODEN features (c) PiCO features (ours)

- Each component plays a vital role in PiCO

Table 2: Ablation study on CIFAR-10 with $q = 0.5$ and CIFAR-100 with $q = 0.05$.

Ablation	$\mathcal{L}_{\text{cont}}$	Label Disambiguation	CIFAR-10 ($q = 0.5$)	CIFAR-100 ($q = 0.05$)
PiCO	✓	Ours	**93.58**	**72.74**
PiCO w/o Disambiguation	✓	Uniform Pseudo Target	84.50	64.11
PiCO w/o $\mathcal{L}_{\text{cont}}$	✗	Uniform Pseudo Target	76.46	56.87
PiCO with $\phi = 0$	✓	Soft Prototype Probs	91.60	71.07
PiCO with $\phi = 0$	✓	One-hot Prototype	91.41	70.10
PiCO	✓	MA Soft Prototype Probs	81.67	63.75

Theoretical Analysis

Why PiCO improves partial label learning?

- **The Clustering Effect of Contrastive Learning**
  - CL Loss decomposing: (a) alignment; (b) uniformity [WI20]

$$\tilde{\mathcal{L}}_{\text{cont}}(g;\tau,\mathcal{D}) = \underbrace{\frac{1}{n}\sum_{\boldsymbol{x}\in\mathcal{D}}\left\{-\frac{1}{|P(\boldsymbol{x})|}\sum_{\boldsymbol{k}_+\in P(\boldsymbol{x})}(\boldsymbol{q}^\top \boldsymbol{k}_+/\tau)\right\}}_{\text{(a)}} + \underbrace{\frac{1}{n}\sum_{\boldsymbol{x}\in\mathcal{D}}\left\{\log\sum_{\boldsymbol{k}'\in A(\boldsymbol{x})}\exp(\boldsymbol{q}^\top \boldsymbol{k}'/\tau)\right\}}_{\text{(b)}}.$$

  - Split the dataset to C subsets $S_j \in \mathcal{D}_C$ having the same prediction

$$\text{(a)} \approx \frac{1}{\tau n}\sum\nolimits_{S_j\in\mathcal{D}_C}\sum\nolimits_{\boldsymbol{x}\in S_j}||g(\boldsymbol{x})-\boldsymbol{\mu}_j||^2 + K,$$

  - K is a constant and $\boldsymbol{\mu}_j$ is the mean center of S_j

Theoretical Analysis

Why PiCO improves partial label learning?

Assumption 1

All labels y_i in the candidate label set have the same probability of generating Y_i, but no label outside of Y_i can generate Y_i, i.e. $P(Y_i|y_i) = \hbar(Y_i)$ if $y_i \in Y_i$ else 0. Here $\hbar(\cdot)$ is some function making it a valid probability distribution.

- **Likelihood Maximization**
  - Establish the relationship between the candidate and the ground-truth label by the above assumption

$$\arg\max_{\theta} \sum_{i=1}^{n} \log P(Y_i, \boldsymbol{x}_i|\theta) = \arg\max_{\theta} \sum_{i=1}^{n} \log \sum_{y_i \in Y_i} P(\boldsymbol{x}_i, y_i|\theta) + \cancel{\sum_{i=1}^{n} \log(\hbar(Y_i))}$$

Theoretical Analysis

Why PiCO improves partial label learning?

- **An Expectation-Maximization Perspective (E-Step)**
  - Define some auxiliary distributions $\pi_i^j \geq 0$ $(1 \leq i \leq n, 1 \leq j \leq C)$ such that $\pi_i^j = 0$ if $j \notin Y_i$ and $\sum_{j \in Y_i} \pi_i^j = 1$

$$\arg\max_{\theta} \sum_{i=1}^{n} \log P(Y_i, \boldsymbol{x}_i|\theta) \geq \arg\max_{\theta} \sum_{i=1}^{n} \sum_{y_i \in Y_i} \pi_i^{y_i} \log \frac{P(\boldsymbol{x}_i, y_i|\theta)}{\pi_i^{y_i}}.$$

  - The condition that inequality holds with equality is,

$$\pi_i^{y_i} = \frac{P(\boldsymbol{x}_i, y_i|\theta)}{\sum_{y_i \in Y_i} P(\boldsymbol{x}_i, y_i|\theta)} = \frac{P(\boldsymbol{x}_i, y_i|\theta)}{P(\boldsymbol{x}_i|\theta)} = P(y_i|\boldsymbol{x}_i, \theta), \quad (1)$$

  - $\pi_i^{y_i}$ is the posterior class probability
- **The Corresponding Component in PiCO**
  - Positive set selection by using classifier's output

Theoretical Analysis

Why PiCO improves partial label learning?

Theorem 1

Assume data from the same class in the contrastive output space follow a d-variate von Mises-Fisher (vMF) distribution whose probabilistic density is given by $f(\boldsymbol{x}|\bar{\boldsymbol{\mu}}_i, \kappa) = c_d(\kappa)e^{\kappa\bar{\boldsymbol{\mu}}_i^\top g(\boldsymbol{x})}$*, where* $\bar{\boldsymbol{\mu}}_i = \boldsymbol{\mu}_i/||\boldsymbol{\mu}_i||$ *is the mean direction,* κ *is the concentration parameter, and* $c_d(\kappa)$ *is the normalization factor. We further assume a uniform class prior* $P(y_i = j) = 1/C$*. Let* $n_j = |S_j|$*. Then, optimizing Eq. (9) and Eq. (10) equal to maximize* R_1 *and* R_2 *below, respectively.*

$$R_1 = \sum_{S_j \in \mathcal{D}_C} \frac{n_j}{n}||\boldsymbol{\mu}_j||^2 \leq \sum_{S_j \in \mathcal{D}_C} \frac{n_j}{n}||\boldsymbol{\mu}_j|| = R_2. \tag{2}$$

- **An Expectation-Maximization Perspective (M-Step)**
  - Minimizing contrastive loss (alignment term) also maximizes a lower bound of likelihood (Theorem 1)

Summary

Why PiCO improves partial label learning?

- We propose PiCO, a coherent and synergistic framework that pioneers the exploration of contrastive learning for partial label learning.
- PiCO achieves SOTA performance on common PLL benchmarks as well as new fine-grained classification tasks.
- Theoretical analysis shows that PiCO can be interpreted from an EM-algorithm perspective. In particular, we show contrastive learning manifests a clustering effect in the embedding space, and is beneficial for weakly-supervised learning.

References I

[CST11] Timothée Cour, Benjamin Sapp, and Ben Taskar, *Learning from partial labels*, J. Mach. Learn. Res. **12** (2011), 1501–1536.

[HFW+20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick, *Momentum contrast for unsupervised visual representation learning*, CVPR, Computer Vision Foundation / IEEE, 2020, pp. 9726–9735.

[WI20] Tongzhou Wang and Phillip Isola, *Understanding contrastive representation learning through alignment and uniformity on the hypersphere*, ICML, Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 9929–9939.