Ambiguity-Induced Contrastive Learning for Instance-Dependent Partial Label Learning

Shi-Yu Xia[1,2], **Jiaqi Lv**[3], **Ning Xu**[1,2] and **Xin Geng**[1,2*]
[1]School of Computer Science and Engineering, Southeast University, Nanjing 211189, China
[2]Key Laboratory of Computer Network and Information Integration (Ministry of Education), Southeast University, Nanjing 211189, China
[3]RIKEN Center for Advanced Intelligence Project
shiyu_xia@seu.edu.cn, is.jiaqi.lv@gmail.com, {xning, xgeng}@seu.edu.cn

Abstract

Partial label learning (PLL) learns from a typical weak supervision, where each training instance is labeled with a set of ambiguous candidate labels (CLs) instead of its exact ground-truth label. Most existing PLL works directly eliminate, rather than exploiting the label ambiguity, since they explicitly or implicitly assume that incorrect CLs are noise independent of the instance. While a more practical setting in the wild should be *instance-dependent*, namely, the CLs depend on both the true label and the instance itself, such that each CL may describe the instance from some sensory channel, thereby providing some noisy but really valid information about the instance. In this paper, we leverage such additional information acquired from the ambiguity and propose *AmBiguity-induced contrastive LEarning* (ABLE) under the framework of contrastive learning. Specifically, for each CL of an anchor, we select a group of samples currently predicted as that class as *ambiguity-induced positives*, based on which we synchronously learn a *representor* (RP) that minimizes the *weighted sum* of contrastive losses of all groups and a *classifier* (CS) that minimizes a classification loss. Although they are circularly dependent: RP requires the ambiguity-induced positives on-the-fly induced by CS, and CS needs the first half of RP as the representation extractor, ABLE still enables RP and CS to be trained simultaneously within a coherent framework. Experiments on benchmark datasets demonstrate its substantial improvements over state-of-the-art methods for learning from the instance-dependent partially labeled data.

1 Introduction

The remarkable performance of modern deep neural networks (DNNs) owes much to the large amount of fully supervised training data, and the stringent data requirements can be a barrier to the application of DNNs to certain tasks. Researchers therefore often resort to cheap non-expert labelers, but this invariably leads to low-quality data, a typical example of which is the ambiguity of the labels. *Label ambiguity* is pervasive [Chen *et al.*, 2017] for a simple reason: people have difficulty making exact judgments about tasks that are beyond their expertise, which means that each instance may be labeled with a set of candidate labels (CLs) such that a *fixed and unknown* candidate is the ground truth. Such supervision constantly has negative impacts on the performance of DNNs, since memorization effects [Feldman and Zhang, 2020] make them prone to overfitting all CLs. Thus, *partial label learning* (PLL) [Xu *et al.*, 2019; Lv *et al.*, 2020; Feng *et al.*, 2020; Wang *et al.*, 2020b; Wang *et al.*, 2020a; Wang *et al.*, 2022] which can handle the label ambiguity has drawn a lot of attention in recent years. The goal of PLL is to induce the optimal hypothesis which can generalize well for fully supervised data.

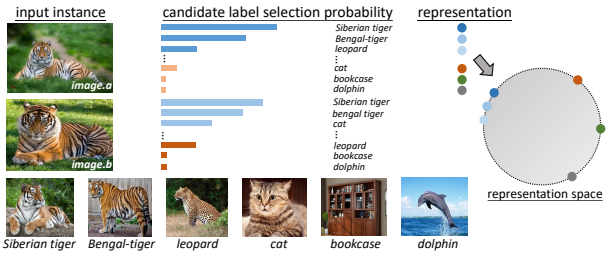

Figure 1: Compared with "dolphin", "bookcase" and etc., "Bengal-tiger" and "leopard" are more likely to be labeled as CLs for the given input instance (e.g., image.a) belonging to "Siberian tiger". Therefore, we reasonably pull the input instance (anchor) and instances predicted to be "Siberian tiger", "Bengal-tiger" or "leopard" (ambiguity-induced positives) closer in the representation space, and meanwhile, push the remaining instances away.

Recent works have presented promising methods on PLL with a common goal to *disambiguate* incorrect label association [Cour *et al.*, 2011], i.e., purifying the CLs heuristically in the training phase to avoid undesired memorization of incorrect CLs. For this purpose, a strand of works [Yu and Zhang, 2017] regard the ground-truth label as a latent variable and identify it by leveraging the *information from feature space*. For example, [Xu *et al.*, 2019] construct a weighted graph over the training instances to characterize the structure of feature space, and then migrate this graph to label space

*Corresponding Author

to identify the ground-truth label. Average-based PLL approaches [Cour *et al.*, 2011; Zhang and Yu, 2015] treat all the candidate labels equally and average the modeling outputs as the prediction. Deep PLL methods of late [Lv *et al.*, 2020; Feng *et al.*, 2020] capitalize on the *inductive bias of the learning model itself*. DNNs typically exhibit an important behavior in that they learn patterns first [Arpit *et al.*, 2017]: this means that labels that are remembered first can be considered as the ground-truth labels. Revisiting all previous PLL works, we note that few of them exploited the label ambiguity, since they explicitly or implicitly assume that incorrect CLs are *noise independent of the instance*.

Unfortunately, this is often the case where human labeling is prone to varying degrees of confusion for instances of varying ambiguity. An overwhelming majority of previous PLL works assume that given the true category (e.g., "Siberian tiger"), each of the other categories (e.g., "Bengal-tiger", "leopard" or "cat") has a fixed probability of being the CL [Zhang and Yu, 2015; Feng *et al.*, 2020]. But apparently, human labeling tends to pick CLs related to both the true label and the instance itself. Let us focus on the two input instances (referred to as image.a and image.b respectively) in the upper left of Figure 1. Despite they belong to the same category ("Siberian tiger"), each of the other categories (e.g., "Bengal-tiger", "leopard" or "cat") has a *unfixed* probability of being the CL which depends on both the true label and the instance itself. This setting is more realistic, i.e., the CLs are *instance-dependent*, such that it is arguable that each CL tends to describe the instance from some sensory channels, such as physics, geometry and semantics. Therefore, it is natural to conclude that the potentially useful information from label ambiguity should also be exploited rather than eliminated directly in more practical instance-dependent PLL.

In this paper, we leverage such additional information acquired from the ambiguity and propose *AmBiguity-induced contrastive LEarning* (ABLE) under the framework of contrastive learning [Khosla *et al.*, 2020; Chen *et al.*, 2020]. Specifically, we construct various positives per anchor by considering each CL of this anchor and selecting a group of samples currently predicted as that class as the *ambiguity-induced positives*. Then each training instance has multiple (the number of its CLs) groups of ambiguity-induced positives for building contrastive losses to pull the anchor and its ambiguity-induced positives closer in the representation space, and push the remaining instances away. Based on them, we learn a *representor* (RP) to minimize the *weighted sum* of these contrastive losses, where each contrastive loss serves as a sub-objective, assigned with an *ambiguity-induced weight*. The weights should be learned, and the larger weights will bias sub-objectives that lead to better representations. To learn the weights and estimate the class of training samples, we train a *classifier* (CS) that minimizes a classic PLL classification loss [Lv *et al.*, 2020]. It deserves a particular mention that there exists a circular dependency between RP and CS: RP requires the ambiguity-induced positives and weights on-the-fly induced by CS, and CS needs the first half of RP as the representation extractor. ABLE proposes a synchronous update strategy of RP and CS to break the circular dependency hopefully. Our **contributions** are summarized as follows:

- We consider a more general setting – instance-dependent PLL and for the first time propose the label ambiguity containing valid information should be exploited.
- We introduce a novel instance-dependent PLL method, which for the first time adapts contrastive learning, and propose an end-to-end training strategy.
- Experiments on benchmark datasets demonstrate substantial improvements over state-of-the-art methods for learning from the instance-dependent partially labeled data.

2 Related Work

2.1 Partial Label Learning

Most recent PLL methods focus on label disambiguation which aims to identify the ground-truth label from the CL set [Lv *et al.*, 2020; Feng *et al.*, 2020; Xu *et al.*, 2021b]. For averaging-based disambiguation [Cour *et al.*, 2011; Zhang and Yu, 2015], all the CLs of each instance are treated equally and the prediction is made by averaging their modeling outputs. For identification-based disambiguation [Yu and Zhang, 2017], the ground-truth label is regarded as a latent variable and identified. For deep-based methods, [Lv *et al.*, 2020] proposes a classifier-consistent risk estimator and a progressive identification algorithm. [Feng *et al.*, 2020] deduces a risk-consistent method and a classifier-consistent method by proposing a statistical model. These methods corrupt data without considering the CLs are always *instance-dependent* in practice. [Xu *et al.*, 2021b] firstly considers the instance-dependent PLL and proposes VALEN which recovers the latent label distribution via inferring the true posterior density of the latent label distribution [Xu *et al.*, 2021a] by Dirichlet density parameterized with an inference model and deduces the evidence lower bound for optimization. However, we note that few of them exploited the label ambiguity. In this paper, we aim to leverage such potentially useful information acquired from the label ambiguity for learning from the partially labeled data.

2.2 Contrastive Learning

Contrastive learning is an approach which is committed to learning an representation space where representations from the same instance are pulled closer and representations from different instances are pushed apart [Khosla *et al.*, 2020]. Positives and negatives are generated for each instance to construct the loss. A plethora of works have explored the effectiveness in unsupervised representation learning [Chen *et al.*, 2020; He *et al.*, 2020]. Lately, [Khosla *et al.*, 2020] proposes Supervised Contrastive Learning which combines explicit supervision to aggregate data from the same class as the positive set. Recently, the success has stimulated a series of works to utilize contrastive learning to weakly supervised learning problems [Li *et al.*, 2021], etc. In this paper, we aim to construct various positives per anchor by considering each CL of this anchor and selecting ambiguity-induced positives currently predicted as that class. Then each training instance has multiple groups of ambiguity-induced positives for building contrastive losses. We further consider that each group of ambiguity-induced positives contributes differently to learning from the partially labeled instances.

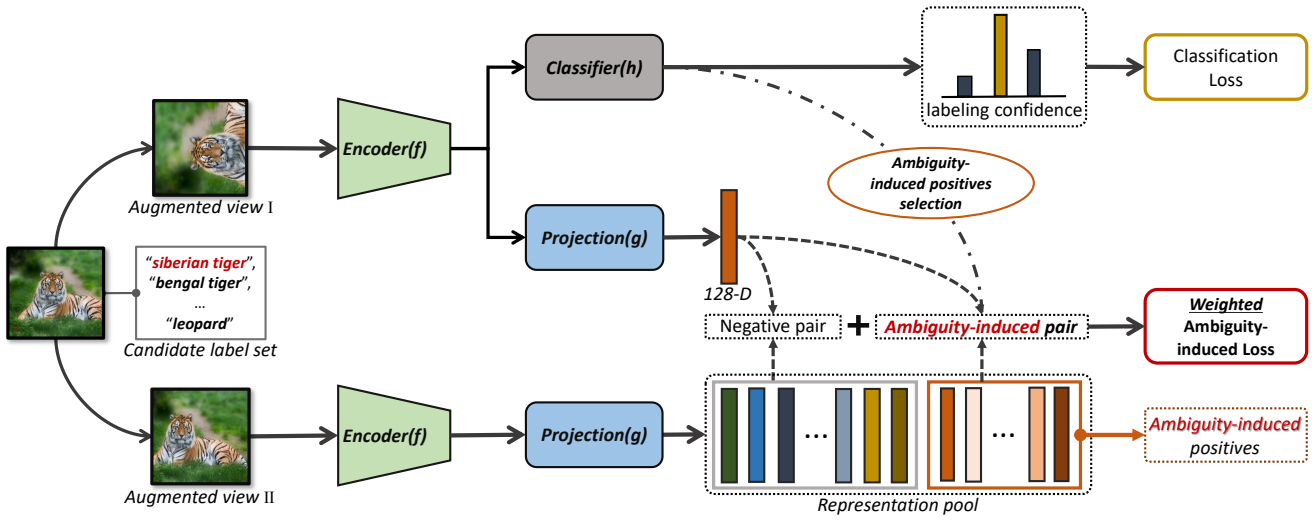

Figure 2: Illustration of ABLE. We construct various positives per anchor by selecting ambiguity-induced positives currently predicted as that class which corresponds to each CL of this anchor. Based on its multiple groups of ambiguity-induced positives, we learn a RP (f and g) to minimize the weighted ambiguity-induced loss, where the weights are obtained by training a CS (h) that minimizes a classification loss. We propose a synchronous update strategy of RP and CS to break a circular dependency existed between RP and CS.

3 Method

In this section, we describe our novel ABLE method in detail. First of all, we give an overview of ABLE (Section 3.1). Then, we describe two key components of ABLE which consists of ambiguity-induced positives selection mechanism (Section 3.2) and ambiguity-induced contrastive learning (Section 3.3). Figure 2 gives an illustration of ABLE.

3.1 Overview of ABLE

First of all, we briefly introduce some necessary notations. Let $\mathcal{X}$ be the input space, $\mathcal{Y} = \{1, 2, ..., c\}$ be the label space with c class labels. Given PLL training set $\mathcal{D} = \{(\boldsymbol{x}_k, S_k)|1 \leq k \leq M\}$ where $\boldsymbol{x}_k$ denotes the training instance and $S_k \subseteq \mathcal{Y}$ denotes the CL set. The key definition of PLL is that the latent ground-truth label $y_k \in \mathcal{Y}$ of an instance $\boldsymbol{x}_k$ is always included in its CL set. And the goal is as the same with supervised classification: learning a classifier h that can make correct predictions on unseen inputs. Here, we get rid of the traditional instance-independent assumption [Feng *et al.*, 2020], i.e., $p(S_k|\boldsymbol{x}_k, y_k) = p(S_k|y_k)$, and consider a more general instance-dependent case.

Given each mini-batch B, $\{(\boldsymbol{x}_k, S_k)|1 \leq k \leq n\}$, we generate two random augmentations [Khosla *et al.*, 2020], i.e., $B_{aug} = \{(aug(\boldsymbol{x}_k), S_k)|1 \leq k \leq n\}$ and $B_{aug'} = \{(aug'(\boldsymbol{x}_k), S_k)|1 \leq k \leq n\}$, where $aug(\cdot)$ and $aug'(\cdot)$ represent two augmentation functions. Therefore, for each mini-batch B, the corresponding batch used for training consists of $2n$ samples, $\{(\widetilde{\boldsymbol{x}}_i, \widetilde{S}_i)|1 \leq i \leq 2n\}$, i.e., $B_{aug} \bigcup B_{aug'}$. For the remainder of this paper, we refer to the n samples as a "batch" and the $2n$ samples as an "augmented batch". Within an augmented batch, let $I = \{1, 2, ..., 2n\}$ be the index set corresponding to the representation pool, $i \in I$ be the index of an arbitrary augmented sample. Following [Chen *et al.*, 2020], first of all, both augmented views are separately fed into the same encoder network $f(\cdot)$ which maps $\widetilde{\boldsymbol{x}}_i$ to an representation $\widetilde{\boldsymbol{v}}_i = f(\widetilde{\boldsymbol{x}}_i) \in \mathbb{R}^{d_e}$, yielding a pair of representations. Note that $\widetilde{\boldsymbol{v}}_i$ is normalized to the unit hypersphere in $\mathbb{R}^{d_e}$. Afterwards we utilize the projection network, $g(\cdot)$, which maps $\widetilde{\boldsymbol{v}}_i$ to a low-dimensional representation $\widetilde{\boldsymbol{z}}_i = g(\widetilde{\boldsymbol{v}}_i) \in \mathbb{R}^{d_p}$. $\widetilde{\boldsymbol{z}}_i$ is also normalized to the unit hypersphere in $\mathbb{R}^{d_p}$. Now we have representations corresponding to the augmented batch. We call $f(\cdot)$ and $g(\cdot)$ as RP. For each augmented training instance $\widetilde{\boldsymbol{x}}_i$, we select multiple groups of ambiguity-induced positives for building contrastive losses. Based on them, we learn the RP to minimize the weighted sum of these contrastive losses which is called weighted ambiguity-induced loss. Meanwhile, CS, i.e., $h(\cdot)$, receives $\widetilde{\boldsymbol{v}}_i$ as input and outputs $\widetilde{\boldsymbol{p}}_i = h(\widetilde{\boldsymbol{v}}_i)$, which is trained by minimizing a PLL classification loss. It is worth noting that there exists a circular dependency between RP and CS: RP requires the ambiguity-induced positives and weights on-the-fly induced by CS, and CS needs $f(\cdot)$ which is the first half of RP as the representation extractor. To break the circular dependence, we jointly train the classification loss and the weighted ambiguity-induced loss which is a synchronous update strategy of RP and CS.

3.2 Ambiguity-Induced Positives Selection

In more practical and realistic instance-dependent PLL, the potentially useful information from label ambiguity should be exploited rather than eliminated directly. We leverage such additional information acquired from the ambiguity by adopting an *ambiguity-induced positives selection* mechanism. Given each PLL training sample $(\widetilde{\boldsymbol{x}}_i, \widetilde{S}_i)$ in the augmented batch, we construct various positives by considering each CL in $\widetilde{S}_i$ and selecting a group of samples currently predicted as that class as the *ambiguity-induced positives*. Note that we impose limits on the class prediction $\widetilde{y}_i = argmax_{j \in \widetilde{S}_i} \widetilde{p}_{ij}$ to be in the CL set $\widetilde{S}_i$, where $\widetilde{p}_{ij}$ denotes the j-th entry of CS output $\widetilde{\boldsymbol{p}}_i$. Then each train-

ing sample has multiple groups of ambiguity-induced positives and the number of groups equals to the number of CLs. We define ambiguity-induced positives set as $A^p(\widetilde{\boldsymbol{x}}_i) = \bigcup_{m\in\widetilde{S}_i} A^p_m(\widetilde{\boldsymbol{x}}_i)$ for the training instance $\widetilde{\boldsymbol{x}}_i$ which is composed of multiple ambiguity-induced positives groups. For each CL in $\widetilde{S}_i$, the corresponding ambiguity-induced positives related to $\widetilde{\boldsymbol{x}}_i$ is defined as follows:

$$A^p_m(\widetilde{\boldsymbol{x}}_i) = \{k' | k' \in N(i), \widetilde{y}_{k'} = m\}, \tag{1}$$

where $N(i) = I\backslash\{i\}$ be the index set of the other samples originating from the same augmented batch. For efficient consideration, we select the ambiguity-induced positives in the current augmented batch. Despite its simplicity, we obtain superior experimental results. We also consider maintaining a queue to store the most current representations and updated predictions [He *et al.*, 2020], which is left for future research.

3.3 Ambiguity-Induced Contrastive Learning

After completing ambiguity-induced positives selection, each training instance has multiple (the number of its CLs) groups of ambiguity-induced positives. Then we construct *ambiguity-induced pairs* which include the training instance and its corresponding ambiguity-induced positives for the following contrastive learning. First of all, we build respective contrastive loss to pull the anchor and its ambiguity-induced positives closer in the representation space, and push the remaining instances away. Given the training representation $\widetilde{\boldsymbol{z}}_i$ and an arbitrary ambiguity-induced positive $\widetilde{\boldsymbol{z}}_p$ which is selected from its ambiguity-induced positives set, we define the contrastive loss as follows:

$$-\log \frac{\exp(\widetilde{\boldsymbol{z}}_i \cdot \widetilde{\boldsymbol{z}}_p/\tau)}{\sum_{l\in N(i)} \exp(\widetilde{\boldsymbol{z}}_i \cdot \widetilde{\boldsymbol{z}}_l/\tau)}, \tag{2}$$

where $N(i) = I\backslash\{i\}$ be the index set of the other representations originating from the same augmented batch, τ is the temperature parameter and $\cdot$ denotes the dot product. We further consider that each group of ambiguity-induced positives contributes differently to learning from the partially labeled instances. In other words, we consider learning a progressively contrastive representation space to facilitate the process of learning from the partially labeled data. We tackle it using the labeling confidences of CLs for progressively putting more weights on more reliable ambiguity-induced pairs. Specifically, we define the *ambiguity-induced weight* in a progressive fashion:

$$w_{ij} = \begin{cases} h_j(f(\widetilde{\boldsymbol{x}}_i)) \Big/ \sum_{k\in\widetilde{S}_i} h_k(f(\widetilde{\boldsymbol{x}}_i)) & \text{if} \quad j \in \widetilde{S}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

where j denotes the indices of CLs. We initialize w_{ij} with uniform weights, i.e., $w_{ij} = 1/|\widetilde{S}_i|$ if $j \in \widetilde{S}_i$, otherwise $w_{ij} = 0$.

Given each training representation $\widetilde{\boldsymbol{z}}_i$ and its ambiguity-induced positives set $A^p(\widetilde{\boldsymbol{x}}_i)$, we define a novel *weighted ambiguity-induced loss* $\mathcal{L}_{wcon}(\widetilde{\boldsymbol{z}}_i, \widetilde{S}_i, A^p, N, \tau)$ as follows:

$$-\sum_{j\in\widetilde{S}_i} \frac{1}{|A^p_j(\widetilde{\boldsymbol{x}}_i)|} \sum_{p\in A^p_j(\widetilde{\boldsymbol{x}}_i)} w_{ij} \cdot \log \frac{\exp(\widetilde{\boldsymbol{z}}_i \cdot \widetilde{\boldsymbol{z}}_p/\tau)}{\sum_{l\in N(i)} \exp(\widetilde{\boldsymbol{z}}_i \cdot \widetilde{\boldsymbol{z}}_l/\tau)}, \tag{4}$$

Algorithm 1 Pseudo code of ABLE (one epoch)

Input: The PLL training dataset $\mathcal{D}$, encoder network $f(\cdot)$, projection network $g(\cdot)$, classifier $h(\cdot)$, uniform ambiguity-induced weights w, constant α.
Output: Parameters of encoder network $f(\cdot)$ and classifier $h(\cdot)$.

for $iter = 1, 2, ..,$ **do**
  Sample a mini-batch $\{(\boldsymbol{x}_k, S_k)\}_{k=1}^n$ from $\mathcal{D}$.
  for $k = 1$ to n **do**
    $\widetilde{\boldsymbol{p}}_k = h(f(aug(\boldsymbol{x}_k)))$
    $\widetilde{\boldsymbol{x}}_{2k-1}, \quad \widetilde{\boldsymbol{x}}_{2k} = aug(\boldsymbol{x}_k), \quad aug'(\boldsymbol{x}_k)$
    $\widetilde{\boldsymbol{z}}_{2k-1}, \quad \widetilde{\boldsymbol{z}}_{2k} = g(f(\widetilde{\boldsymbol{x}}_{2k-1})), \quad g(f(\widetilde{\boldsymbol{x}}_{2k}))$
    $\widetilde{S}_{2k-1} = \widetilde{S}_{2k} = S_k, \quad \widetilde{\boldsymbol{p}}_{2k-1} = \widetilde{\boldsymbol{p}}_{2k} = \widetilde{\boldsymbol{p}}_k$
  end for
  for $k = 1$ to $2n$ **do**
    $\widetilde{y}_k = argmax_{j\in\widetilde{S}_k} \widetilde{p}_{kj}$
    $N(k) = \{1, 2, ..., 2n\}\backslash\{k\}$
    $A^p(\widetilde{\boldsymbol{x}}_k) = \bigcup_{m\in\widetilde{S}_k} \{k' | k' \in N(k), \widetilde{y}_{k'} = m\}$
  end for
  $\mathcal{L}_w = -\frac{1}{2n} \sum_{k=1}^{2n} \sum_{j\in\widetilde{S}_k} \frac{1}{|A^p_j(\widetilde{\boldsymbol{x}}_k)|} \sum_{p\in A^p_j(\widetilde{\boldsymbol{x}}_k)} w_{kj} \cdot \log \frac{\exp(\widetilde{\boldsymbol{z}}_k \cdot \widetilde{\boldsymbol{z}}_p/\tau)}{\sum_{l\in N(k)} \exp(\widetilde{\boldsymbol{z}}_k \cdot \widetilde{\boldsymbol{z}}_l/\tau)}$
  $\mathcal{L}_c = -\frac{1}{n} \sum_{k=1}^{n} \sum_{j=1}^{c} w_{kj} \cdot \log(h_j(f(aug(\boldsymbol{x}_k))))$
  Minimize $\mathcal{L}_{total} = \mathcal{L}_c + \alpha\mathcal{L}_w$.
  Update ambiguity-induced weights w.
end for

where τ is the temperature parameter and $\cdot$ denotes the dot product. We minimize the weighted ambiguity-induced loss which is the weighted sum of these contrastive losses to train the RP, where each contrastive loss serves as a sub-objective to pull the anchor and its ambiguity-induced positives closer in the representation space, and push the remaining instances away. And the larger ambiguity-induced weights will bias sub-objectives that lead to better representations. To learn the weights and estimate the class of training samples, we train the CS that minimizes a classic PLL classification loss [Lv *et al.*, 2020] for each training instance $\boldsymbol{x}_i$:

$$\mathcal{L}_{cls}(\boldsymbol{x}_i, S_i) = -\sum_{j=1}^{c} w_{ij} \cdot \log(h_j(f(aug(\boldsymbol{x}_i)))). \tag{5}$$

It is worth mentioning in particular that there exists a circular dependency between RP and CS: RP requires the ambiguity-induced positives and weights on-the-fly induced by CS, and CS needs the first half of RP as the representation extractor. To break the circular dependency, we propose a synchronous update strategy of RP and CS. Specifically, we jointly train the RP and CS. Therefore, the overall loss $\mathcal{L}_{tot}$ for each training instance $\boldsymbol{x}_i$ is defined as:

$$\mathcal{L}_{tot} = \mathcal{L}_{cls} + \alpha\mathcal{L}_{wcon}, \tag{6}$$

where α is the trade-off parameter for the classification loss and the weighted ambiguity-induced loss. The pseudo-code of our ABLE is shown in Algorithm 1.

	MNIST	Fashion-MNIST	Kuzushiji-MNIST	CIFAR-10
Supervised	99.37±0.05%	95.26±0.12%	98.84±0.06%	97.19±0.13%
ABLE	**99.32±0.04%**	**92.34±0.09%**	**98.50±0.14%**	**92.30±0.24%**
VALEN	99.21±0.02%	91.45±0.18%•	96.82±0.14%•	90.16±0.52%•
PRODEN	99.18±0.05%	91.42±0.12%•	96.71±0.15%•	89.53±0.28%•
RC	98.88±0.05%•	91.03±0.13%•	95.34±0.17%•	89.58±0.26%•
CC	98.72±0.06%•	90.87±0.09%•	93.86±0.45%•	89.21±0.64%•
D2CNN	95.96±0.12%•	87.92±0.22%•	94.25±0.21%•	84.28±0.24%•

Table 1: Classification accuracy (mean±std) on benchmark datasets corrupted by the instance-dependent generating procedure.

4 Experiments

4.1 Setup

Datasets. We adopt four widely used benchmark datasets including MNIST [LeCun *et al.*, 1998], Fashion-MNIST [Xiao *et al.*, 2017], Kuzushiji-MNIST [Clanuwat *et al.*, 2018], CIFAR-10[1]. To generate the instance-dependent candidate labels, we set the flipping probability of each incorrect label corresponding to each instance by using the confidence prediction of a neural network trained with clean labels [Xu *et al.*, 2021b]. We run five trials and record the mean accuracy as well as standard deviation for all comparing methods.

Baselines. We compare the performance of ABLE against five methods, each configured with parameters suggested in respective literature: 1) VALEN [Xu *et al.*, 2021b]:An instance-dependent PLL method which recovers the label distribution and deduces the evidence lower bound for optimization; 2) PRODEN [Lv *et al.*, 2020]: A progressive identification PLL method which approximately minimizes a risk estimator and identifies the true labels in a seamless manner; 3) RC [Feng *et al.*, 2020]: A risk-consistent PLL method which employs the importance reweighting strategy to converge the true risk minimizer; 4) CC [Feng *et al.*, 2020]: A classifier-consistent PLL method which uses a transition matrix to form an empirical risk estimator; 5) D2CNN [Yao *et al.*, 2020]: A deep PLL method which designs an entropy-based regularizer to maximize the margin between the potentially correct label and the unlikely ones.

Implementation details. For the encoder network $f(\cdot)$, we experiment with ResNet-18 [He *et al.*, 2016]. The normalized activations of the final pooling layer (d_e = 512) are used as the representation. For the projection network $g(\cdot)$, we instantiate $g(\cdot)$ as a multi-layer perceptron with a single hidden layer of size 512 (as well as ReLU activation) and output representation of size 128. We also normalize the low-dimensional representation to lie on the unit hypersphere. For the classifier $h(\cdot)$, we instantiate $h(\cdot)$ as a single linear layer. We use two data augmentation modules following [Khosla *et al.*, 2020; Wang *et al.*, 2022]. The trade-off parameter is set as $\alpha = 1.0$. The temperature parameter is set as $\tau = 0.1$. The mini-batch size, the number of training epochs, the initial learning rate and the weight decay are set to 64, 500, 0.01 and 1e-3, respectively. We adopt cosine learning rate scheduling. The optimizer is stochastic gradient descent (SGD) with momentum 0.9. We implement ABLE with PyTorch. Source code is available at https://github.com/AlphaXia/ABLE. We also want to use ABLE on MindSpore[2], which is a new deep learning framework. These problems are left for future work.

[1]https://www.cs.toronto.edu/~kriz/cifar.html

	Fashion-MNIST	CIFAR-10
ABLE	**92.34±0.09%**	**92.30±0.24%**
*Version*1	91.52±0.12%•	90.48±0.26%•
*Version*2	91.06±0.25%•	89.82±0.24%•
*Version*3	91.42±0.12%•	89.53±0.28%•

Table 2: Effect of exploiting label ambiguity and utilizing contrastive learning in ABLE on Fashion-MNIST and CIFAR-10.

4.2 Experimental Results

ABLE achieves SOTA results. We report the classification accuracy of each method in Table 1. • indicates whether ABLE is statistically superior to the comparing method on each dataset (pairwise t-test at 0.05 significance level). In addition, the best results are highlighted in bold. As shown in Table 1, we can observe that ABLE always outperforms all the compared methods by a significant margin on all datasets, which validates the effectiveness of our ABLE. Finally, we observe that ABLE achieves results that are comparable to the fully supervised learning model on some datasets, showing that exploiting label ambiguity facilitates the process of learning from the instance-dependent partially labeled data.

Effect of exploiting label ambiguity and utilizing contrastive learning. We ablate the contributions of two key components of ABLE: ambiguity-induced positives selection mechanism and ambiguity-induced contrastive learning. Specifically, we compare ABLE with three weakened versions: (1)*Version*1: ABLE *w/o weighted ambiguity-induced positives* which removes the ambiguity-induced weights. (2)*Version*2: ABLE *w/o ambiguity-induced positives* which removes the ambiguity-induced positives. (3)*Version*3: ABLE *w/o utilizing contrastive learning* which removes the contrastive learning part. As shown in Table 2, we can observe that ABLE obtains more superior results than *Version*1 (e.g., +2% on CIFAR-10) and *Version*2 (e.g., +3%

[2]https://www.mindspore.cn/

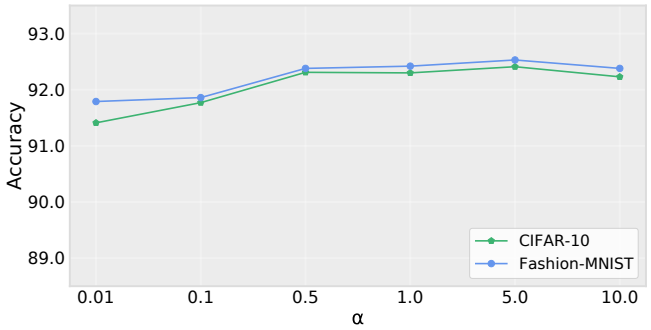

Figure 3: Performance of ABLE with varying α values on Fashion-MNIST and CIFAR-10.

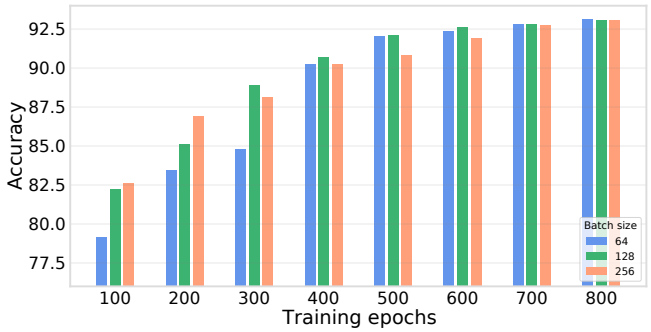

Figure 4: Performance of ABLE with different mini-batch sizes and training epochs on CIFAR-10.

on CIFAR-10), which verifies the effectiveness of two key components of ABLE.

Effect of trade-off factor α. As illustrated in Figure 3, we also report the performance of ABLE with varying α values that trade-off the classification loss and our weighted ambiguity-induced loss. In our setting, α is selected from $\{0.01, 0.1, 0.5, 1.0, 5.0, 10.0\}$. We can find that the result is stable when performing on Fashion-MNIST and CIFAR-10. Similar empirical results were also found on other benchmark datasets. We also want to use dynamic trade-off factor to balance the classification loss and our weighted ambiguity-induced loss, which is left for future work.

ABLE benefits from longer training. We report the impact of mini-batch size when models are trained for different numbers of training epochs in Figure 4. We observe that training longer progressively puts more ambiguity-induced weights on more reliable ambiguity-induced pairs, which improves the results. We also find that larger mini-batch size has a significant advantage over the smaller one when the number of training epochs is small (e.g. 200 epochs). With more training epochs, the gaps between different mini-batch sizes decrease or disappear.

Bigger encoder networks promote ABLE. We report that ABLE benefits from bigger encoder networks as illustrated in Table 3 while similar findings hold for supervised learning. We consider that means the power of contrastive learning can be released by using bigger encoder networks.

Encoder	ResNet18	ResNet34	ResNet50
Supervised	97.19±0.13%	97.72±0.14%	98.24±0.10%
ABLE	92.30±0.24%	92.59±0.20%	92.94±0.24%

Table 3: Performance of ABLE with different size of encoder network on CIFAR-10.

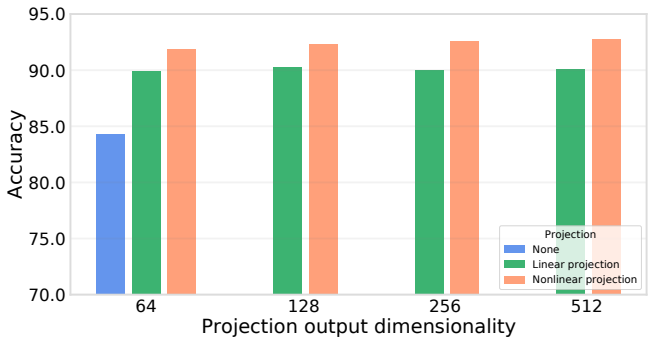

Figure 5: Performance of ABLE with different projection networks and different projection output dimensionalities on CIFAR-10.

Nonlinear projection networks improve ABLE. We then study the necessity of designing a suitable projection network for ABLE, i.e. $g(\cdot)$. Figure 5 shows results using different settings for the projection network: (1) None, i.e., no projection network. (2) Linear projection network with one linear layer. (3) Nonlinear projection network with one additional hidden layer and ReLU activation. We find that a nonlinear projection network achieves better results than a linear projection network (e.g., +2% on CIFAR-10), and also much better than no projection network (e.g., +7% on CIFAR-10). We consider that more knowledge which benefits training can be obtained by utilizing the nonlinear projection network.

5 Conclusion

In this paper, we considered a more practical case of PLL than those have been well-studied. We rethought label ambiguity in instance-dependent PLL and pointed out that it contains valid information which may help in representation learning and deep classifier training. To leverage such useful information, we proposed a novel method named ABLE that extended the contrastive loss by selecting ambiguity-induced positives, and updated the representor and classifier synchronously within a coherent framework. To the best of our knowledge, this is the first time to apply contrastive learning to instance-dependent PLL. Experiments on benchmark datasets validated the effectiveness of our method.

Acknowledgments

This research was supported by the National Key Research & Development Plan of China (No. 2018AAA0100104, No. 2018AAA0100100), the National Science Foundation of China (62125602, 62076063), China Postdoctoral Science Foundation (2021M700023), Jiangsu Province Science Foundation for Youths (BK20210220).

References

[Arpit *et al.*, 2017] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. C. Courville, Y. Bengio, and S. Lacoste-Julien. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242, Portland, OR, USA, August 2017. ACM.

[Chen *et al.*, 2017] Ching-Hui Chen, Vishal M Patel, and Rama Chellappa. Learning from ambiguously labeled face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(7):1653–1667, 2017.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607, Virtual Event, July 2020. ACM.

[Clanuwat *et al.*, 2018] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *CoRR*, 2018.

[Cour *et al.*, 2011] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12(5):1501–1536, 2011.

[Feldman and Zhang, 2020] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Advances in Neural Information Processing Systems*, pages 2881–2891, Virtual Event, December 2020. MIT Press.

[Feng *et al.*, 2020] L. Feng, J. Lv, B. Han, M. Xu, G. Niu, X. Geng, B. An, and M. Sugiyama. Provably consistent partial-label learning. In *Advances in Neural Information Processing Systems*, pages 10948–10960, Virtual Event, December 2020. MIT Press.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE.

[He *et al.*, 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9726–9735, Seattle, WA, USA, June 2020. IEEE.

[Khosla *et al.*, 2020] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, pages 18661–18673, Virtual Event, December 2020. MIT Press.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Li *et al.*, 2021] Junnan Li, Caiming Xiong, and Steven C. H. Hoi. Mopro: Webly supervised learning with momentum prototypes. In *International Conference on Learning Representations*, Virtual Event, May 2021. OpenReview.net.

[Lv *et al.*, 2020] J. Lv, M. Xu, L. Feng, G. Niu, X. Geng, and M. Sugiyama. Progressive identification of true labels for partial-label learning. In *International Conference on Machine Learning*, pages 6500–6510, Virtual Event, July 2020. ACM.

[Wang *et al.*, 2020a] Haobo Wang, Weiwei Liu, Yang Zhao, Tianlei Hu, Ke Chen, and Gang Chen. Learning from multi-dimensional partial labels. In *International Joint Conference on Artificial Intelligence*, pages 2943–2949, Virtual Event, Japan, January 2020. ijcai.org.

[Wang *et al.*, 2020b] Haobo Wang, Yuzhou Qiang, Chen Chen, Weiwei Liu, Tianlei Hu, Zhao Li, and Gang Chen. Online partial label learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 455–470, Ghent, Belgium, September 2020. Springer.

[Wang *et al.*, 2022] Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. PiCO: Contrastive label disambiguation for partial label learning. In *International Conference on Learning Representations*, 2022.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, 2017.

[Xu *et al.*, 2019] N. Xu, J. Lv, and X. Geng. Partial label learning via label enhancement. In *Association for the Advance of Artificial Intelligence*, pages 5557–5564, Honolulu, Hawaii, January 2019. AAAI Press.

[Xu *et al.*, 2021a] Ning Xu, Yun-Peng Liu, and Xin Geng. Label enhancement for label distribution learning. *IEEE Trans. Knowl. Data Eng.*, 33(4):1632–1643, 2021.

[Xu *et al.*, 2021b] Ning Xu, Congyu Qiao, Xin Geng, and Min-Ling Zhang. Instance-dependent partial label learning. In *Advances in Neural Information Processing Systems*, Virtual Event, December 2021. MIT Press.

[Yao *et al.*, 2020] Y. Yao, C. Gong, J. Deng, X. Chen, J. Wu, and J. Yang. Deep discriminative CNN with temporal ensembling for ambiguously-labeled image classification. In *Association for the Advance of Artificial Intelligence*, pages 12669–12676, New York, NY, USA, February 2020. AAAI Press.

[Yu and Zhang, 2017] F. Yu and M. Zhang. Maximum margin partial label learning. *Machine Learning*, 106(4):573–593, 2017.

[Zhang and Yu, 2015] M. Zhang and F. Yu. Solving the partial label learning problem: An instance-based approach. In *International Joint Conference on Artificial Intelligence*, pages 4048–4054, Buenos Aires, Argentina, July 2015. AAAI Press.