Distilling Reliable Knowledge for Instance-Dependent Partial Label Learning

Dong-Dong Wu*, Deng-Bao Wang*, Min-Ling Zhang†

School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China
{dongdongwu, wangdb, zhangml}@seu.edu.cn

Abstract

Partial label learning (PLL) refers to the classification task where each training instance is ambiguously annotated with a set of candidate labels. Despite substantial advancements in tackling this challenge, limited attention has been devoted to a more specific and realistic setting, denoted as instance-dependent partial label learning (IDPLL). Within this contex, the assignment of partial labels depends on the distinct features of individual instances, rather than being random. In this paper, we initiate an exploration into a self-distillation framework for this problem, driven by the proven effectiveness and stability of this framework. Nonetheless, a crucial shortfall is identified: the foundational assumption central to IDPLL, involving what we term as *partial label knowledge* stipulating that candidate labels should exhibit superior confidence compared to non-candidates, is not fully upheld within the distillation process. To address this challenge, we introduce DIRK, a novel distillation approach that leverages a rectification process to *DIstill Reliable Knowledge*, while concurrently preserves informative fine-grained label confidence. In addition, to harness the rectified confidence to its fullest potential, we propose a knowledge-based representation refinement module, seamlessly integrated into the DIRK framework. This module effectively transmits the essence of similarity knowledge from the label space to the feature space, thereby amplifying representation learning and subsequently engendering marked improvements in model performance. Experiments and analysis on multiple datasets validate the rationality and superiority of our proposed approach.

Introduction

Partial label learning (PLL) is a prominent weakly supervised learning paradigm, which has been studied a lot in the past decade (Guillaumin, Verbeek, and Schmid 2010; Xu, Lv, and Geng 2019; Lv et al. 2020). PLL refers to the classification task where each training instance is associated with a set of candidate labels, among which only one is the ground-truth label. This problem arises naturally in various real-world scenarios, such as automatic image annotation (Briggs, Fern, and Raich 2012; Liu and Dietterich 2012), web mining (Luo and Orabona 2010; Xu et al. 2022), and multimedia content analysis (Zeng et al. 2013).

The major difficulty of PLL lies in label ambiguity, which means the ground-truth label is unknown during training. Over the past decade, non-deep label disambiguation methods have been proposed (Hüllermeier and Beringer 2006; Zhang and Yu 2015; Feng and An 2019; Wang, Li, and Zhang 2019). However, these methods are inefficient for large-scale datasets and high-dimensional features. With the emergence of deep learning, deep PLL methods have been explosively studied (Lv et al. 2020; Feng et al. 2020; Wen et al. 2021; Wu, Wang, and Zhang 2022; Wang et al. 2022b; Qiao, Xu, and Geng 2023; Xu et al. 2023; Jia and Zhang 2022). For example, self-training techniques are utilized in (Lv et al. 2020; Wen et al. 2021) to progressively identify the ground-truth label during training. Wu, Wang, and Zhang (2022) develop a regularized training framework preserving manifold structure of feature space and label space. Recently, various popular techniques such as contrastive learning (Wang et al. 2022b; He et al. 2022), graph neural network (Lyu, Wu, and Feng 2022), optimal transport (Wang et al. 2022a), meta-learning (Xie, Sun, and Huang 2021), have been exploited in deep PLL. However, these studies assume that each false label has a random or fixed probability of being the candidate label. Realistically, annotators prefer to select candidate labels related to the true label, making candidate labels instance-dependent. To cope with the instance-dependent partial label learning (IDPLL) problem, Xu et al. (2021) employ variational inference to estimate the latent label distribution. Xia et al. (2022) perform contrastive learning by utilizing the additional information acquired from the ambiguity. Nevertheless, these methods are either have high computational complexity and difficulty in optimization, or rely on ideal assumptions.

In this paper, we attempt to address IDPLL by leveraging the distillation framework (Allen-Zhu and Li 2020; Mobahi, Farajtabar, and Bartlett 2020) due to its computational efficiency (Kim et al. 2021) and training stability (Furlanello et al. 2018). In this framework, the label confidences produced by the teacher model are distilled to guide the student model training, as suggested in (Xu, Lv, and Geng 2019; Jia et al. 2018). However, we found that the distillation framework faces limitations, as illustrated in "Before rectification" in Figure 1, where the label confidence does not com-

*These authors contributed equally.
†Corresponding Author.

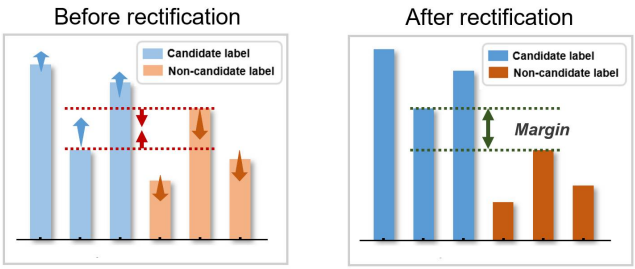


Figure 1: For an instance "cat" with a candidate set {cat, fox, monkey}, the teacher produces inaccurate label confidence, wherein the confidence of non-candidate label "bird" is higher than that of the candidate label "monkey". After rectification, the highest confidence among non-candidate labels is lower than the lowest confidence among candidate labels.

ply with the principles of partial label knowledge. Partial label knowledge dictates that *for each instance, the confidence of each candidate label should be higher than that of any non-candidate label*. This resonates with the empirical fact that labels prone to confusion with the ground-truth labels are more likely to be chosen by annotators.

To address the inherent limitations of the distillation framework in handling IDPLL, we propose an innovative self-distillation framework accompanied by a rectification process to *DIstill Reliable Knowledge*, termed as DIRK. Specifically, we disentangle the teacher's output into two complementary constituents, subsequently rescaling them with an adaptive factor. Experiments demonstrate superiority of this proposed method over other approaches. In addition, to harness the rectified confidence to its fullest potential, we propose a knowledge-based representation refinement module, seamlessly integrated into the DIRK framework. This module effectively transmits the essence of similarity knowledge from the label space to the feature space, thereby amplifying representation learning and subsequently engendering marked improvements in model performance. Our contributions can be summarized as follows:

- We consider the more realistic instance-dependent partial label learning problem and for the first time exploit the distillation framework to solve this problem.
- Oberserving the distortion of partial label knowledge through direct application of distillation, we propose DIRK with an incorporated rectification process to distill more reliable knowledge.
- Through theoretical analysis, we shed light on the underlying rationale of the rectification process, unveiling that it not only rectifies the knowledge but also in serving as a mechanism for hard sample mining.
- We further extend the distilled reliable knowledge into the representation space through the development of a knowledge-based representation refinement module, yielding an additional boost to performance.
- Extensive evaluations demonstrate the superiority of our approach over state-of-the-art methods.

Related Work

Partial Label Learning

Partial label learning is also known as ambiguous-label learning (Chen, Patel, and Chellappa 2017) or superset-label learning (Gong et al. 2017), has been extensively studied in the past decade (Wang and Zhang 2020; Wang et al. 2022b; Yang, Li, and Jiang 2023). PLL methods can be categorized into *non-deep PLL* and *deep PLL* methods, where non-deep methods often use linear or kernel-based models, while deep methods adopt stochastic optimization on deep neural networks. Non-deep PLL methods usually disambiguate partial labels by averaging (Hüllermeier and Beringer 2006) or identification strategy (Yu and Zhang 2015; Wang, Li, and Zhang 2019). These methods are inefficient for large-scale datasets and high-dimensional features.

With the emergence of deep learning, deep PLL methods have been explosively studied recently (Lv et al. 2020; Wen et al. 2021; Wu, Wang, and Zhang 2022; Wang et al. 2022b; Gong, Yuan, and Bao 2022; Qiao, Xu, and Geng 2023; Xu et al. 2023). Lv et al. (2020) use a simple self-training strategy to progressively identify ground-truth labels during network training. Similarly, Wen et al. (2021) introduce a family of loss functions named leveraged weighted loss. Feng et al. (2020) propose a risk-consistent method and a classifier-consistent method under the uniform partial label generation assumption. Wang et al. (2022b) propose a prototype-based disambiguation mechanism via contrastive learning. Wu, Wang, and Zhang (2022) revisit a consistency regularization in PLL for the first time, and they regard the method in (Lv et al. 2020) as one of its special cases. Wang et al. (2022b) propose a prototype-based disambiguation mechanism by leveraging the contrastively learned embeddings. Realistically, IDPLL is a more practical setting in many real-world scenarios. To cope with that, a series of methods like contrastive learning, bayesian models and variational inference have been proposed for IDPLL (Xu et al. 2021; Xia et al. 2022; Qiao, Xu, and Geng 2023; Xu et al. 2023). However, these existing methods often computationally expensive or rely on ideal assumptions.

Knowledge Distillation

Self-Distillation is proposed to utilize a model's own knowledge without extra networks in knowledge distillation (Hinton et al. 2014; Zhou, Jiang, and Chen 2003; Zhou and Jiang 2004; Ji et al. 2023; Yun et al. 2020). A common practice in self-distillation is to directly use outputs from a teacher whose architecture is exactly the same as the student. Born-Again Networks (BANs) (Furlanello et al. 2018) is the initial self-distillation method where a student model is trained sequentially, with later generations supervised by earlier ones. The student generations are then assembled into an aggregate model. Similar to BANs, Yuan et al. (2020) empirically show teacher-free distillation, where a pretrained student teaches a single new student. Kim et al. (Kim et al. 2021) progressively distill a model's own knowledge to soften hard targets during training. Other works like (Ji et al. 2023; Xu and Liu 2019; Yun et al. 2020) use a single network to enforce consistent predictions between augmented data (Xu

and Liu 2019) or intra-class samples (Yun et al. 2020). Despite successes in improving classification performance by developing advanced architectures, the logit knowledge does not conform to the partial label knowledge in IDPLL setting.

The Proposed Method

Preliminaries

Let $\mathcal{X} \subset \mathbb{R}^q$ be the q-dimensional feature space and $\mathcal{Y} = \{1, 2, \cdots, c\}$ denote the label space with c distinct labels. We are given a partial label dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \mathcal{S}_i) | 1 \leq i \leq n\}$, where the candidate label set $\mathcal{S}_i \subset \mathcal{Y}$ always contains the ground-truth label y_i.

We choose vanilla self-distillation as the basis of our method, because it can represent the basic model of most self-distillation methods. In the vanilla self-distillation framework, $\mathbf{T}(\cdot)$ and θ_T represent the teacher model and its parameters, while $\mathbf{S}(\cdot)$ and θ_S stand for the student model and the corresponding parameters. A common self-distillation strategy is empolyed to update the teacher as the running average of the student model with momentum: $\theta_T \leftarrow m\theta_T + (1-m)\theta_S$ (Kim et al. 2021; Shen et al. 2022; Tejankar et al. 2021). The student model is learned by optimizing the following objective:

$$\mathcal{L}_{\text{KL}} = \text{CE}(\alpha \boldsymbol{Y} + (1-\alpha)\mathbf{T}(\boldsymbol{x})), \mathbf{S}(\boldsymbol{x})), \tag{1}$$

where $\text{CE}(\cdot, \cdot)$ means the commonly used cross-entropy loss, and α is the trade-off hyperparameter used to balance the supervision from oracle and teacher model. Here, the oracle supervision means the uniform label confidence $\boldsymbol{Y}$ on partial labels:

$$\boldsymbol{Y}_k = \begin{cases} \frac{1}{|\mathcal{S}|} & \text{if } k \in \mathcal{S}, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The Destruction of Partial Label Knowledge

Figure 2 (left) depicts the illustrative experiments with various α on the CIFAR-10-ID dataset. When $\alpha = 0$, no oracle supervision is available, and the student learns blindly. In this case, each class has an equal probability of being the true label, which leads to the model collapsing issue. When $\alpha > 0$, partial label knowledge is induced by the oracle supervision $\boldsymbol{Y}$, which encourages the model's confidence of candidate labels to be higher than that of non-candidate labels. As is also shown, the larger α is, the faster the student learns in the early stages. It implies that **partial label knowledge is essential in the early stage of distillation**. However, we also observe that if $\boldsymbol{Y}$ continues to dominate the learning in the later stage, the student would overfit this coarse-grained knowledge, leading to a decrease in accuracy. On the other hand, although a small α faces the underfitting problem in the early stage, informative label confidence induced by the teacher model makes a stable learning procedure in the later stage, which implies that **fine-grained knowledge is essential for further learning in the later stage**.

Figure 2 (right) shows the destruction of partial label knowledge in the distillation process. Specifically, we define the correct rate as the proportion of instances whose label confidence satisfies the partial label knowledge in the final epoch. The general trend of correct rates is clearly different from that of test accuracy. In particular, a larger α benefits the reliability of teacher knowledge, but it leads the student model to overfit partial labels. With small α, partial label knowledge is not maintained anymore and hence resulting in an underfitting problem. When $\alpha = 0.2$ yields the best accuracy among all choices in our experiment, the distillation process still contains a large amount of misleading knowledge. Therefore, even with a proper balance factor for distillation, there is still much room to further improve the quality of teacher knowledge for IDPLL.

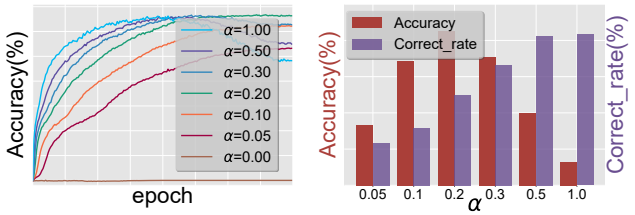


Figure 2: With fixed $m = 0.99$ and various α, (left figure) shows the accuracy curve of the student model, and (right figure) shows the test accuracy of the student model and correct rate of distilled label confidence.

Rectification of Label Confidence

A simple strategy to alleviate the destruction of partial label knowledge is to force the distilled label confidences of non-candidate labels to be zero. However, this process will eliminate the informative dark knowledge on instance's complementary labels, which has been shown important to improve model performance (Hinton et al. 2014; Chen et al. 2020). To better utilize the label confidence from the teacher model, it is crucial to alleviate the destruction of partial label knowledge and preserve the fine-grained information of label confidence on all labels. To this end, we propose to rescale the sum of label confidences for both candidate labels and non-candidate labels, thereby rectifying the misleading of partial label knowledge. We first decouple the teacher outputs into two complementary components $\mathbf{T}^c(\boldsymbol{x})$ and $\mathbf{T}^n(\boldsymbol{x})$, which can be calculated as:

$$\begin{aligned} \mathbf{T}^c_i(\boldsymbol{x}) &= \frac{\exp(\delta_i)}{\sum_{k \in \mathcal{S}} \exp(\delta_k)} \text{ if } i \in \mathcal{S}, \text{and } 0 \text{ otherwise,} \\ \mathbf{T}^n_i(\boldsymbol{x}) &= \frac{\exp(\delta_i)}{\sum_{k \notin \mathcal{S}} \exp(\delta_k)} \text{ if } i \notin \mathcal{S}, \text{and } 0 \text{ otherwise,} \end{aligned} \tag{3}$$

where δ_i represents the logit output of the teacher model on the i-th class of $\boldsymbol{x}$. $\mathbf{T}^c(\boldsymbol{x})$ denotes the teacher's output confidence on candidate labels and $\mathbf{T}^n(\boldsymbol{x})$ stands for the output confidence on all non-candidate labels. Then, we utilize a scaling process on these two terms to rectify the teacher's label confidence:

$$\widetilde{\mathbf{T}}(\boldsymbol{x}) = \gamma \cdot \mathbf{T}^c(\boldsymbol{x}) + (1-\gamma) \cdot \mathbf{T}^n(\boldsymbol{x}). \tag{4}$$

The following proposition states how to determine the scaling factor γ to make $\widetilde{\mathbf{T}}(\boldsymbol{x})$ comply with the above discussed partial label knowledge assumption.

Proposition 1 *With the rescaling in (4) and* $\gamma = \frac{\eta \max_j \mathbf{T}_j^n(\boldsymbol{x})}{\eta \max_j \mathbf{T}_j^n(\boldsymbol{x}) + \min_i \mathbf{T}_i^c(\boldsymbol{x})}$*, for* $\eta > 1$*, the rescaled label confidence satisfeis* $\min_i \widetilde{\mathbf{T}}_i(\boldsymbol{x}) = \eta \max_j \widetilde{\mathbf{T}}_j(\boldsymbol{x}) > \max_j \widetilde{\mathbf{T}}_j(\boldsymbol{x})$*, where* $i \in \mathcal{S}$ *and* $j \notin \mathcal{S}$*.*

The proof is presented in Appendix A.1. Now, the reactificated label confidence $\widetilde{\mathbf{T}}(\boldsymbol{x})$ with proper parameter η complies the above assumption, then we can discard $\boldsymbol{Y}$ that is used to preserve partial label knowledge as in Eq. (1). Therefore, the new distilled label confidence would not induce the overfitting problem caused by the uniform label confidence $\boldsymbol{Y}$. The rectificated distillation loss function could be simplified as:

$$\mathcal{L}_{\texttt{DIRK}}(\boldsymbol{x}) = \text{CE}\left(\widetilde{\mathbf{T}}(\boldsymbol{x}), \mathbf{S}(\boldsymbol{x})\right). \tag{5}$$

In this way, the parameter η presents the margin between the output confidence of candidate labels and non-candidate labels. The pseudo-code of our complete algorithm DIRK is shown in Appendix A.3. We next show that η can be chosen in an instance-dependent manner during training.

The choice of η. A simple principle to choose η is making the student model focus more on instances that are not yet properly classified in the current training iteration. Denote $p^c = \frac{\sum_{i \in \mathcal{S}} \exp(\delta_i)}{\sum_{j \in \mathcal{Y}} \exp(\delta_j)}$ as the probability that the model's prediction lies in candidate labels. We achieve this principle with $\eta = 1/p^c$. By this, the rectification strength applied to the teacher's raw outputs could be determined adaptively during training. For instances that tend to be misclassified by the teacher model, a large η can be assigned to increase the margin as shown in Figure 1. In contrast, a relatively small η is assigned to smooth the label confidence for instances that the model has properly learned. This principle can be formally explained from the gradient perspective, as is shown in Proposition 2.

Proposition 2 *Under the assumption that the recitificated label confidence of non-candidate labels of the teacher model is lower than that of the student model, i.e.,* $\boldsymbol{S}_i(\boldsymbol{x}) - \widetilde{\mathrm{T}}_i(\boldsymbol{x}) > 0$*, for* $i \notin \mathcal{S}$*, there exists a subset* $\mathcal{S}'$ *of candidate labels that satisfies* $\boldsymbol{S}_i(\boldsymbol{x}) - \widetilde{\mathrm{T}}_i(\boldsymbol{x}) < 0$*, for* $i \in \mathcal{S}'$*. Then* L_1 *norm of the gradient of* $\mathcal{L}_{DIRK}$ *w.r.t. the logit is given by:*

$$\sum_i \left| \frac{\partial \mathcal{L}_{DIRK}}{\partial \delta_i} \right| = 2 \sum_{i \in \mathcal{S}'} \left(\gamma \boldsymbol{T}_i^c(\boldsymbol{x}) - \boldsymbol{S}_i(\boldsymbol{x}) \right). \tag{6}$$

The proof can be found in Appendix A.2. In practice, the assumption in Proposition 2 is generally well satisfied as the teacher model tends to produce more discriminative outputs than the student model. As demonstrated, a larger gradient norm can be derived with a greater γ. By the relationship between γ and η, we have $\gamma \propto 1/p^c$. Therefore, Eq. (6) deduces that instances with smaller p^c would contribute more gradient in each update process.

Representation Refinement Module

To fully leverage the high-quality rectificated label confidence, we further introduce a knowledge-based representation refinement module to enhance the learning of latent features. Following the architecture and pepeline in (Xia et al. 2022), an instance is fed into the encoder network $f(\cdot)$ which maps to a latent representation $\boldsymbol{v} = f(\boldsymbol{x}) \in \mathbb{R}^{d_e}$, Afterwards we utilize the projection network $g(\cdot)$ to map $\boldsymbol{v}$ to a low-dimensional embedding $\boldsymbol{z} = g(\boldsymbol{v}) \in \mathbb{R}^{d_p}$. Here, the representation $\boldsymbol{v}$ is also fed to the last full-connected classifier $h(\cdot)$ to make the final prediction $\boldsymbol{p} = h(\boldsymbol{v}) \in \mathbb{R}^c$. For the teacher model, we maintain an embedding queue $\boldsymbol{E}$ and label confidence queue $\boldsymbol{I}$ storing the embeddings and rectificated label confidences respectively, which are updated chronologically. To this end, we have the following embedding pool $\mathcal{E}$ and rectificated label confidence pool $\mathcal{I}$:

$$\mathcal{E} = \mathcal{B}_T^E \cup \mathcal{B}_S^E \cup \boldsymbol{E}, \;\; \mathcal{I} \;\; = \mathcal{B}_T^L \cup \mathcal{B}_T^L \cup \boldsymbol{I}, \tag{7}$$

where $\mathcal{B}_\cdot^E$/ $\mathcal{B}_\cdot^L$ are vectorial embeddings and rectificated label confidences of the current mini-batch.

Since the rectificated label confidence preserves informative knowledge, we assume that the similarity between instances in the output space can be converted to the embedding space. Formally, this can be achieved by the following representation refinement loss:

$$\mathcal{L}_{\texttt{REF}}(\boldsymbol{x}) = \sum_{j \in P(\boldsymbol{x})} w_j \log \frac{\exp\left(\text{sim}\left(\boldsymbol{z}, \mathcal{E}_j\right)/\tau_2\right)}{\sum_{k \in \mathcal{E}(\boldsymbol{x})} \exp\left(\text{sim}\left(\boldsymbol{z}, \mathcal{E}_k\right)\right)/\tau_2)}, \tag{8}$$

where

$$w_j = \frac{\exp(\text{sim}(\widetilde{\mathbf{T}}(\boldsymbol{x}), \mathcal{I}_j)/\tau_1)}{\sum_{k \in P(\boldsymbol{x})} \exp(\text{sim}(\widetilde{\mathbf{T}}(\boldsymbol{x}), \mathcal{I}_k)/\tau_1)}. \tag{9}$$

$P(\boldsymbol{x})$ denotes the index of positive instance in the rectificated label confidence pool, and $\mathcal{E}(\boldsymbol{x})$ represents the index of $\mathcal{E} \backslash \{\boldsymbol{z}\}$. In Eq. (8) and Eq. (9), $\text{sim}(\cdot, \cdot)$ refers to the similarity between two vectors, and τ_1 and τ_2 are temperature factors used to scale the similarity. In our experiments, we use cosine similarity which is widely used in most self-distillation methods (Tung and Mori 2019; Tejankar et al. 2021). Note that we only select positive instances that satisfy the following two requirements for an anchor $\boldsymbol{x}$: (1) instances that share candidate labels with the anchor instance; (2) instances with the same label prediction as the anchor.

It is noteworthy that $\mathcal{L}_{\texttt{REF}}$ can be considered as a generalized form of recent contrastive loss used in PLL problem (Wang et al. 2022b; Xia et al. 2022), although this representation refinement loss is derived from the perspective of feature distillation. In their research, the weights of positive instances are set equally or label-wise, while the weight assigned to each positive instance is instance-wise in our work. Inspired (Tejankar et al. 2021), which argues that not all negative instances are equally negative in self-supervised learning, we make the reasonable assumption that *not all positive instances are equally positive*. Instance-wise contrastive learning can provide more discriminative information in representation learning.

The architecture of **RE**presentation re**F**inement module is described in Appendix A.4. After equipping the representation refinement module into DIRK, the overall loss function can be formulated as:

$$\mathcal{L}_{\texttt{DIRK-REF}} = \mathcal{L}_{\texttt{DIRK}} + \lambda \mathcal{L}_{\texttt{REF}}, \tag{10}$$

where λ is the trade-off parameter between the label distillation and the representation refinement module. Thoroughly, the pseudo-code and flowchart of DIRK-REF is shown in Appendix A.4.

Experiment

Datasets. We evaluated our method on seven commonly used benchmark image dataset: Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), Kuzushiji-MNIST (Clanuwat et al. 2018), CIFAR-10 (Krizhevsky, Hinton et al. 2009), CIFAR-100 (Krizhevsky, Hinton et al. 2009), CUB-200 (Welinder et al. 2010), Flower (Nilsback and Zisserman 2008) and Oxford-IIIT Pet (Parkhi et al. 2012). We manually corrupted these datasets into IDPLL datasets by an instance-dependent generating process (Xu et al. 2021). Besides, five real-world PLL datasets collected from different application domains were used, including Lost (Cour, Sapp, and Taskar 2011), Soccer Player (Zeng et al. 2013), Yahoo! News (Guillaumin, Verbeek, and Schmid 2010), MSRCv2 (Liu and Dietterich 2012), and BirdSong (Briggs, Fern, and Raich 2012). Due to the space limitation, the detailed generation process of corrupted benchmarkd datasets and characteristics of PLL datasets are introduced in Appendix A.5.

Compared Methods. We compared our method against the following deep PLL methods: (1) POP (Xu et al. 2023), a progressive purification approach that iteratively purifies labels and refines the classifier; (2) IDGP (Qiao, Xu, and Geng 2023), an approach that models instance-dependence via decomposed categorical and Bernoulli distributions, and uses MAP optimization; (3) ABLE (Xia et al. 2022), a contrastive learning method that leverages additional information in the partial labels; (4) VALEN (Xu et al. 2021), an instance-dependent method that recovers latent label distributions using variational inference; (5) PICO (Wang et al. 2022b), a contrastive learning method that identifies true labels using learned prototypes; (6) CAVL (Zhang et al. 2021), a method that selects the true label based on the class activation value; (7) CR-DPLL (Wu, Wang, and Zhang 2022), a regularized training framework using consistency regularization on candidate labels; (8) LWS (Wen et al. 2021), a discriminative approach balancing losses on candidate and non-candidate labels; (9) RC (Feng et al. 2020), a risk-consistent method using importance re-weighting strategy; (10) CC (Feng et al. 2020), a classifier-consistent method applying the cross entropy loss and transition matrix to form an empirical risk estimator; (11) PRODEN (Lv et al. 2020), a self-training like method that progressively identifies the true labels.

Implementation details. To make fair comparisons, we used the same network architecture, learning rate, optimizer, and augmentation strategy across all compared methods in various datasets. Our implementation was executed using PyTorch (Paszke et al. 2019), and all experiments were conducted with NVIDIA Tesla V100 GPU. For the encoder network $f(\cdot)$, we use ResNet-18 (He et al. 2016) on Fashion-MNIST, Kuzushiji-MNIST, and ResNet-34 (He et al. 2016) on other datasets. The normalized activations of the final pooling layer (d_e=512) were used as the representation. For the projection network $g(\cdot)$, we instantiated $g(\cdot)$ with a multi-layer perceptron with a single hidden layer of size 512 (as well as ReLU activation) and output representation of size $d_p = 128$. For the classifier $h(\cdot)$, we instantiated $h(\cdot)$ with a single linear layer with the softmax activation. We set the momentum hyperparameter m as 0.99 and the trade-off parameter λ as 0 in DIRK. It is worth noting that we only discuss the performance of DIRK-REF in the last subsection, where temperature hyperparameters $\tau_1 = 0.01$, $\tau_2 = 0.07$, and the sizes of both queues are fixed to be 1024.

For all methods on benchmark datasets, we used SGD as the optimizer with a momentum of 0.9, a weight decay of 1e-3, an initial learning rate of 1e-2, and set the epoch number to 500. We adopted cosine annealing learning rate scheduling and did not use any pre-training models. For large-scale datasets CUB-200, Flower, and Oxford-IIIT Pet, we set the mini-batch size as 32, while 256 for other datasets. For all methods on real-world datasets, we adopted the widely-used linear model as the backbone. Since data augmentation cannot be employed on real-world datasets that contain extracted features from audio and video data. We adopt no data augmentation strategy for data-augmentation-free methods (POP, IDGP, VALEN, PRODEN, RC, CC, LWS, and CAVL) to make fair comparisons. The hyperparameters were selected so as to maximize the accuracy on a validation set (10% of the training set). We recorded the mean and standard deviation in each case based on five independent runs with different random seeds.

Experiment Results

DIRK achieves SOTA results. We report the classification accuracy of all methods on benchmark and real-world datasets in Table 1 and Table 2, respectively. Table 1 shows that DIRK consistently outperforms all compared methods, demonstrating its superiority in handling instance-dependent partial labels. Notably, as dataset size increases, DIRK's performance advantage becomes more pronounced, with consistent and stable improvements over other methods. Since the linear model provides limited information, we simply set γ to 1 in solving PLL real-world datasets. In Table 2, DIRK significantly outperforms all compared approaches in Lost, BirdSong, and Yahoo! News. Even on MSRCv2 and Soccer Player, DIRK exhibits stronger performance than other methods. We conjecture DIRK's insignificant improvements arise from semantic sparsity and limited training data. Constructing large-scale semantically rich real-world datasets remains future work.

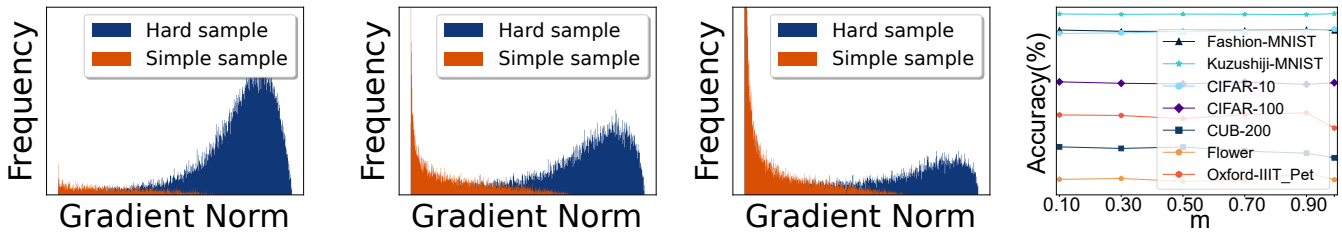
Analysis of the gradient norm. Proposition 2 suggests adopting an adaptive η in the rectification process can enable hard samples to contribute larger gradients during each iteration, thereby facilitating hard sample mining. To investigate this, we conducted an analysis of the gradient norm evolution over training for both simple and hard samples. Specifically, simple samples are defined as those where the student model exhibits high confidence in the true label, while hard samples are those with low true label confidence. The results are illustrated in Figure 4 (a-c). As the training epochs increase, we can observe that more hard samples become simple ones. However, the remaining hard samples consistently exhibit higher gradient norms than simple samples, indicating that they still contribute significantly more to the

Method	F-MNIST	K-MNIST	CIFAR10	CIFAR100	CUB-200	Flower	Pet
DIRK	**91.48 ± 0.21%**	**96.80 ± 0.52%**	**90.87 ± 0.25%**	**68.77 ± 0.49%**	**49.29 ± 1.00%**	**44.03 ± 0.02%**	**64.95 ± 2.11%**
POP	81.91 ± 0.37%	94.99 ± 1.18%	89.55 ± 0.36%	64.57 ± 0.37%	39.13 ± 0.14%	33.02 ± 2.62%	53.83 ± 1.55%
IDGP	85.14 ± 0.39%	93.88 ± 0.72%	84.12 ± 0.99%	62.27 ± 1.89%	46.71 ± 0.64%	41.26 ± 1.33%	59.74 ± 0.71%
ABLE	89.81 ± 0.08%	94.67 ± 0.02%	83.92 ± 0.67%	63.92 ± 0.39%	45.82 ± 0.27%	43.51 ± 0.93%	54.19 ± 1.01%
VALEN	82.91 ± 0.12%	90.09 ± 0.28%	81.29 ± 0.39%	60.19 ± 0.82%	31.94 ± 0.49%	32.89 ± 0.88%	50.74 ± 0.91%
CR-DPLL	79.67 ± 6.95%	92.78 ± 4.04%	80.29 ± 0.18%	56.93 ± 1.47%	42.15 ± 1.70%	42.42 ± 0.12%	49.13 ± 2.49%
CAVL	78.06 ± 3.10%	91.08 ± 6.06%	79.89 ± 0.07%	59.76 ± 0.49%	41.67 ± 0.82%	40.69 ± 1.93%	50.64 ± 0.11%
PICO	85.92 ± 0.38%	92.12 ± 0.42%	82.91 ± 1.43%	58.56 ± 0.13%	40.21 ± 0.84%	33.76 ± 1.30%	62.64 ± 0.45%
PRODEN	83.77 ± 0.58%	92.51 ± 0.67%	83.18 ± 0.29%	67.77 ± 0.38%	41.85 ± 0.26%	41.28 ± 0.57%	52.77 ± 0.66%
LWS	75.52 ± 0.29%	83.61 ± 2.11%	78.32 ± 2.11%	65.74 ± 0.60%	15.99 ± 0.60%	21.24 ± 0.63%	31.21 ± 0.47%
RC	84.87 ± 1.48%	93.21 ± 0.78%	87.53 ± 0.94%	65.26 ± 0.40%	42.04 ± 0.24%	41.20 ± 0.23%	54.01 ± 0.05%
CC	79.98 ± 0.01%	92.08 ± 0.02%	82.14 ± 0.01%	62.28 ± 0.01%	40.87 ± 0.28%	40.92 ± 0.40%	52.99 ± 1.20%

Table 1: Accuracy comparison (mean ± std) on benchmark datasets, with the best result among each column highlighted in bold and the second best result in each column underlined.

Method	Lost	BirdSong	MSRCv2	Soccer Player	Yahoo!News
DIRK	**79.24±0.63%**	**74.52±0.23%**	48.59±0.28%	55.83±0.35%	**67.65±0.32%**
POP	78.57±0.45%	74.47±0.36%	45.86±0.28%	54.48±0.10%	66.38±0.07%
IDGP	77.02±0.82%	74.23±0.17%	**50.45±0.47%**	**55.99±0.28%**	66.62±0.19%
VALEN	76.87±0.86%	73.39±0.26%	49.97±0.43 %	55.81±0.10 %	66.26±0.13%
CAVL	75.89±0.42%	73.47±0.13 %	44.73±0.96%	54.06±0.67%	65.44±0.23%
RC	76.26±0.46%	69.33±0.32%	49.47±0.43 %	56.02±0.59 %	63.51±0.20 %
CC	63.54±0.25 %	69.90±0.58 %	41.50±0.44 %	49.07±0.36 %	54.86±0.48 %
LWS	73.13±0.32 %	51.45±0.26 %	49.85±0.49 %	50.24±0.45 %	48.21±0.29 %
PRODEN	76.47±0.25 %	73.44±0.12 %	45.10±0.16 %	54.05±0.15%	66.14±0.10%

Table 2: Accuracy comparison (mean±std) on real-world partial label datasets, with the best result among each column highlighted in bold and the second best result in each column underlined.



(a) Epoch=10 (b) Epoch=20 (c) Epoch=30 (d) Impact of hyperparameter m

Figure 3: (a-c) show the change of gradient norm for simple and hard samples over increasing epochs. (d) demonstrates the performance of DIRK with varing values of momentum hyperparameter.

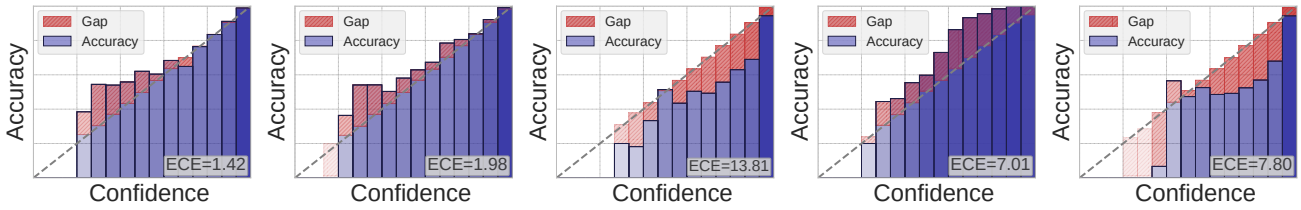
gradient. Consequently, the adaptive η in rectification process is essential for enabling continuous hard sample mining during training.

Effect of momentum hyperparamter m. We also wonder how the momentum hyperparameter, which controls the updation rate of the teacher model, impacts performance. Figure 4 (d) shows the accuracy with varied values $\{0.1, 0.3, 0.5, 0.7, 0.9, 0.99\}$. We can observe that our proposed method DIRK exhibits stability on all benchmark datasets. This implies the method is insensitive to the momentum value, ensuring efficient deployment in practice without requiring extensive hyperparameter tuning. Due to the space limitation, ablation experiments related to γ can be found in Appendix A.6.

Uncertainty quantification of DIRK. When deploying machine learning systems, algorithms must be not only accurate but also trustworthy, aware of potential errors (Guo et al. 2017). Thus, we preliminarily investigated the output uncertainty of different PLL methods, as shown in Figure 4. DIRK(TEA) and DIRK(STU) refer to the teacher and student models respectively. The results demonstrate DIRK improved model output confidence by aligning it with the expected data distribution, achieving competitive ECE scores. Specifically, our method obtained the lowest ECE score

	F-MNIST	K-MNIST	CIFAR-10	CIFAR-100	CUB-200	Flower	Pet
$\lambda = 0$	$91.48 \pm 0.21\%$	$96.80 \pm 0.52\%$	$90.87 \pm 0.25\%$	$68.77 \pm 0.49\%$	$49.29 \pm 1.00\%$	$44.03 \pm 0.02\%$	$64.95 \pm 2.11\%$
$\lambda = 0.1$	$92.01 \pm 0.24\%$	$\mathbf{98.31 \pm 0.20\%}$	$93.50 \pm 0.16\%$	$70.94 \pm 1.17\%$	$50.91 \pm 0.24\%$	$47.66 \pm 0.74\%$	$68.28 \pm 0.14\%$
$\lambda = 0.3$	$\mathbf{92.10 \pm 0.08\%}$	$98.14 \pm 0.20\%$	$94.00 \pm 0.13\%$	$70.72 \pm 0.54\%$	$52.78 \pm 0.15\%$	$50.24 \pm 0.31\%$	$68.80 \pm 0.33\%$
$\lambda = 0.5$	$91.88 \pm 0.32\%$	$98.09 \pm 0.17\%$	$94.24 \pm 0.03\%$	$71.53 \pm 1.35\%$	$51.89 \pm 0.11\%$	$\mathbf{52.18 \pm 0.28\%}$	$68.27 \pm 0.13\%$
$\lambda = 0.7$	$92.03 \pm 0.58\%$	$97.84 \pm 0.18\%$	$\mathbf{94.25 \pm 0.26\%}$	$\mathbf{71.72 \pm 0.63\%}$	$52.91 \pm 0.24\%$	$48.06 \pm 0.16\%$	$\mathbf{68.95 \pm 0.12\%}$
$\lambda = 1.0$	$91.88 \pm 0.36\%$	$97.58 \pm 0.32\%$	$93.73 \pm 0.31\%$	$70.61 \pm 0.85\%$	$\mathbf{52.93 \pm 0.31\%}$	$48.26 \pm 0.38\%$	$68.78 \pm 0.42\%$

Table 3: Accuracy comparison (mean $\pm$ std) with diffrent λ of the representation refinement module.



(a) DIRK (TEA) (b) DIRK (STU) (c) PRODEN (d) CC (e) RC

Figure 4: Reliability diagram and expected calibration error on CIFAR-10. Darker color of bars indicates that more samples are assigned with the corresponding confidence intervals.

among compared methods on CIFAR10. Due to space limitations, we present reliability histograms on other datasets and further detail analysis in Appendix A.7.

DIRK-REF: Enhancing DIRK with Knowledge-Based Representation Refinement

To fully exploit the knowledge of rectificated label confidences, we equip DIRK with a knowledge-based representation refinement module, termed DIRK-REF, whose loss function is Eq. (10). Implementation details are described before. The comparison results on various datasets with different λ values are reported in Table 3. The empirical results show that DIRK-REF consistently outperforms DIRK across all datasets, validating the efficacy of the knowledge-based representation refinement module. Furthermore, we find that the performance gains on complex datasets benefit from a larger λ, reflecting the importance of refinement for difficult data. Complete results of DIRK-REF on real-world datasets are presented in Appendix A.9, demonstrating conclusions consistent with the main results.

Analysis of τ_1 and queue size. The above results used $\tau_1 = 0.01$ and $\tau_2 = 0.07$, with $\tau_2 = 0.07$ being common in complementations of contrastive learning (He et al. 2020; Khosla et al. 2020; Xia et al. 2022; Wang et al. 2022b). Thus, we only analyze the robustness of DIRK-REF to varying τ_1 in Appendix 10.

Discriminative weight analysis in contrastive learning. To reflect the effect of the fine-grained weight Eq. (9), i.e. instance-wise weight of the representation refinement module, we compared DIRK-REF to two variant methods: DIRK-PICO and DIRK-ABLE. DIRK-PICO refers to setting the weight of all positive instances to be equal, while DIRK-ABLE adjusts the weights of positive instances based on their pseudo labels. As results observed in Appendix A.11, our proposed method DIRK-REF achieves the best performance across all four datasets. The instance-wise contrastive learning aligns with intuitive expectations. For example, a Sphynx cat should not be assigned the same weight as a Garfield cat, even though they are both cats. Ablation experiments analyzing the impact of batch size on DIRK-REF are presented in Appendix A.6.

Conclusion

In this work, we considered a more realistic setting of PLL problem where the candidate labels are instance-dependent. We for the first time explored a self-distillation framework for this problem and pointed out that partial label knowledge should be preserved during distillation. Based on the observation of the destruction of partial label knowledge in vanilla self-distillation, we proposed a practical framework named DIRK that utilizes a rectification process to distill reliable knowledge in training. Furthermore, a representation refinement module, which transfers the reactificated knowledge into the latent embedding space, is proposed to be incorporated into DIRK. Our experiments on benchmark datasets and real-world PLL datasets demonstrate the superiority of our proposed method compared with other state-of-the-art PLL methods. Source code is available at https://github.com/wu-dd/DIRK.

Acknowledgments

The authors wish to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Science Foundation of China (62176055), and the Big Data Computing Center of Southeast University.

References

Allen-Zhu, Z.; and Li, Y. 2020. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *International Conference on Learning Representations*.

Briggs, F.; Fern, X. Z.; and Raich, R. 2012. Rank-loss support instance machines for MIML instance annotation. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 534–542.

Chen, C.-H.; Patel, V. M.; and Chellappa, R. 2017. Learning from ambiguously labeled face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40: 1653–1667.

Chen, S.; Wang, J.; Chen, Y.; Shi, Z.; Geng, X.; and Rui, Y. 2020. Label distribution learning on auxiliary label space graphs for facial expression recognition. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*, 13984–13993.

Clanuwat, T.; Bober-Irizar, M.; Kitamoto, A.; Lamb, A.; Yamamoto, K.; and Ha, D. 2018. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*.

Cour, T.; Sapp, B.; and Taskar, B. 2011. Learning from partial labels. *Journal of Machine Learning Research*, 12: 1501–1536.

Feng, L.; and An, B. 2019. Partial Label Learning by Semantic Difference Maximization. In *International Joint Conference on Artificial Intelligence*, 2294–2300.

Feng, L.; Lv, J.; Han, B.; Xu, M.; Niu, G.; Geng, X.; An, B.; and Sugiyama, M. 2020. Provably consistent partial-label learning. In *Advances in neural information processing systems*, 10948–10960.

Furlanello, T.; Lipton, Z.; Tschannen, M.; Itti, L.; and Anandkumar, A. 2018. Born again neural networks. In *International Conference on Machine Learning*, 1607–1616.

Gong, C.; Liu, T.; Tang, Y.; Yang, J.; Yang, J.; and Tao, D. 2017. A regularization approach for instance-based superset label learning. *IEEE Transactions on Cybernetics*, 48: 967–978.

Gong, X.; Yuan, D.; and Bao, W. 2022. Partial Label Learning via Label Influence Function. In *International Conference on Machine Learning*, 7665–7678.

Guillaumin, M.; Verbeek, J.; and Schmid, C. 2010. Multiple instance metric learning from automatically labeled bags of faces. In *European Conference on Computer Vision*, 634–647.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, 1321–1330.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*, 770–778.

He, S.; Feng, L.; Lv, F.; Li, W.; and Yang, G. 2022. Partial Label Learning with Semantic Label Representations. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 545–553.

Hinton, G.; Vinyals, O.; Dean, J.; et al. 2014. Distilling the knowledge in a neural network. *Multimedia Tools and Applications*, 80: 4037–4051.

Hüllermeier, E.; and Beringer, J. 2006. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10: 419–439.

Ji, Z.; Ni, J.; Liu, X.; and Pang, Y. 2023. Teachers cooperation: team-knowledge distillation for multiple cross-domain few-shot learning. *Frontiers of Computer Science*, 17: 172312.

Jia, B.-B.; and Zhang, M.-L. 2022. Multi-dimensional classification via selective feature augmentation. *Machine Intelligence Research*, 19: 38–51.

Jia, X.; Li, W.; Liu, J.; and Zhang, Y. 2018. Label distribution learning by exploiting label correlations. In *Association for the Advancement of Artificial Intelligence*, 3310–3317.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. 18661–18673.

Kim, K.; Ji, B.; Yoon, D.; and Hwang, S. 2021. Self-knowledge distillation with progressive refinement of targets. In *International Conference on Computer Vision*, 6567–6576.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Liu, L.; and Dietterich, T. G. 2012. A conditional multinomial mixture model for superset label learning. In *Advances in neural information processing systems*, 557–565.

Luo, J.; and Orabona, F. 2010. Learning from candidate labeling sets. In *Advances in neural information processing systems*, 1504–1512.

Lv, J.; Xu, M.; Feng, L.; Niu, G.; Geng, X.; and Sugiyama, M. 2020. Progressive identification of true labels for partial-label learning. In *International Conference on Machine Learning*, 6500–6510.

Lyu, G.; Wu, Y.; and Feng, S. 2022. Deep Graph Matching for Partial Label Learning. In *International Joint Conference on Artificial Intelligence*, 3306–3312.

Mobahi, H.; Farajtabar, M.; and Bartlett, P. 2020. Self-distillation amplifies regularization in hilbert space. In *Advances in neural information processing systems*, 3351–3361.

Nilsback, M.-E.; and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 722–729.

Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; and Jawahar, C. 2012. Cats and dogs. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*, 3498–3505.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.;

et al. 2019. Pytorch: An imperative style, high-performance deep learning library.

Qiao, C.; Xu, N.; and Geng, X. 2023. Decompositional Generation Process for Instance-Dependent Partial Label Learning. In *International Conference on Learning Representations*.

Shen, Y.; Xu, L.; Yang, Y.; Li, Y.; and Guo, Y. 2022. Self-Distillation from the Last Mini-Batch for Consistency Regularization. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*, 11943–11952.

Tejankar, A.; Koohpayegani, S. A.; Pillai, V.; Favaro, P.; and Pirsiavash, H. 2021. ISD: Self-supervised learning by iterative similarity distillation. In *International Conference on Computer Vision*, 9609–9618.

Tung, F.; and Mori, G. 2019. Similarity-preserving knowledge distillation. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*, 1365–1374.

Wang, D.-B.; Li, L.; and Zhang, M.-L. 2019. Adaptive graph guided disambiguation for partial label learning. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 83–91.

Wang, H.; Xia, M.; Li, Y.; Mao, Y.; Feng, L.; Chen, G.; and Zhao, J. 2022a. Solar: Sinkhorn label refinery for imbalanced partial-label learning. In *Advances in neural information processing systems*, 8104–8117.

Wang, H.; Xiao, R.; Li, Y.; Feng, L.; Niu, G.; Chen, G.; and Zhao, J. 2022b. PiCO: Contrastive Label Disambiguation for Partial Label Learning. In *International Conference on Learning Representations*.

Wang, W.; and Zhang, M.-L. 2020. Semi-supervised partial label learning via confidence-rated margin maximization. In *Advances in neural information processing systems*, 6982–6993.

Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; and Perona, P. 2010. Caltech-UCSD birds 200.

Wen, H.; Cui, J.; Hang, H.; Liu, J.; Wang, Y.; and Lin, Z. 2021. Leveraged weighted loss for partial label learning. In *International Conference on Machine Learning*, 11091–11100.

Wu, D.-D.; Wang, D.-B.; and Zhang, M.-L. 2022. Revisiting Consistency Regularization for Deep Partial Label Learning. In *International Conference on Machine Learning*, 24212–24225.

Xia, S.-Y.; Lv, J.; Xu, N.; and Geng, X. 2022. Ambiguity-Induced Contrastive Learning for Instance-Dependent Partial Label Learning. In *International Joint Conference on Artificial Intelligence*, 3615–3621.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xie, M.-K.; Sun, F.; and Huang, S.-J. 2021. Partial multi-label learning with meta disambiguation. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1904–1912.

Xu, N.; Liu, B.; Lv, J.; Qiao, C.; and Geng, X. 2023. Progressive Purification for Instance-Dependent Partial Label Learning. In *International Conference on Machine Learning*.

Xu, N.; Lv, J.; and Geng, X. 2019. Partial label learning via label enhancement. In *Association for the Advancement of Artificial Intelligence*, 5557–5564.

Xu, N.; Qiao, C.; Geng, X.; and Zhang, M.-L. 2021. Instance-dependent partial label learning. In *Advances in neural information processing systems*, 27119–27130.

Xu, T.-B.; and Liu, C.-L. 2019. Data-distortion guided self-distillation for deep neural networks. In *Association for the Advancement of Artificial Intelligence*, 5565–5572.

Xu, Y.-Y.; Shen, Y.; Wei, X.-S.; and Yang, J. 2022. Webly-Supervised Fine-Grained Recognition with Partial Label Learning. In *International Joint Conference on Artificial Intelligence*, 1502–1508.

Yang, W.; Li, C.; and Jiang, L. 2023. Learning from crowds with robust support vector machines. *Science China Information Sciences*, 66: 132103.

Yu, F.; and Zhang, M.-L. 2015. Maximum Margin Partial Label Learning. In *Asian Conference on Machine Learning*, 96–111.

Yuan, L.; Tay, F. E.; Li, G.; Wang, T.; and Feng, J. 2020. Revisiting knowledge distillation via label smoothing regularization. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*, 3903–3911.

Yun, S.; Park, J.; Lee, K.; and Shin, J. 2020. Regularizing class-wise predictions via self-knowledge distillation. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*, 13876–13885.

Zeng, Z.; Xiao, S.; Jia, K.; Chan, T.-H.; Gao, S.; Xu, D.; and Ma, Y. 2013. Learning by associating ambiguously labeled images. In *IEEE/CVF conference on Computer Vision and Pattern Recognition*, 708–715.

Zhang, F.; Feng, L.; Han, B.; Liu, T.; Niu, G.; Qin, T.; and Sugiyama, M. 2021. Exploiting Class Activation Value for Partial-Label Learning. In *International Conference on Learning Representations*.

Zhang, M.-L.; and Yu, F. 2015. Solving the partial label learning problem: An instance-based approach. In *International Joint Conference on Artificial Intelligence*, 4048–4054.

Zhou, Z.-H.; and Jiang, Y. 2004. NeC4.5: neural ensemble based C4. 5. *IEEE Transactions on knowledge and data engineering*, 16: 770–773.

Zhou, Z.-H.; Jiang, Y.; and Chen, S.-F. 2003. Extracting symbolic rules from trained neural network ensembles. *Ai Communications*, 16: 3–15.