2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Class Prototype-guided Disambiguation in Partially Labeled Learning

Yu He[1], YaLing Ge[1] and Jun Zhou[1](✉)

[1] College of Computer and Information Science, Southwest University, China
zhouj@swu.edu.cn

Abstract. Partial Label Learning (PLL) is a prominent research direction in weak supervision, in which each instance is associated with a set of ambiguous candidate labels. Recent PLL methods primarily focus on uncovering the latent true label using label ambiguity information. However, the candidate label set contains only one true label. We directly utilize the entire candidate set will introduce label noise and hinder performance improvement of model training. To address this issue, we propose a guided model learning method called Class Prototype-induced Weighted Contrastive Partial Label Learning method (PIWCL) to effectively reduce the impact of label noise. Specifically, PIWCL consists of the Class Prototype-guided Module (CPGM) and the Weighted Contrastive Learning Module (WCLM). WCLM employs a novel weighting scheme to learn more compact and discriminative representations, mitigating the confusion caused by ambiguous class samples while capturing useful latent information. Meanwhile, CPGM guides the classifier's learning process, further improving its ability to distinguish between positive and negative samples and facilitating the training of WCLM. Experimental results show that, compared to existing PLL methods, PIWCL achieves significant improvements in effectiveness.

Keywords: Partial Label Learning, Class Prototype, Contrastive Learning, Weakly Supervised Learning

1 Introduction

The excellent performance of deep neural networks heavily relies on a large amount of accurately labeled data. To reduce the cost of data labeling, non-experts annotators are often chosen to label the data, but this may lead to label ambiguity[1, 2](see Fig. 1). To address this issue, Partial Label Learning (PLL) has emerged as a promising solution [2-11]. Its goal is to train models using candidate labels that contain ambiguous labels, helping the model infer the true label for each sample and resolve label uncertainty. PLL is also widely applied in fields such as malignant prediction of lung nodules[12], image classification[6], and object detection[8].

To address the issue of label ambiguity in training instances in PLL, researchers have explored methods to identify true labels from candidate labels [13-15]. Early works optimized models using maximum likelihood models and the expectation maximization algorithm [16]. Current PLL methods mainly employ techniques such as contrastive

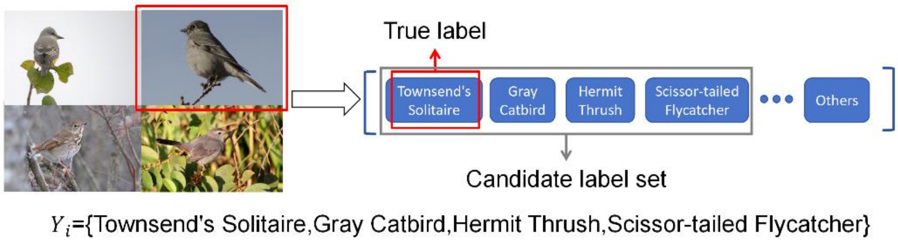

Fig. 1. In Partial Label Learning, we typically assign a candidate label set to images that are difficult to distinguish. For example, the bird image highlighted with a red box in the figure is assigned a candidate label set containing four labels, of which only one is the true label for the image.

learning, label disambiguation and feature representation optimization [2, 17]. Some approaches suggest treating all candidate labels equally and using the model's output average for prediction [18, 19]. Additionally, SAUTE selects features related to label information by maximizing mutual information [20]. However, these methods often rely on specific assumptions, such as the independence of candidate labels or an over-reliance on label information. ABLE introduced an instance-based PLL approach, suggesting that each candidate label in the candidate labels can be used as a basis for contrastive learning to extract additional information from ambiguous labels [9]. Some researchers have proposed a novel "mutual supervision" paradigm by introducing a partner classifier and designing a collaboration term to mutually supervise with the base classifier, aiming to identify and correct mislabeled samples in PLL [4].

However, noise is often introduced during the process of utilizing label ambiguity information. Specifically, since the model is not effective in distinguishing samples that are difficult to distinguish, if the label information containing ambiguity continues to be used, it is easy to aggravate the uncertainty of the model to select the wrong label and introduce noise. As training progresses, these noises gradually accumulate and eventually affect the performance of the model. Therefore, it is necessary to pay special attention to the potential interference of these ambiguous labels on the learning of real labels. We compared the performance of several methods (see Fig. 2) and the experimental results showed that the performance improvement of most methods significantly slowed down or even stagnated in the late training period due to the influence of noise.

In order to solve the above problems, we are inspired by class prototype and contrast learning and establish a prototype-sample bidirectional cooperative optimization mechanism. A new Class Prototype-induced Weighted Contrastive Partial Label Learning (PIWCL) method is proposed. Specifically, we use a classifier-based feature selection mechanism to divide positive pairs into positive and fuzzy sets and apply weights to them. This module is called the Weighted Contrastive Learning Module (WCLM). The weighted design not only captures potentially ambiguous information while clarifying the boundaries between classes, but also generates better class prototypes. In addition, the proposed Class Prototype-guided Module (CPGM) calculates the feature similarity between samples and the constructed class prototypes,which is used to construct the class prototype-guided loss. The class prototype guides the classifier to train the model

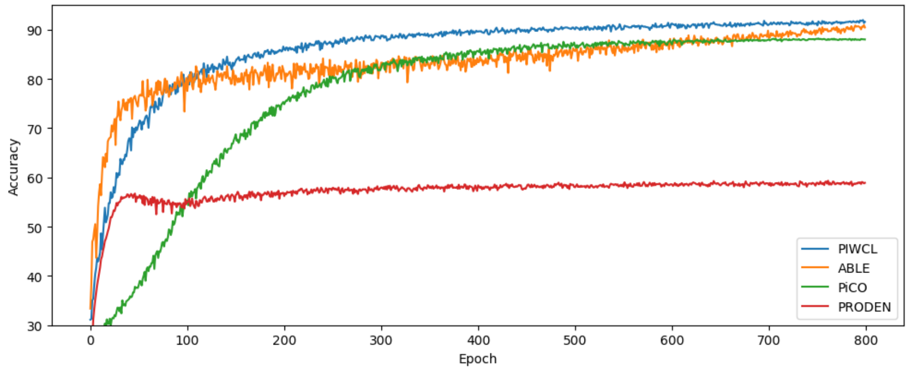

Fig. 2. In PLL, we typically assign a candidate label set to images that are difficult to distinguish. For example, the bird image highlighted with a red box in the figure is assigned a candidate label set containing four labels, of which only one is the true label for the image.

and enhances its ability to distinguish between positive class and fuzzy class, which facilitates the training of PIWCL. These two modules complement each other and work together to improve the performance of the model. The detailed method is described in Section 3. Our contributions are summarized as follows:

1. We reconsidered the ambiguity-induced contrastive learning module in ABLE [9] and proposed that while ambiguous labels have value for utilization, it is also important to be mindful of their boundaries with the true labels.
2. To solve the problem of model noise accumulation in partial label learning, we propose a dynamic class prototype guidance method named Class Prototype-induced Weighted Contrastive Partial Label Learning (PIWCL). This method uses Weighted Contrastive Learning Module (WCLM) to perceive ambiguous information and assign different attention to different categories to generate better representations, thus promoting the construction of high-quality class prototypes. Then the Class Prototype-guided Module (CPGM) guides the classifier to learn. Furthermore, the semantic ambiguity propagation in partial label learning is suppressed, and the discriminant ability of the model is enhanced.
3. Experiments on multiple datasets show that PIWCL outperforms most of the current advanced PLL methods.

2 Related Works

2.1 Partial Label Learning (PLL)

PLL originates from weakly supervised learning and multilabel learning. It can be divided into three aspects: disambiguation-based strategies, transformation-based strategies and theorydriven strategies [1, 4-17, 21]. Recently, transformation-based strategies have improved label disambiguation and the overall performance of PLL by enhancing

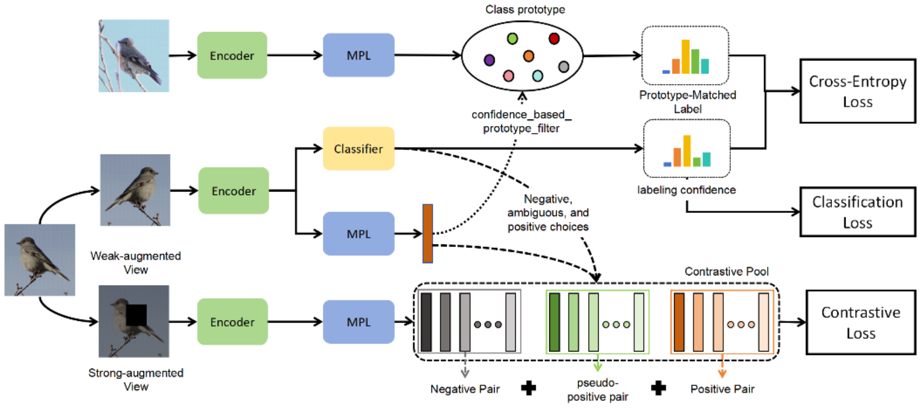

Fig. 3. Illustration of PIWCL. In Class Prototype-guided Module, an appropriate feature is selected for updating the class prototypes through a confidence-based prototype filter. Then, the similarity between the instance and prototype features is used as the basis for pseudo-target updates, thereby constructing a class prototype-guided loss to guide the classifier's learning. Additionally, classifier-based feature selection mechanism is used to select the positive example set, fuzzy example set, and negative example set, which are then used to build a weighted contrastive loss. Finally, the model improves performance and accuracy by minimizing the two losses above, jointly optimizing the classifier and the projector.

feature space consistency [4, 5]. Another group of researchers has pointed out the potential of theoretical approaches [21, 22]. Disambiguation-based strategies mainly focus on eliminating incorrect labels in candidate labels. IMVPML uses lowrank and sparse decomposition to remove noisy labels, combined with graph Laplacian regularization and orthogonality constraints to constrain the true labels [17]. MILe eliminates label ambiguity by propagating binary predictions between teacher and student networks within an iterative learning framework [22]. ABLE minimizes the contrastive and classification losses to avoid over-reliance on label information [9]. However, we notice that few approaches address the distinction between ambiguous and true labels while utilizing ambiguous label information.

2.2 Class prototype

The class prototype method is a prototype-based learning approach that aims to achieve classification or clustering tasks by representing each class with a "typical" or "representative" sample (i.e., class prototype)[23, 24]. Some researchers have proposed to use multi-class label information and a pyramid feature fusion module during the training process to encourage the network to generate compact features and robust prototypes for each semantic class [25]. Others have proposed a non-parametric method based on unlearnable prototypes, which overcomes the limitations of parametric segmentation mechanisms [26]. Inspired by the above, we introduce the class prototype method into PLL, using prototypes to guide the classifier's training.

2.3 Contrastive learning

Contrastive learning is an unsupervised learning method that helps the model learn data features by pulling similar samples closer and pushing dissimilar samples apart [27]. RegionPLC achieved high-quality 3D learning without human annotations using a 3D perception strategy and contrastive learning [28]. Other researchers use label information to cluster similar samples, and separate dissimilar samples through supervised contrastive loss to optimize the representation in the embedding space [29]. Contrastive learning has been widely applied in fields such as image, text and audio. In this paper, we utilize contrastive learning to capture potential ambiguous information in the candidate label set.

3 Methods

In this section, we will provide a detailed introduction to the PIWCL method. First, in Section 3.1, we will introduce some basic methods and notations. This is followed by an explanation of our PIWCL method. Sections 3.2 and 3.3 give a detailed description of the Class Prototype Guidance Module and the Weighted Contrastive Learning Module. A brief description of PIWCL is also provided (see Fig. 3).

3.1 Preliminaries

We assume that χ is the input space, and $y=\{1,2,\cdots,m\}$ is the label space containing m class labels. Given a training dataset $D=\{(x_i,Y_i)\,|\,1\le i\le N\}$ for PLL, where x_i represents the i-th training sample and $Y_i\subseteq y$ represents the candidate label set corresponding to the i-th sample. PLL aims to identify the true label of sample x_i from the candidate label set Y, which contains multiple possible labels and uses these candidate labels to train the model. Through iterative optimization, the model gradually infers the most likely true label. Here, we consider the general instance-dependent case, which is consistent with ABLE [9]. Given a batch of samples $B=\{(x_k,Y_k)\,|\,1\le k\le b\}$, we adopt the universal method to randomly generate a weak augmented view $(Aug_w(x_i),Y_i)$ and $Aug_s(\bullet)$ a strong augmented view $(Aug_s(x_i),Y_i)$ [29]. Here, $Aug_w(\bullet)$ and represent the weak augmentation function and strong augmentation function, respectively. Therefore, the two sets of augmented samples for the batch are defined as $B_w=\{(Aug_w(x_i),Y_i)\,|\,1\le k\le b\}$ and $B_s=\{(Aug_s(x_i),Y_i)\,|\,1\le k\le b\}$. Each sample has a corresponding index, and their indices in the contrastive pool are given by $I=\{1,2,\cdots,2b\}$. During training, these two sets of samples $B_w\cup B_s$, with a total of $2b$ samples, are used as the training samples for the batch. Based on SimCLR [30], these two augmented views of the sample $x_k{}^w$ and $x_k{}^s$ are input into a shared-weight encoder network $f(\bullet)$, resulting in a pair of representations $v_k{}^w=f(Aug_w(x_k^w))$ and $v_k{}^s=f(Aug_s(x_k{}^s))$. Subsequently, these representations v_k^w and v_k^s are mapped through a projection network $g(\bullet)$ to $z_k^w=g(v_k^w)\in\mathbb{R}^{d_p}$ and

$z_k^s = g(v_k^s) \in \mathbb{R}^{d_p}$, and are further normalized onto the unit sphere in $\mathbb{R}^{d_p}$. To ensure the reliability of the prototypes, we use confidence-based prototype filter to select high-confidence features for prototype updating. The classifier's performance is improved by minimizing the crossentropy loss between the prototype similarity and the disambiguated labels. Meanwhile, the classifier $h(\bullet)$ receives v_k^w as input and outputs $y_k = h(v_k^w)$, training the model by minimizing the classification loss for PLL. Additionally, we introduce WCLM, which divides the received sample features z_k into positive, ambiguous and negative classes using confidence-based prototype filter. A weighted contrastive loss is then constructed to encourage the model to focus on learning positive samples and extracting information from ambiguous samples. During training, CPGM guides the classifier to better distinguish between real and ambiguous classes, while WCLM generates better representations and class prototypes to guide the classifier's learning.

3.2 Class Prototype Guidance Module(CPGM)

As analyzed earlier, when we utilize information with label ambiguity, it is easy to confuse the boundaries between categories. To address this issue, we are inspired by PiCO [31]. We propose using CPGM to guide the classifier's training, enabling it to better distinguish between real and ambiguous classes.

Confidence-based Prototype Filter. Considering that the model may be unstable in the early stages of training and may assign incorrect pseudo-labels to some samples, we propose a Confidence-based Prototype Filter to filter reliable samples for effective prototype updates. Specifically, given each class prototype μ_k ,where $k \in y = \{1, 2, \cdots, m\}$. We update the prototype using high-confidence samples predicted to belong to the same class, with their features z_k^w . Moreover, to avoid the high computational cost of recalculating class prototypes μ_k at each iteration, we adopt a moving average method to update the class prototypes μ_k :

$$\mu_k = \begin{cases} \varphi\mu_k + (1-\varphi)\, z_k^w & \text{if} \quad y_k > \varepsilon \\ \mu_k & otherwise \end{cases} \tag{1}$$

Where φ is the update coefficient, which decreases as the epoch increases. ε is the set confidence threshold, and $k = \arg\max_{c \in Y} h^c(Aug_w(x))$.

Class Prototype Guidance Loss. As analyzed earlier, we need to use the prototypes μ_k to guide the classifier during training. For each given sample (x_i, Y_i) , we compute its highest similarity with the class prototype μ_k as the basis for updating, and use a moving average approach to update the pseudo-targets:

$$s_{ij} = \gamma s_{ij} + (1-\gamma) c_j , c_j = \begin{cases} 1 & \text{if} \quad \mathrm{j} = \arg\max_{j \in Y} z_i^\top \mu_j \\ 0 & \text{else} \end{cases} \tag{2}$$

Where γ is the smoothing factor, and the pseudo-target s_{ij} is initialized using a uniform distribution, $s_{ij} = \frac{1}{|Y_i|}\mathbb{I}(j \in Y_i)$, which provides a good initialization for the classifier $h(\bullet)$. Now, using the pseudo-target s_{ij} as the learning objective, we employ cross-entropy loss as the class prototype guidance loss to train the classifier $h(\bullet)$:

$$L_{pgl} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m} -s_{ij}\log\left(\frac{\exp\left(y_{ij}\right)}{\sum_{k}^{m}\exp\left(y_{ik}\right)}\right) \tag{3}$$

Where y_{ij} is the logits value for the j-th class of the i-th sample, N is the total number of samples and m is the number of classes.

3.3 Weighted Contrastive Learning Module(WCLM)

Based on the previous section, the update of class prototypes relies on the representations of the samples. Contrastive learning can play the role of class clustering in the embedding space to generate a good representation for the samples, thus facilitating the generation of class prototypes [31]. Here, we propose a Weighted Contrastive Learning Module. WCLM employs a classifier-based feature selection mechanism to construct contrastive learning by selecting the positive set $P_{positive}$, pseudo-positive set P_{pseudo} and negative set $P_{negative}$. The positive set $P(x_i)$ of sample x_i is defined as follows:

$$P\left(x_i\right) = P_{positive} \cup P_{pseudo} \tag{4}$$

Here, $N\left(i\right) = \left\{1, 2, \cdots, N\right\} \setminus \left\{i\right\}$ represents the index set of all samples in the augmented batch except for sample i, and $\hat{y}_i$ is the predicted label of sample i. Then, $P_{positive} = \{k' | k' \in N(i), \hat{y}_{k'} = \hat{y}_i\}$ and $P_{pseudo} = \{k' | k' \in N(i), \hat{y}_{k'} \neq \hat{y}_i, \hat{y}_{k'} \in Y_i\}$. After obtaining the positive set $P(x_i)$, we can construct the weighted contrastive loss. We create a contrastive loss for each sample, aiming to bring the distance to the instances in the positive set $P(x_i)$ closer and push away the remaining instances. It is worth noting that, due to the presence of pseudo-positive examples selected through the pseudo-positive set $P(x_i)$, we need to be cautious about the "pulling force" of the positive example set. Clearly, the "pulling force" should be strongest for instances of the same class, while it should be weaker for other instances. Based on this, we construct a weight ω_{ij} for the contrastive loss:

$$\omega_{ij} = \sigma_{ij}\left(\phi\mathbb{I}(i \in P_{pseudo}) + \mathbb{I}(i \in P_{positive})\right) \tag{5}$$

Similarly, we use a uniform distribution $\omega_{ij} = \frac{1}{|Y_i|}\mathbb{I}(j \in Y_i)$ to ensure a well-initialized weights. ϕ is an adjustable factor for the "pulling force" of pseudo-positive instances, with a default value of 0.5. The formula for σ_{ij} is as follows:

$$\sigma_{ij} = \begin{cases} \frac{y_{ij}}{\sum_{k \in Y_i} y_{ik}} & \text{if} \quad j \in Y_i \\ 0 & otherwise \end{cases} \tag{6}$$

Now, we can construct the weighted contrastive loss using the positive set and the weights:

$$L_{wcl} = \sum_{j \in Y_i} \frac{1}{|\mathcal{P}(x_i)|} \sum_{k \in \mathcal{P}(x_i)} \omega_{ij} \log \frac{\exp(z_i \cdot z_k / \tau)}{\sum_{l \in N(i)} \exp(z_i \cdot z_l / \tau)} \tag{7}$$

Where $\tau \geq 0$ is the temperature coefficient. We train the model by minimizing the weighted contrastive loss for each sample to generate better representations. By learning the weighted contrastive loss, the samples in the positive set $P(x_i)$ are pulled closer together, while the samples in the negative set are pushed further apart. Additionally, under the weighted influence, the pseudo-positive samples in the positive set $P(x_i)$ not only allow the model to learn their potential information but also help distinguish them from the positive samples. Furthermore, the classification loss for each sample is as follows:

$$L_{cls} = \sum_{k \in P(x_i)} \sigma_{ij} \log \left(\frac{\exp(y_{ij})}{\sum_k^m \exp(y_{ik})} \right) \tag{8}$$

Throughout the process, WCLM and CPGM work in tandem. WCLM uses classifier-based feature selection mechanism to divide samples into positives, pseudo-positives and negatives. Then, WLCM employs a reweighting strategy to construct a weighted contrastive loss, enabling the model to focus on learning from positives while also exploring the potential information in pseudo-positives. Meanwhile, CPGM assists the classifier in accurately distinguishing positives from pseudo-positives, providing WCLM with more precise samples. This drives WCLM to generate higher-quality class representations, optimizing decision boundaries and class prototypes μ_k. Ultimately, the overall loss for each sample is as follows:

$$L = L_{cls} + \alpha L_{pgl} + \beta L_{wcl} \tag{9}$$

Here, α and β are the weight parameters for the class prototype-guided loss and weighted contrastive loss, respectively, with a default value of 1.0.

4 EXPERIMENTATS

4.1 Setup

Dataset. We used four benchmark datasets widely applied in computer vision tasks: MNIST [32], FashionMNIST [33], Kuzushiji-MNIST [34], and CIFAR-10 [35]. These datasets cover different image classification tasks, providing high representativeness and comparability. Additionally, for the candidate labels required by PLL, we adopted the same method as in ABLE [9]. The prediction confidence of the trained neural network is used as the probability of flipping the incorrect label for an instance. We conducted multiple experiments and recorded the best result.

Baseline. We compared PIWCL with six advanced PLL methods: (1) ABLE [9], which introduces ambiguity information into contrastive learning and enables the collaborative training of the encoder and classifier. (2) PiCO [31], which addresses the challenges of representation learning and label disambiguation through contrastive learning and a class prototype-based label disambiguation algorithm. (3)RECORDS

[36] proposes a dynamic rebalancing strategy that, through biased output parameter decomposition, performs harmless adjustment to the label disambiguation process without assuming any prior knowledge of class distribution. (4) PRODEN [11] proposes a new classification risk estimator and a progressive identification algorithm that is compatible with stochastic optimization. (5) LWS [10] introduces a new loss function that balances the relationship between partial and non-partial labels. (6) VALEN [13] recovers the label distribution through a label enhancement process, iteratively training the prediction model. These methods were configured according to the parameters referenced in their respective papers.

Implementation. For all datasets, we employ commonly used data augmentation methods [9]. We chose ResNet-18 as the encoder network, which outputs features with 512 dimensions [37]. The projector is a multilayer perceptron with one hidden layer (using ReLU activation function) and outputs a 128-dimensional embedding vector for contrastive learning. Additionally, the classifier is instantiated as a single linear layer. During training, we set the batch size to 64, the number of epochs to 800 and the initial learning rate to 0.001 to ensure model convergence stability. The optimizer uses stochastic gradient descent. To evaluate the model's performance, we conducted three independent experiments and selected the best value as the final result.

Table 1. The seven methods are compared on four benchmark datasets: MNIST, FashionMNIST, Kuzushiji-MNIST, and CIFAR-10, where the best performance is indicated in bold.

	MNIST	FashionMNIST	Kuzushiji-MNIST	CIFAR-10
PIWCL(ours)	**99.41%**	**92.34%**	**98.66%**	**91.94%**
ABLE	99.32%	92.14%	98.38%	90.89%
PiCO	99.28%	88.98%	91.99%	88.54%
RECORDS-LTPLL	99.36%	90.93%	98.74%	78.44%
PRODEN	97.61%	88.67%	88.93%	59.52%
LWS	98.86%	90.45%	87.60%	87.14%
VALEN	97.89%	89.14%	89.10%	84.83%

Table 2. Accuracy of PIWCL under Different Confidence Thresholds ε .

500epoch	$\varepsilon = 0.6$	$\varepsilon = 0.7$	$\varepsilon = 0.8$	$\varepsilon = 0.9$	$\varepsilon = 0.95$	$\varepsilon = 0.98$
CIFAR-10	91.29%	91.79%	91.58%	91.94%	91.66%	91.84%

Table 3. The impact of using CPGM or WCLM in PIWCL on the CIFAR-10 and Fashion-MNIST datasets.

	CIFAR-10	FashionMNIST
PIWCL	91.94%	92.34%
w/o CPGM	91.34%	91.65%
w/o WCLM	90.26%	90.76%

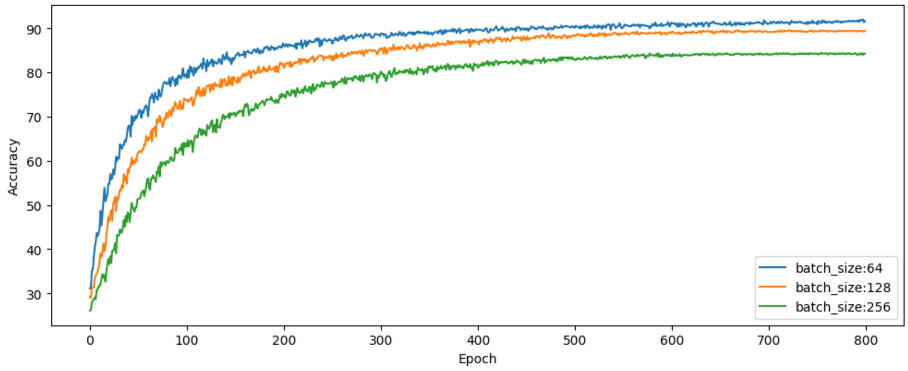

Fig. 5. Accuracy Variation of PIWCL with Different Batch Sizes as Epochs Increase on the CIFAR-10 Dataset.

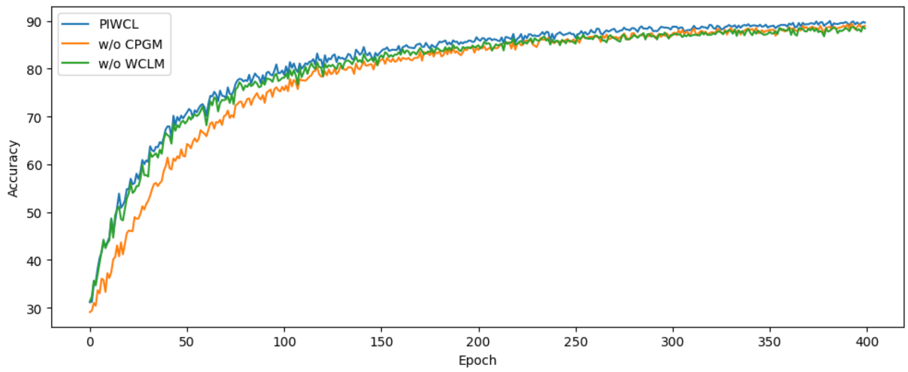

Fig. 4. Comparison of Classification Accuracy of PIWCL with and without Class Prototype Guidance and Weighted Contrastive Learning.

4.2 Experimental Results

PIWCL achieves state-of-the-art performance. Table 1 presents the results of various methods on four datasets: MNIST [32], FashionMNIST [33], Kuzushiji-MNIST [34] and CIFAR-10 [35]. Bold values in the table indicate the best performance. Clearly, PIWCL consistently outperforms all baselines across all datasets. Notably, on the CIFAR-10 dataset, it surpasses the ABLE method by 1.05% and the PICO method by 3.4%. This demonstrates the effectiveness of the PIWCL method.

Impact of Confidence Threshold ε. We investigated the performance of the confidence threshold ε in CPGM at values of 0.6, 0.7, 0.8, 0.9, 0.95 and 0.98 as shown in Table 2. The experiment results indicate that the best performance is achieved when the threshold. This suggests that using more confident samples to update prototypes is beneficial, but a higher threshold is not always better.

The impact of batch size. The best performance is achieved when the batch size is 64 (see Fig. 5). This suggests that smaller batch sizes can increase the flexibility of the training process, helping to escape local minima and find the global optimum, especially when dealing with samples that have ambiguous labels.

Ablation study of class prototype guidance and weighted contrastive learning. To demonstrate the individual effects of the CPGM and the weighted WCLM, we conducted ablation experiments, as shown in Table 3. (1) PIWCL represents the original version. (2) w/o CPGM indicates the model without the class prototype guidance module. (3) w/o WCLM indicates the model without the weighted contrastive learning module. From the table, we can see that PIWCL performs the best. It is important to note that CPGM improves the classifier's performance, especially in the early stages of training (see Fig. 4). When CPGM is not used, the performance in the early stages is significantly worse than that of PIWCL, demonstrating that CPGM enables faster convergence in the initial training phases. Furthermore, without WCLM, the model's performance improves more slowly after 300 epochs and tends to plateau, further proving the necessity of WCLM. These experiments confirm that both CPGM and WCLM are indispensable.

5 Conclusion

In this paper, we propose a simple PLL method called PIWCL. Specifically, to address the issue of exacerbated label ambiguity caused by directly using information from candidate labels, we introduce CPGM and WCLM. In PIWCL, CPGM and WCLM work collaboratively. CPGM effectively enhances the classifier's ability to judge ambiguous labels in candidate labels, helping the model better distinguish positive and negative sample pairs to facilitate the effective training of WCLM. Additionally, it accelerates the model's convergence in the early training stages. Meanwhile, WCLM leverages contrastive learning to generate more compact representations. By employing a reweighting strategy, it avoids confusion among ambiguous class labels. This approach optimizes the decision boundary between positive and negative sample pairs, resulting in more accurate class prototypes. Experimental results demonstrate that our method is effective on benchmark datasets.

References

1. G. Wang, J. Huang, Y. Lai, C.-M. Vong. Dealing with partial labels by knowledge distillation. Pattern Recognition 158, 110965 (2025)
2. J. Yuan, H. Gao, F. Wu, J. Zhao. Learning node representations under partial label learning. In: 2024 International Joint Conference on Neural Networks (IJCNN) 2024, vol. 30, pp. 1–8. IEEE (2024)
3. C. Liu, J. Wen, Y. Liu, C. Huang, Z. Wu, X. Luo, Y. Xu. Masked Two-channel Decoupling Framework for Incomplete Multi-view Weak Multi-label Learning. Advances in Neural Information Processing Systems (NeurIPS) 36, (2024)

4. C. Si, Z. Jiang, X. Wang, Y. Wang, X. Yang, W. Shen. Partial Label Learning with a Partner. Proceedings of the AAAI Conference on Artificial Intelligence 38(13), (2024)
5. W.-X. Bao, Y. Rui, M.-L. Zhang. Disentangled Partial Label Learning. Proceedings of the AAAI Conference on Artificial Intelligence 38(10), (2024)
6. X. Yu, S. Sun, Y. Tian. Self-distillation and self-supervision for partial label learning. Pattern Recognition 146, 110016 (2024)
7. Y. Tian, X. Yu, S. Fu. Partial label learning: Taxonomy, analysis and outlook. Neural Networks 161, 708–734 (2023)
8. C. Liu, W. Zhang, X. Lin, W. Zhang, X. Tan, J. Han, X. Li, E. Ding, J. Wang. Ambiguity-Resistant Semi-Supervised Learning for Dense Object Detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2023, vol. pp. 15579–15588. IEEE (2023)
9. S.-Y. Xia, J. Lv, N. Xu, X. Geng. Ambiguity-Induced Contrastive Learning for Instance-Dependent Partial Label Learning. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence 2022, vol. 4, pp. 3615–3621. (2022)
10. H. Wen, J. Cui, H. Hang, J. Liu, Y. Wang, Z. Lin. Leveraged Weighted Loss for Partial Label Learning. In: International Conference on Machine Learning (ICML) 2021, vol. pp. 11091–11100. PMLR (Proceedings of Machine Learning Research) (2021)
11. J. Lv, M. Xu, L. Feng, G. Niu, X. Geng, M. Sugiyama. Progressive Identification of True Labels for Partial-Label Learning. In: International Conference on Machine Learning (ICML) 2020, vol. pp. 6500–6510. PMLR (Proceedings of Machine Learning Research) (2020)
12. Z. Liao, Y. Xie, S. Hu, Y. Xia. Learning from ambiguous labels for lung nodule malignancy prediction. IEEE Transactions on Medical Imaging 41(7), 1874–1884 (2022)
13. N. Xu, C. Qiao, X. Geng, M.-L. Zhang. Instance-Dependent Partial Label Learning. Advances in Neural Information Processing Systems (NeurIPS) 34, 27119–27130 (2021)
14. L. Feng, J. Lv, B. Han, M. Xu, G. Niu, X. Geng, B. An, M. Sugiyama. Provably Consistent Partial-Label Learning. Advances in Neural Information Processing Systems (NeurIPS) 33, 10,948–10,960 (2020)
15. C.-Z. Tang, M.-L. Zhang. Confidence-Rated Discriminative Partial Label Learning. In: Proceedings of the AAAI Conference on Artificial Intelligence 2017, vol. 31, pp. (2017)
16. Z.G. R. Jin. Learning with multiple labels. Advances in Neural Information Processing Systems (NeurIPS) 15, (2002)
17. X. Liu, L. Sun, S. Feng, X. Liu, L. Sun, S. Feng. Incomplete multi-view partial multi-label learning. Applied Intelligence 2021 52:3 52(3), 1–14 (2024)
18. B.-B.Z. M.-L. Zhang, X.-Y. Liu. Partial label learning via feature-aware disambiguation. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2016, vol. pp. 1335–1344. ACM (2016)
19. F.Y. M.-L. Zhang. Solving the partial label learning problem: An instance-based approach. In: International Joint Conference on Artificial Intelligence (IJCAI) 2015, vol. pp. 4048–4054. IJCAI (2015)
20. J.-Y.H. W.-X. Bao, M.-L. Zhang. Submodular feature selection for partial label learning. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2022, vol. 2, pp. 26–34. ACM (2022)
21. D.-B.W. Dong-Dong Wu, Min-Ling Zhang Revisiting consistency regularization for deep partial label learning. In: Proceedings of the 39th International Conference on Machine Learning 2022, vol. pp. 24,212–24,225. PMLR (2022)

22. S. Rajeswar, P. Rodriguez, S. Singhal, D. Vazquez, A. Courville. Multi-label Iterated Learning for Image Classification with Label Ambiguity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022, vol. pp. 4783–4793. IEEE (2022)
23. W. Hu, Y. Yang, H. Hu. Pseudo label association and prototype-based invariant learning for semi-supervised NIR-VIS face recognition. IEEE Transactions on Image Processing 33, 1448–1463 (2024)
24. Y. Dai, Z. Chen, J. Li, S. Heinecke, L. Sun, R. Xu. Tackling data heterogeneity in federated learning with class prototypes. In: Proceedings of the AAAI Conference on Artificial Intelligence 2023, vol. 37, pp. 7314–7322. (2023)
25. T. Chen, G.-S. Xie, Y. Yao, Q. Wang, F. Shen, Z. Tang, J. Zhang. Semantically meaningful class prototype learning for one-shot image segmentation. IEEE Transactions on Multimedia 24, 968–980 (2022)
26. T. Zhou, W. Wang. Prototype-based semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 46(10), (2024)
27. J.W.R. Ching-Yao Chuang, Yen-Chen Lin, Antonio Torralba, Stefanie Jegelka. Debiased Contrastive Learning. Advances in Neural Information Processing Systems (NeurIPS) 33, 8765–8775 (2020)
28. J. Yang, R. Ding, W. Deng, Z. Wang, X. Qi. Regionplc: Regional point-language contrastive learning for open-world 3d scene understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2024, vol. pp. 19 823–19 832. (2024)
29. P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan. Supervised Contrastive Learning. Advances in Neural Information Processing Systems (NeurIPS) 33, 18 661–18 673 (2020)
30. T. Chen, S. Kornblith, M. Norouzi, G. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In: International Conference on Machine Learning 2020, vol. pp. 1597–1607. PMLR (2020)
31. R.X. Haobo Wang, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, Junbo Zhao. PiCO: Contrastive Label Disambiguation for Partial Label Learning. In: Proceedings of the International Conference on Learning Representations (ICLR) 2022, January, vol. pp. (2022, January)
32. Y. Lecun, L. Bottou, Y. Bengio, P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
33. H. Xiao, K. Rasul, R. Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. (2017)
34. T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, D. Ha. Deep Learning for Classical Japanese Literature. In: 2018 Digital Humanities Conference 2018, vol. pp. (2018)
35. G.H. Alex Krizhevsky. Learning multiple layers of features from tiny images. Handbook of Systemic Autoimmune Diseases 1(4), (2009)
36. F. Hong, J. Yao, Z. Zhou, Y. Zhang, Y. Wang. Long-Tailed Partial Label Learning via Dynamic Rebalancing. (2023)
37. K. He, X. Zhang, S. Ren, J. Sun. Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016, vol. pp. IEEE (2016)