Candidate-aware Selective Disambiguation Based On Normalized Entropy for Instance-dependent Partial-label Learning

Shuo He[1] Guowu Yang[1]* Lei Feng[2]
[1]University of Electronic Science and Technology of China
[2]Nanyang Technological University, Singapore
shuohe@std.uestc.edu.cn, guowu@uestc.edu.cn, lfengqaq@gmail.com

Abstract

In partial-label learning (PLL), each training example has a set of candidate labels, among which only one is the true label. Most existing PLL studies focus on the instance-independent (II) case, where the generation of candidate labels is only dependent on the true label. However, this II-PLL paradigm could be unrealistic, since candidate labels are usually generated according to the specific features of the instance. Therefore, instance-dependent PLL (ID-PLL) has attracted increasing attention recently. Unfortunately, existing ID-PLL studies lack an insightful perception of the intrinsic challenge in ID-PLL. In this paper, we start with an empirical study of the dynamics of label disambiguation in both II-PLL and ID-PLL. We found that the performance degradation of ID-PLL stems from the inaccurate supervision caused by massive under-disambiguated (UD) examples that do not achieve complete disambiguation. To solve this problem, we propose a novel two-stage PLL framework including selective disambiguation and candidate-aware thresholding. Specifically, we first choose a part of well-disambiguated (WD) examples based on the magnitude of normalized entropy (NE) and integrate harmless complementary supervision from the remaining ones to train two networks. Next, the remaining examples whose NE is lower than the specific class-wise WD-NE threshold are selected as additional WD ones. Meanwhile, the remaining UD examples, whose NE is lower than the self-adaptive UD-NE threshold and whose predictions from two networks are agreed, are also regarded as WD ones for model training. Extensive experiments demonstrate that our proposed method outperforms state-of-the-art PLL methods.

1. Introduction

The development of modern deep neural networks (DNNs) relies on a large amount of perfectly labeled data in real-world tasks. However, collecting such high-quality data in real-world scenarios is considerably time-consuming and laborious even for experienced labeling experts [28]. To alleviate this issue, researchers have paid much attention to partial-label learning (PLL), a weakly-supervised learning paradigm that is labeling-friendly for non-expert annotators, allowing to assign a candidate label set [6]. In particular, a majority of previous PLL works implicitly employ an *instance-independent* (II) assumption that the generation of each candidate label for an instance is of equiprobability and independent of the instance itself. But, this ideal assumption is unrealistic in real-world scenarios [22]. From the perspective of an image annotator, the probability of a domain class being selected as a candidate label is related to the visual perception information (i.e., semantic, color, and shape) of an image itself. This case is called *instance-dependent* (ID) [23] where the generation of candidate labels is according to the specific features of

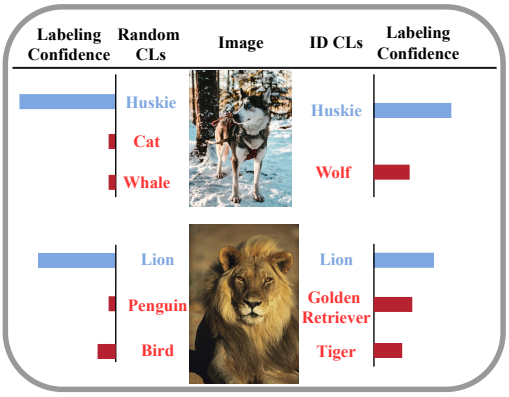

Figure 1. Illustration of the difference of labeling confidence after label disambiguation between II-PLL (left) and ID-PLL (right). We can see that the difficulty of identifying the true label (blue) of examples with various types of candidate labels (CLs) is prominently different. In addition, the image with sharp labeling confidence on random CLs is well-disambiguated, while the image with flat labeling confidence on ID CLs is under-disambiguated.

*Corresponding author.

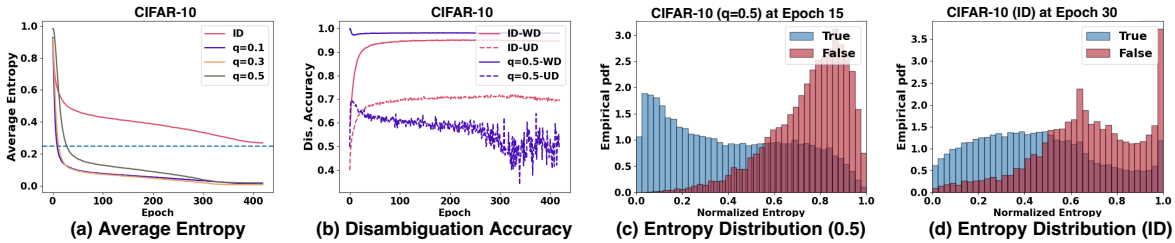

Figure 2. Dynamics of label disambiguation during the training on CIFAR-10 under ID and II cases. Figure (a) presents the average normalized entropy towards all training examples at each iteration, reflecting the whole status of label disambiguation. A lower entropy value implies that more partially labeled training examples are well-disambiguated; Figure (b) shows the accuracy of label disambiguation for WD and UD examples respectively; Figures (c) and (d) display the normalized entropy distribution of correctly disambiguated (blue) and falsely disambiguated (red) examples at the early epoch respectively.

the instance, and has attracted much attention in the PLL community recently.

To learn from the partially-labeled examples in II-PLL, self-training based label disambiguation (LD) has achieved great success, which progressively refines soft labeling confidences [6, 24, 19] at each iteration towards each partially-labeled example for the model training based on various means such as the network prediction [11, 19], class activation map (CAM) [24], class prototypes [16] or consistency regularization [19]. Unfortunately, these PLL methods show a serious degradation of performance in ID-PLL [22, 23]. To alleviate this issue, researchers have started to design specific ID-PLL algorithms recently [23, 13]. However, existing ID-PLL studies lack an insightful perception of intrinsic challenges in ID-PLL. In this paper, we start with an empirical study of the dynamics of LD both in II-PLL and ID-PLL. To achieve a thorough exploration, we first introduce a pivotal statistic to evaluate the status of LD: normalized entropy (NE) of labeling confidences (the detailed definition is referred to as Eq.(3)). The implication is that a low NE indicates a better status of LD. Based on this metric, we record the labeling confidence and corresponding NE towards all training examples at each iteration based on PRODEN [11] on CIFAR-10 under both II and ID cases. Based on the statistics, we can analyze the dynamics of LD from three various aspects: the whole status of LD, disambiguation accuracy, and NE distribution. The experiment results are shown in Figure 2. In Figure 2 (a), we calculate the average NE of labeling confidence towards all training examples at each iteration, reflecting the whole status of LD. We can see that the curves of LD on II-PLL ($q = [0.1, 0.3, 0.5]$) reach near zero, while the curve on ID-PLL is close to 0.22. This phenomenon means that most examples in II-PLL have been well-disambiguated (WD), while a mass of under-disambiguated (UD) examples on ID-PLL do not achieve complete disambiguation (i.e., maintaining a relative high NE). To further explore the characteristic of UD examples, we divide the whole training examples into well-disambiguated WD and UD ones based on a fixed threshold and calculate the accuracy of labeling confidence for them at each iteration respectively. The result is shown in Figure 2 (b). We can see that UD examples maintain a lower oscillating disambiguation accuracy than WD ones. On the other hand, we plot the empirical NE distribution of correctly identified (True in blue) and falsely identified (False in red) examples at epoch 15 on II ($q = 0.5$) and 30 on ID respectively. As shown in Figure 2 (c) and (d), we can see that both in II-PLL and ID-PLL, correctly identified examples have lower NE than falsely identified ones, and correctly identified examples in ID-PLL have more falsely identified low NE examples which is a knotty problem in the model training. Based on these empirical observations, we have revealed the intrinsic challenge in ID-PLL compared with II-PLL lies in massive UD examples with inaccurate supervision in the training procedure. Moreover, the under-disambiguation phenomenon also motivates us to think about how to determine whether a partially labeled training example has been disambiguated. In the above empirical study, we simply use a fixed threshold of NE to discriminate WD from UD ones. However, training examples with various candidate label sets have different intrinsic difficulties of LD, and the NE of training examples varies dynamically with the learning process of networks. Therefore, it may be intractable to assign an appropriate fixed threshold for all training examples [17].

To address these challenges, we propose a novel two-stage PLL framework including selective disambiguation and candidate-aware thresholding. Specifically, due to the negative role of the UD example in ID-PLL, we first choose a part of WD examples based on the magnitude of NE and meanwhile integrate harmless complementary supervision from the remaining ones to train two networks simultaneously. This selective disambiguation procedure is expected to learn two well-trained networks that are powerful enough to handle the remaining examples. Next, instead of using a fixed threshold of NE, we propose a dynamic candidate-aware thresholding scheme that respectively maintains a class-wise WD-NE threshold reflecting the different diffi-

culties of LD on each class and a self-adaptive UD-NE threshold indicating the whole status of LD on UD examples. Based on these two thresholds, we select the remaining examples whose NE is lower than the specific class-wise WD-NE threshold as additional WD ones to supplement the selected WD set in Stage 1. Meanwhile, we also select UD examples whose NE is lower than the UD-NE threshold and whose predictions from two networks are agreed, as WD ones for model training. In this way, we only leverage these selected WD examples for model training in Stage 2. Extensive experiments demonstrate that our proposed method outperforms state-of-the-art PLL methods. Our contribution is summarized as follows:

- We discover that the performance degradation of ID-PLL stems from the inaccurate supervision caused by a mass of UD examples.
- We propose a novel two-stage PLL framework including selective disambiguation and candidate-aware thresholding for UD examples.
- Empirically, extensive experiments show our proposed framework's superiority and effectiveness in both II-PLL and ID-PLL.

2. Related Work

In this section, we first brief conventional partial-label learning (PLL) methods with hand-crafted features and then introduce modern deep PLL methods in an end-to-end training scheme.

2.1. Conventional Partial-label Learning

Based on hand-crafted features, the core goal of conventional PLL methods is *label disambiguation* which identifies the concealed true label inside the candidate label set. For this purpose, there are two common frameworks based on *averaging* and *identification* strategy respectively. The former aims to discriminate candidate labels from non-candidate ones [2, 8]. This manner treats all candidate labels equally and thus degrades the classifier due to noisy labels in the candidate label set. To overcome this issue, the latter is devoted to identifying the true label by treating it as a latent variable [27]. For this purpose, they progressively refined the labeling confidence of each candidate label based on different meanings, such as maximum likelihood criterion [9], topological structure in the feature space [26, 22], graph matching selection [12].

The potential limitation of conventional PLL methods is that they rely on hand-crafted features, naive linear models, and low-efficiency optimization technologies, thus leading to the limited scalability of large-scale datasets.

2.2. Deep Partial-label Learning

To solve the mentioned weakness of conventional PLL methods, researchers pay much attention to training a deep neural network with partial-labeled data in an end-to-end manner referred to as deep PLL [19, 24, 14].

Instance-independent PLL. Most previous PLL works implicitly employed an instance-independent assumption about the uniform generation process of the candidate label set. Specifically, the first theoretically guaranteed algorithm was proposed by [6] with risk consistency and classifier consistency. Similarly, [11] proposed to progressively identify the true label with theoretical guarantees and achieved impressive performance on image classification benchmarks. Meanwhile, [18] proposed the leveraged weighted loss, a family of loss functions with risk consistency, that considered the trade-off between losses on candidate and non-candidate labels. Inspired by contrastive learning, [16] utilized it to establish well-separated class prototypes and thereby performed label disambiguation based on the distance between instances and these class prototypes. Recently, [19] utilized consistency regularization to achieve state-of-the-art performances in deep PLL.

Instance-dependent PLL. Recently, instance-dependent PLL (ID-PLL), a more realistic case considering the generation of the candidate label set related to the specific features of the instance, has attracted much attention in the community. The first work [23] aimed to recover the latent label distribution via a label enhancement process. [20] proposed to utilize contrastive learning by leveraging additional information from label ambiguity. Recent work [13] considered a decompositional generation process of candidate labels and explicitly modeled this process via decomposed probability distribution models. However, these studies lack an insightful perception of the intrinsic challenge in ID-PLL. Moreover, these PLL methods based on naive label disambiguation could suffer from under-disambiguation shown in the empirical study, thereby leading to serious performance degradation.

3. Problem Setup

In this section, we introduce the symbols and terminologies to define the problem of partial-label learning (PLL). Supposed we have a partially-labeled training dataset with n examples $\mathcal{D} = \{\boldsymbol{x}_i, Y_i\}_{i=1}^{n}$, each pair consists of an example $\boldsymbol{x}_i \in \mathcal{X}$ and a corresponding candidate label set $Y_i \subset \mathcal{Y}$ where $\mathcal{X}$ and $\mathcal{Y} = \{1, 2, ..., c\}$ denote the input space and the label space respectively. Note that different from conventional supervised learning, the ground-truth label $y_i \in \mathcal{Y}$ is concealed in the candidate label set.

The objective of deep PLL is to train a deep neural network $f(\cdot)$ based on these partially-labeled examples $\mathcal{D}$. Obviously, directly utilizing category cross-entropy loss cal-

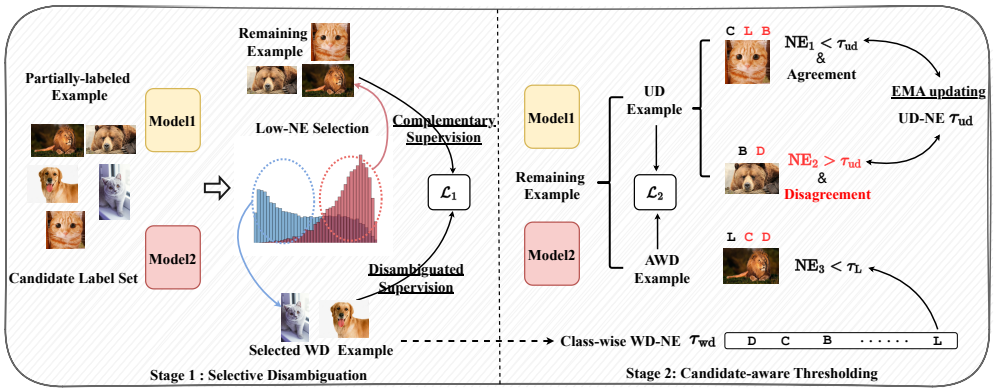

Figure 3. The pipeline of our proposed framework. The first stage is selective disambiguation which selects a part of WD examples based on the magnitude of normalized entropy (NE) and meanwhile integrates harmless complementary supervision from the remaining examples to learn well-trained models simultaneously. In the second stage, the remaining examples whose NE is lower than the specific class-wise WD-NE threshold (τ_{wd}) are additionally selected into the WD set. Meanwhile, the remaining UD examples, whose NE is lower than the self-adaptive UD-NE threshold τ_{ud} which is the average NE of UD examples updated by the exponential moving average (EMA), and whose predictions from two models are agreed, are also regarded as the WD ones for model training. We abbreviate "Cat", "Dog", "Lion", and "Bear" to C, D, L, and B respectively.

culated by multiple candidate labels could not train an effective network with convergence, due to the inherent *label ambiguity* in Y_i. Hence, the principle approach is *label disambiguation* that identifies true label y_i from the candidate label set Y_i. To achieve this aim, an example $\boldsymbol{x}_i$ is generally equipped with a labeling confidence vector $\boldsymbol{p}_i \in [0,1]^c$ (corresponding to a candidate label vector $\boldsymbol{y}_i$) where each entry denotes the probability of the corresponding candidate label being the ground-truth one. Based on this, The update of $\boldsymbol{p}_i$ implies the process of label disambiguation. Generally, a vanilla means of label disambiguation [11] is as follows:

$$\boldsymbol{p}_i = \mathrm{NM}(\mathrm{SM}(f(\boldsymbol{x}_i)) \cdot \boldsymbol{y}_i), \tag{1}$$

where $f(\boldsymbol{x}_i)$ is the model output, $\mathrm{NM}(\cdot)$ is a normalization operator, and $\mathrm{SM}(\cdot)$ is the softmax function. The implication of this equation is twofold. First, the labeling confidence of non-candidate labels maintains zero. Second, the labeling confidence of candidate labels is updating slightly and progressively based on the magnitude of model output. After this procedure, the model is trained by the cross-entropy loss with the updating $\boldsymbol{p}_i$:

$$\ell(\boldsymbol{x}_i, \boldsymbol{p}_i) = \sum_{j=1}^{c} -p_{ij} \log(f_j(\boldsymbol{x}_i)). \tag{2}$$

The above is a versatile PLL framework [6, 11, 19] that archives satisfying performance in instance-independent (II) PLL where the generated candidate label set Y is randomly sampled from the label space $\mathcal{Y}$. However, label ambiguity in instance-dependent (ID) PLL is more inherently challenging because more indistinguishable candidate labels co-occur with the true label in the candidate label set Y, thereby leading to a serious under-disambiguation phenomenon in ID-PLL.

4. Methodology

To address the challenge in instance-dependent partial label learning (ID-PLL), we propose a novel two-stage PLL framework including selective disambiguation and candidate-aware thresholding. An overview of the proposed framework is shown in Figure 3. Next, we introduce a key metric normalized entropy (NE) to evaluate the status of label disambiguation (LD).

4.1. Normalized Entropy

An intuitive measurement to evaluate the status of LD is the common metric entropy of labeling confidence $\boldsymbol{p}$. However, the entropy of labeling confidence on different numbers of candidate labels is not comparable due to the inherent property of entropy [7]. For example, suppose there are three examples with different numbers of candidate labels and corresponding labeling confidences: [0.8, 0.2], [0.8, 0.1, 0.1], [0.8, 0.05, 0.03, 0.02], the calculated entropy values are $0.72 < 0.92 < 1.02$ respectively. We can see that although these three examples hold a similar status of LD, their entropy values incrementally increase with the number of candidate labels. Actually, the training example with more candidate labels should be more difficult to achieve the same status of LD, and thereby should possess a relatively lower entropy. For this purpose, we employ *normalized entropy* [7] that considers the magnitude of the candidate label set to modulate the original entropy:

$$\mathrm{NE}(\boldsymbol{p}_i) = -\sum_j \frac{p_{ij}\log(p_{ij})}{\log(|Y_i|)}, \tag{3}$$

where $|Y_i|$ is the number of candidate labels of x_i. Based on this metric, the normalized entropy values of the above

three examples are $0.72 > 0.58 > 0.51$ respectively, which satisfies our consideration.

4.2. Selective Disambiguation

Based on the NE metric, we are able to effectively evaluate the status of LD towards all training examples. In the empirical study, we have found that the supervision of well-disambiguated (WD) examples is more accurate and stable than that of under-disambiguated (UD) ones. Therefore, we propose selective disambiguation that first selects a part of WD examples and meanwhile integrates harmless complementary supervision from the remaining ones to learn well-trained networks. Before the selection procedure, we first warm-up networks with the ordinal LD procedure in Eq.(1) for a certain epoch ϕ_1, making the network produce a more efficient NE distribution that is beneficial for the subsequent selection process.

Low-NE selection. Towards all training examples at each iteration, we globally select a part of low NE examples as WD ones.

$$\widetilde{\mathcal{D}}_{\mathrm{wd}} = \mathrm{argmin}_{\mathcal{D}':|\mathcal{D}'|\leqslant\gamma|\mathcal{D}|} \sum_{(\boldsymbol{x}_i,\boldsymbol{p}_i)\in\mathcal{D}'} \mathrm{NE}(\boldsymbol{p}_i), \quad (4)$$

where γ is a selection proportion parameter and $|\mathcal{D}|$ is the number of training data. Note that we execute the selection procedure in each iteration before training the networks. After the selection procedure, during a mini-batch training process, we can calculate the ordinal cross-entropy loss for the selected WD set $\widetilde{\mathcal{B}}_{\mathrm{wd}} \subset \widetilde{\mathcal{D}}_{\mathrm{wd}}$ in a training mini-batch:

$$\mathcal{L}_{\mathrm{wd}} = \frac{1}{|\widetilde{\mathcal{B}}_{\mathrm{wd}}|} \sum\nolimits_{\boldsymbol{x}_i\in\widetilde{\mathcal{B}}_{\mathrm{wd}}} \ell(\mathcal{W}(\boldsymbol{x}_i), \boldsymbol{p}_i), \quad (5)$$

where $|\widetilde{\mathcal{B}}_{\mathrm{wd}}|$ is the number of selected WD examples in a mini-batch and $\mathcal{W}(\cdot)$ is a weakly-augmented transform operator. Following the previous work [19], we also employ the consistency regularization technique for boosting performance:

$$\mathcal{L}_{\mathrm{cr}} = \frac{1}{|\widetilde{\mathcal{B}}_{\mathrm{wd}}|} \sum\nolimits_{\boldsymbol{x}_i\in\widetilde{\mathcal{B}}_{\mathrm{wd}}} \ell(\mathcal{S}(\boldsymbol{x}_i), \boldsymbol{p}_i), \quad (6)$$

where $\mathcal{S}(\cdot)$ is a strongly-augmented transform operator detailed in the experiment. Besides, we further incorporate the widely used technique of mix-up [25]. Concretely, we generate a virtual mixed training example $(\boldsymbol{x}^{'}, \widetilde{\boldsymbol{y}}^{'})$ by linearly interpolating the randomly sampled pair of WD examples $(\boldsymbol{x}_i, \widetilde{\boldsymbol{y}}_i)$ and $(\boldsymbol{x}_j, \widetilde{\boldsymbol{y}}_j)$ where $\widetilde{\boldsymbol{y}}_i$ is a one-hot vector corresponding to the predicted label $\widetilde{y}_i$ in $\boldsymbol{p}_i$. Based on this, we can obtain a virtual mix-up WD set $\widetilde{\mathcal{B}}^{'}_{\mathrm{wd}}$ and calculate the cross-entropy loss towards it:

$$\mathcal{L}_{\mathrm{m}} = \frac{1}{|\widetilde{\mathcal{B}}^{'}_{\mathrm{wd}}|} \sum\nolimits_{(\boldsymbol{x}^{'}_i,\widetilde{\boldsymbol{y}}^{'})\in\widetilde{\mathcal{B}}^{'}_{\mathrm{wd}}} \ell(\boldsymbol{x}^{'}, \widetilde{\boldsymbol{y}}^{'}). \quad (7)$$

Complementary supervision. Due to the inaccurate supervision of the remaining examples, it cannot directly leverage them for model training. Instead of discarding them for model training, motivated by the success of complementary-label learning [5, 19], we further utilize harmless complementary supervision from non-candidates of the remaining examples for model training:

$$\mathcal{L}_{\mathrm{cs}} = -\frac{1}{|\widetilde{\mathcal{B}}_r|} \sum_{i=1} \sum_{j=1}^{c} (1 - p_{ij}) \log(1 - f_j(\mathcal{W}(\boldsymbol{x}_i))), \quad (8)$$

where $\widetilde{\mathcal{B}}_r$ is the remaining non-selected examples in a mini-batch. The implication of Eq.(8) is that the log-likelihoods of predictions from non-candidates should be small since their labeling confidence maintains zeros. Finally, we can use the overall loss to update two networks respectively in Stage 1:

$$\mathcal{L}_1 = \mathcal{L}_{\mathrm{wd}} + \beta(\mathcal{L}_{\mathrm{cs}} + \mathcal{L}_{\mathrm{cr}} + \mathcal{L}_{\mathrm{m}}), \quad (9)$$

where β is a trade-off parameter and we ramp it up linearly [19]. This stage continues until a certain epoch ϕ_2 to learn well-trained networks that are expected to be powerful for dealing with remaining examples in the subsequent stage of candidate-aware thresholding.

4.3. Candidate-aware Thresholding

After the selective disambiguation procedure, the well-trained networks are expected to have the ability to handle the remaining examples, i.e., determining whether the remaining examples have been disambiguated and selecting qualified examples as WD ones for model training. For this purpose, instead of using a fixed threshold for all remaining examples, we propose a dynamic candidate-aware thresholding (CAT) scheme that respectively maintains a class-wise WD-NE threshold $\tau_{\mathrm{wd}} = [0, 1]^c$ reflecting the different difficulties of LD on each class c and a self-adaptive UD-NE threshold τ_{ud} indicating the whole LD status on UD examples. Actually, we maintain two different thresholds for two models respectively. For convenience, we consider a single model case in the next. Formally, the class-wise WD-NE threshold τ_{wd} is calculated by:

$$\tau^c_{\mathrm{wd}} = \frac{1}{|\widetilde{\mathcal{D}}^c_{\mathrm{wd}}|} \sum\nolimits_{\boldsymbol{x}_i\in\widetilde{\mathcal{D}}^c_{\mathrm{wd}}} \mathrm{NE}(\boldsymbol{p}_i), \quad (10)$$

where $\widetilde{\mathcal{D}}^c_{\mathrm{wd}} \subset \widetilde{\mathcal{D}}_{\mathrm{wd}} = \{(\boldsymbol{x}_i, c) | c = \mathrm{argmax}(\boldsymbol{p}_i)\}$ is a class-wise subset whose predicted label is c. Note that τ_{wd} is

calculated in a global fashion using the selected WD subset $\widetilde{\mathcal{D}}_{\mathrm{wd}}$ in the selective disambiguation procedure. In a mini-batch training process, we additionally select the remaining examples $\widetilde{\mathcal{B}}_{\mathrm{awd}}$ whose NE is lower than the specific class-wise WD-NE threshold as WD ones:

$$\mathrm{NE}(\boldsymbol{p}_i) < \tau_{\mathrm{wd}}^{c}, \tag{11}$$

where $c = \mathrm{argmax}(\boldsymbol{p}_i)$ is the predicted label in $\widetilde{\mathcal{B}}_{\mathrm{r}}$. The potential advantage of additionally selecting WD examples is to combat the class-imbalanced selection bias in the first stage [15]. This is because the class-wise WD-NE threshold would possess a larger (lower) value for a hard (easy) class, thereby producing more (less) additional WD examples selected from a hard (easy) class. After this process, the remaining examples are regarded as UD ones $\widetilde{\mathcal{B}}_{\mathrm{ud}} = \widetilde{\mathcal{B}}_{\mathrm{r}} - \widetilde{\mathcal{B}}_{\mathrm{awd}}$. Formally, we also maintain a self-adaptive UD-NE threshold τ_{ud} by the exponential moving average (EMA):

$$\tau_{\mathrm{ud}}^{(t)} = \lambda \tau_{\mathrm{ud}}^{(t-1)} + (1-\lambda)\frac{1}{|\widetilde{\mathcal{B}}_{\mathrm{ud}}|}\sum\nolimits_{\boldsymbol{x}_i \in \widetilde{\mathcal{B}}_{\mathrm{ud}}} \mathrm{NE}(\boldsymbol{p}_i), \tag{12}$$

where $\tau_{\mathrm{ud}}^{(t)}$ is the UD-NE threshold at the t-th step and $\lambda \in (0,1)$ is the momentum decay of EMA. The initial value of τ_{ud} is set to 0. Based on this, we hold that the UD examples whose NE is lower than the UD-NE threshold have been disambiguated. Due to the inaccurate supervision of UD examples, we additionally employ the agreement condition of two networks to reduce the selection error:

$$(\mathrm{NE}(\boldsymbol{p}_i) < \tau_{\mathrm{ud}}) \& (c^1 == c^2), \tag{13}$$

where $c^i (i = \{1,2\})$ is the predicted label of the i-th network. In Stage 2, we further leverage the additionally selected WD examples for model training:

$$\mathcal{L}_2 = \mathcal{L}_{\mathrm{wd}} + \beta(\mathcal{L}_{\mathrm{cr}} + \mathcal{L}_{\mathrm{m}}). \tag{14}$$

The pseudo-code of the proposed method is in Algorithm 1.

Algorithm 1: Our Framework

Input: Partially-labeled dataset $\mathcal{D}$, model1 $f_1(\cdot)$, model2 $f_2(\cdot)$ parameters: loss parameter β, selection proportion γ, the epoch ϕ_1, ϕ_2, and $T_{\max}$, weak and strong data augmentation $\mathcal{W}(\cdot)$ and $\mathcal{S}(\cdot)$, $\tau_{\mathrm{ud}} = 0$, $\lambda = 0.99$.

Output: model1 and model2

```
for t < T_max do
    for t < φ1 do
        Warm-up f1(·) and f2(·) by Eq.(1);
    end
    for t >= φ1 do
        Select WD examples D̃_wd by Eq.(4)
          respectively for each model;
        Calculate the WD-NE threshold τ_wd by
          Eq.(10);
        if t >= φ2 then
            Select addition WD examples B̃_awd by
              Eq.(11);
            Update the UD-NE threshold τ_ud by
              Eq.(12);
            Select qualified UD examples B̃_ud by
              Eq.(13);
        end
        Calculate the overall loss L1 Eq.(9) or L2
          Eq.(14) to update two models;
        Update labeling confidence by Eq.(1);
    end
end
```

5. Experiment

In this section, we conduct extensive experiments with benchmark datasets: Kuzushiji-MNIST [1], Fashion-MNIST [21], CIFAR-10 and CIFAR-100 [10] to demonstrate the superiority and effectiveness of our proposed method. We first introduce the generation of instance-dependent (ID) and instance-independent (II) candidate labels and then show compared PLL methods and the detail of implementation. Finally, we present experiment results including test accuracy comparison, ablation study, and hyper-parameter analysis.

5.1. Setup

Datasets. On four benchmark datasets, to generate instance-dependent (ID) candidate labels, we follow the work [23] where the probability of each domain class being selected as a candidate label is related to each instance itself by utilizing the prediction of a neural network trained with original clean labels. On the other hand, we generate instance-independent (II) candidate labels by a uniform flipping probability $q = [0.2, 0.4, 0.6]$ where q implies a higher degree of label ambiguity [6]. Particularly, we generate candidate labels that belong to the same super-class on CIFAR-100 with hierarchical labels denoted by CIFAR-100-H ($q = 0.5$).

Compared methods. Two ID-PLL methods are compared: IDGP [13] that assumes that the generation process of the candidate labels could decompose into two sequential parts, and ABLE [20], a contrastive learning method that utilizes the label ambiguity information. Moreover, we compare five II-PLL algorithms: (1) CC [6], a classifier-consistent method that assumes a set-level uniform data generation process; (2) PRODEN [11], a self-training-based

Table 1. Test Accuracy comparisons on Fashion-MNIST, Kuzushiji-MNIST, CIFAR-10 and CIFAR-100 with instance-dependent partial labels. The best result is highlighted.

Datasets	CC	PRODEN	LWS	PiCO	CRDPLL	ABLE	IDGP	Ours
K-MNIST	92.65±.12	96.18±.12	97.88±.26	96.27±.14	98.23±.10	98.15±.10	97.85±.25	**98.35±.12**
F-MNIST	86.23±.15	88.32±.10	87.65±.13	88.15±.06	89.15±.08	90.14±.15	88.98±.39	**92.81±.18**
CIFAR-10	79.37±.28	86.54±.16	89.25±.19	91.15±.18	87.29±.25	92.25±.18	85.62±.21	**95.04±.15**
CIFAR-100	61.45±.33	64.27±.20	71.14±.25	70.88±.12	77.82±.12	74.36±.29	66.88±.37	**78.67±.22**

Table 2. Test Accuracy comparisons on Fashion-MNIST, Kuzushiji-MNIST, CIFAR-10 and CIFAR-100 with uniform partial labels on different levels of label disambiguity q. The best result is highlighted. in bold.

Datasets	q	CC	PRODEN	LWS	PiCO	CRDPLL	ABLE	IDGP	Ours
K-MNIST	0.2	96.12±.18	97.65±.24	98.12±.15	97.86±.09	98.66±.13	98.21±.15	98.09±.14	**98.72±.15**
	0.4	95.64±.15	96.67±.10	97.42±.12	97.39±.12	98.15±.05	97.86±.20	97.78±.33	**98.58±.18**
	0.6	94.25±.24	95.43±.21	96.33±.27	97.12±.18	97.74±.12	97.29±.13	97.10±.22	**98.24±.22**
F-MNIST	0.2	92.23±.26	93.42±.15	94.41±.10	94.62±.22	94.53±.20	94.33±.23	94.12±.11	**95.02±.16**
	0.4	91.89±.12	92.56±.13	93.90±.15	94.13±.17	94.22±.09	94.10±.15	93.78±.26	**94.85±.10**
	0.6	90.15±.17	91.26±.22	92.72±.23	93.24±.12	93.76±.13	93.02±.14	93.14±.18	**94.62±.25**
CIFAR-10	0.2	87.62±.15	92.35±.14	93.85±.13	94.53±.17	96.56±.31	95.24±.26	93.34±.15	**96.58±.12**
	0.4	85.88±.12	89.68±.12	92.54±.15	94.15±.15	96.23±.28	94.71±.20	92.85±.16	**96.35±.15**
	0.6	82.57±.23	86.42±.28	90.38±.21	92.95±.10	95.89±.17	93.15±.13	90.15±.24	**96.17±.26**
CIFAR-100	H-0.5	62.24±.10	66.17±.20	74.53±.14	72.37±.24	78.01±.13	73.41±.13	67.85±.26	**80.25±.14**
	0.05	63.02±.17	65.21±.28	74.18±.21	72.18±.26	78.89±.15	73.17±.20	68.39±.17	**81.26±.18**
	0.1	61.63±.18	64.15±.16	72.64±.15	60.12±.20	78.12±.20	62.83±.12	58.22±.14	**80.34±.20**

method that progressively identifies the true labels using the output of the classier itself; (3) LWS [18], a set of loss functions that weights the risk function by means of a trade-off between losses on candidates and non-candidates; (4) PiCO [16], a contrastive learning-based method that establishes class prototypes for label disambiguation; (5) CRDPLL [19], a regularization based method that achieves state-of-the-art performance in instance-independent PLL.

Implementation. We employ a basic training scheme: a ResNet-18 backbone, a standard SGD optimizer with a learning rate of 0.05, a momentum of 0.9, a weight decay of 5e-4, and cosine learning rate schedule with a decay rate of 0.1, and a total training epoch 450, the batch size is 256. For a fair comparison, we reimplement CC, PRODEN, LWS, and CRDPLL (the batch size is 64) using the same training scheme. In particular, CC, PRODEN, and LWS are equipped with weak transform operators *random cropping* and *horizontal flipping*, while CRDPLL is additionally equipped with a strong transform operator *RandAugment* [3] on CIFAR-10 and CIFAR100 (CutOut [4] on Fashion-MNIST, Kuzushiji-MNIST). These weak and strong transform operators are also used in our method as $\mathcal{W}(\cdot)$ and $\mathcal{S}(\cdot)$ respectively. For PiCO, ABLE, and IDGP, we follow their original training schemes. For the parameters in our method, the epoch ϕ_1 (ϕ_2) is set to 60 (300), the selection proportion γ is set to 0.6, the momentum λ is set to 0.99, β is ramped up to 1 linearly with the ramp-up epoch 100, and the shape parameter of beta distribution used in mix-up is set to 4. We present the mean and standard deviation in each case based on three trials.

5.2. Experimental Results

Test accuracy comparison. As shown in Table 1, our method significantly outperforms all counterparts on the ID case. This definitely validates the superiority of the proposed method to combat ID candidate labels. Moreover, we also show the comparison results on the II case with different levels of label ambiguity $q = [0.2, 0.4, 0.6]$ in Table 2. We can see that our method also achieves superior results against state-of-the-art PLL methods. From the comparison of these two tables, we can see that existing PLL methods have a prominent degradation of performance in the ID case compared with the II case. This observation justifies our empirical study in Figure 2 and implies the increased difficulty of label disambiguation against ID candidate labels. Fortunately, our proposed method maintains superior performance in these two cases.

Dynamic evaluation. We evaluate the dynamics of our method on three aspects: the accuracy (precision) of disambiguated labeling confidence ($\boldsymbol{p}_i$), selective selection ($\widetilde{\mathcal{D}}_{\mathrm{wd}}$), and candidate-aware thresholding ($\widetilde{\mathcal{B}}_{\mathrm{awd}}$ and $\widetilde{\mathcal{B}}_{\mathrm{ud}}$). As shown in Figure 4, the accuracy of disambiguated labeling confidence reaches a high level on different datasets, e.g., near 99% on CIFAR-10 (0.2) and 98% on F-MNIST

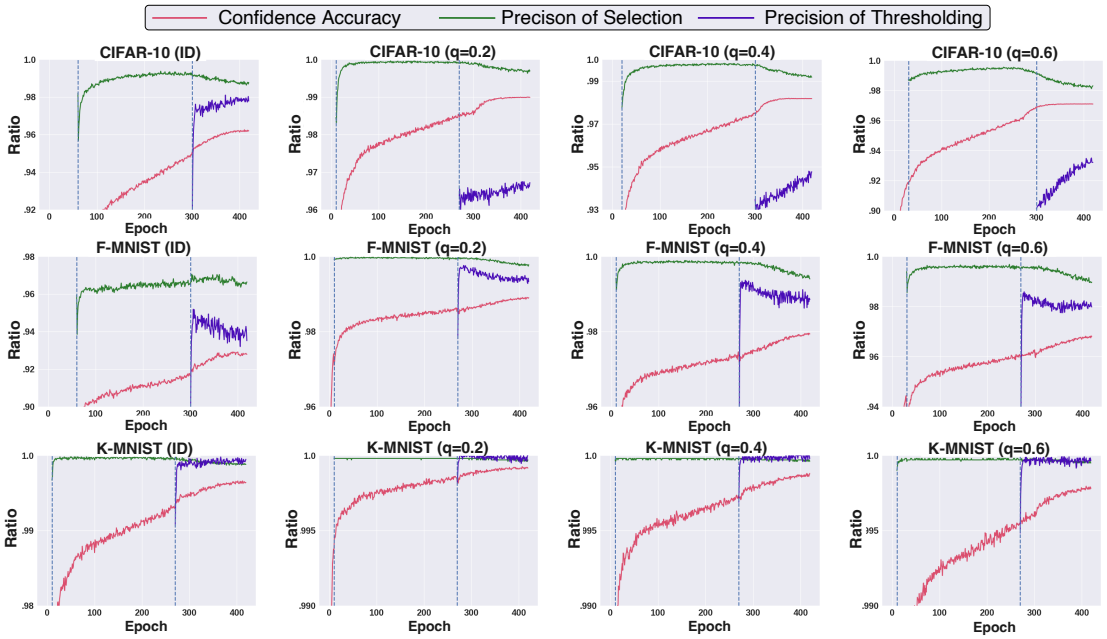

Figure 4. Curves of the accuracy of disambiguated labeling confidence (red), the precision of selective selection (green), and precision of candidate-aware thresholding (blue). The vertical dotted line divides the different stages during the training.

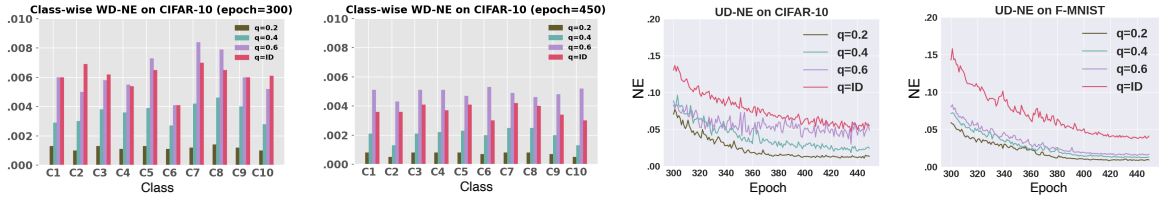

Figure 5. Dynamics of candidate-aware thresholding during the training. The first two sub-figures present class-wise WD-NE at the beginning training of epoch 300 (ending of epoch 450). The last two sub-figures present curves of UD-NE at each iteration.

(0.2-0.6). This definitely verifies the effectiveness of label disambiguation in the proposed method. On the other hand, the precision of selection maintains a high level in all cases except F-MNIST (ID). This procedure ensures that correctly selected WD examples can induce a well-trained network that is beneficial for the subsequent stage. Moreover, the precision of candidate-aware thresholding is kept at a high level in the ID case so that correctly identified examples can be leveraged for model training.

Analysis of candidate-aware thresholding. As shown in Figure 5, the first two sub-figures show class-wise WD-NE at the beginning training of epoch 300 (ending of epoch 450), and the last two sub-figures present curves of UD-NE at each iteration (from epoch 300 to 450). From the first two sub-figures, we can see that under the low level of label ambiguity ($q = 0.2$), WD-NE is well class-balanced at the beginning and end, while under the high level of label ambiguity ($q = 0.4, 0.6$, and ID), WD-NE becomes relatively class-imbalanced at the beginning. This phenomenon implies high label ambiguity degrades selective selection, resulting in a class imbalance in the selected WD set. Fortunately, by virtue of class-wise WD-NE thresholding, the selected WD set becomes much more class-balanced at the end of training. On the other hand, class-wise WD-NE is constantly decreased during the training, which adapts to the learning process of networks. From the last two sub-figures, we can see that UD-NE is adaptively updated at each iteration according to the learning status of UD examples. The overall trend of UD-NE is also to decrease like WD-NE but maintain a higher value reflecting the under-disambiguation status. In particular, as the label ambiguity increases, UD-NE also maintains a larger value. This implies that high label ambiguity also affects UD-NE, leading to a larger threshold for UD examples.

5.3. Ablation Studies

In this section, we present our ablation studies shown in Table 3. The item "w/o consistency regularization $\mathcal{L}_{\text{cr}}$" or "w/o mix-up augmentation $\mathcal{L}_{\text{m}}$" means that we do not use the data augmentation technique by directly discarding the

Table 3. Ablation studies on CIFAR-10 (ID).

Ablation	Accuracy
w/o consistency regularization $\mathcal{L}_{\mathrm{cr}}$	94.35±.23
w/o mix-up augmentation $\mathcal{L}_{\mathrm{m}}$	94.46±.18
w/o complementary supervision $\mathcal{L}_{\mathrm{cs}}$	95.20±.12
with fixed threshold τ_{wd} and τ_{ud}	94.58±.18
w/o UD examples in Stage 2	93.89±.24
with only WD examples in Stage 1	94.12±.20
Ours	**95.62±.15**

loss item $\mathcal{L}_{\mathrm{cr}}$ or $\mathcal{L}_{\mathrm{m}}$ in Eq.(9) and Eq.(14). The degradation of performance implies the significance of data augmentation in the proposed method. The item "w/o complementary supervision $\mathcal{L}_{\mathrm{cs}}$" means that we do not use complementary supervision from the remaining examples by directly discarding the loss item $\mathcal{L}_{\mathrm{cs}}$ in Eq.(9). Although the final accuracy drops sightly, the accuracy at the end of Stage 1 has a serious degradation, i.e., from 93.25% to 91.52%. The last two items explore the effect of selected WD and UD examples for model training in different stages. The item "with fixed threshold τ_{wd} and τ_{ud}" means that we use a fixed threshold $\tau_{\mathrm{wd}} = 1e-3$ in Eq.(11) and $\tau_{\mathrm{ud}} = 1e-2$ in Eq.(13). This results in fewer examples being selected as WD ones in Stage 2 thereby degrading the performance. The item "w/o UD examples in Stage 2" means that we totally discard selected qualified UD examples in candidate-aware thresholding. This would reduce the number of selected training examples, thereby losing useful information for generalization and obtaining the degraded performance. The item "with only WD examples in Stage 1" means that we only leverage selected WD examples in Stage 1 and do not add additional WD ones from Stage 2 for model training. This also leads to a serious reduction in selected training examples and a sharp degradation in performance.

5.4. Hyper-parameter Analysis

Here, we evaluate the main hyper-parameters in our method, including selection proportion γ, the epoch ϕ_1 and ϕ_2, and β. As shown in Figure 3, with a small value of γ, i.e., a few selected WD examples, the performance drops constantly. This is because too few selected WD examples in Stage 1 cannot learn well-trained networks. Otherwise, with a large γ, too many wrongly identified examples in the selected WD set degrade the performance seriously. For the epoch of warm-up ϕ_1, a small value leads to serious degradation of performance. This is because the network can not produce an efficient normalized entropy distribution for selective disambiguation, thereby reducing the precision of the selection procedure. Therefore, sufficient warming up of networks is significant in the proposed method. On the contrary, warming up too long could overfit wrongly identified candidate labels, thereby degrading the performance.

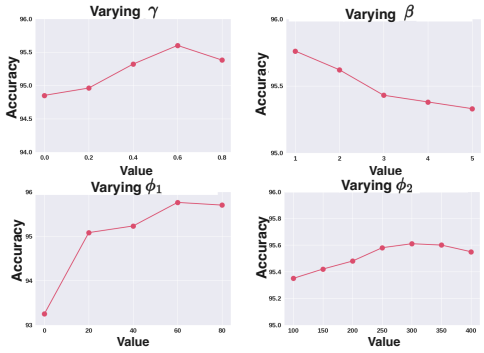

Figure 6. Varying parameters on CIFAR-10 (ID).

For the epoch of training ϕ_2, a large value i.e., the long training of Stage 1 slightly improves the performance. For β, the performance drops sharply as β increases. This is because paying much attention to data augmentation and complementary supervision affects the gradient updating of networks.

6. Conclusion

In this paper, we deal with the problem of instance-dependent partial-label learning (ID-PLL). Specially, we start with an insightful empirical study against the dynamics of label disambiguation during the training process, and discover that the performance degradation of ID-PLL stems from the inaccurate supervision caused by massive under-disambiguated (UD) examples. To address this challenge, we propose a novel two-stage PLL framework including *selective selection* and *candidate-aware thresholding*. The former first selects a part of WD examples and integrates complementary supervision of the remaining ones for model training. The latter maintains two dynamic and self-adaptive thresholds for WD and UD examples respectively, and selects additional WD and qualified UD ones for model training. Extensive experiments verify the superiority and effectiveness of the proposed method. In the future, it is interesting to develop other advanced methods to detect and handle UD examples effectively.

7. Acknowledgement

Lei Feng is supported by the National Natural Science Foundation of China (Grant No. 62106028), Chongqing Overseas Chinese Entrepreneurship and Innovation Support Program, CAAI-Huawei MindSpore Open Fund, and Chongqing Artificial Intelligence Innovation Center. Guowu Yang is supported by the National Natural Science Foundation of China (Grant No. 62172075). The authors also wish to thank the anonymous reviewers for their helpful and valuable comments.

References

[1] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018. 6

[2] Timothee Cour, Ben Sapp, and Ben Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12(5):1501–1536, 2011. 3

[3] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 7

[4] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 7

[5] Lei Feng, Takuo Kaneko, Bo Han, Gang Niu, Bo An, and Masashi Sugiyama. Learning with multiple complementary labels. In *Proceedings of the International Conference on Machine Learning*, pages 3072–3081. PMLR, 2020. 5

[6] Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. In *Advances in Neural Information Processing Systems*, 2020. 1, 2, 3, 4, 6

[7] Babak Hassibi and Sormeh Shadbakht. Normalized entropy vectors, network information theory and convex optimization. In *IEEE Information Theory Workshop on Information Theory for Wireless Networks*, pages 1–5. IEEE, 2007. 4

[8] Eyke Hüllermeier and Jürgen Beringer. Learning from ambiguously labeled examples. *Intell. Data Anal.*, 10(5):419–439, 2006. 3

[9] Rong Jin and Zoubin Ghahramani. Learning with multiple labels. volume 15, 2002. 3

[10] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[11] Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *Proceedings of the International Conference on Machine Learning*, pages 6500–6510, 2020. 2, 3, 4, 6

[12] Gengyu Lyu, Songhe Feng, Tao Wang, Congyan Lang, and Yidong Li. Gm-pll: graph matching based partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, 2019. 3

[13] Congyu Qiao, Ning Xu, and Xin Geng. Decomposition-based generation process for instance-dependent partial label learning. 2023. 2, 3, 6

[14] Haobo Wang, Mingxuan Xia, Yixuan Li, Yuren Mao, Lei Feng, Gang Chen, and Junbo Zhao. Solar: Sinkhorn label refinery for imbalanced partial-label learning. *Advances in neural information processing systems*, 2022. 3

[15] Haobo Wang, Ruixuan Xiao, Yiwen Dong, Lei Feng, and Junbo Zhao. Promix: combating label noise via maximizing clean sample utility. *arXiv preprint arXiv:2207.10276*, 2022. 6

[16] Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. Pico: Contrastive label disambiguation for partial label learning. In *Proceedings of the International Conference on Learning Representations*, 2022. 2, 3, 7

[17] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, et al. Freematch: Self-adaptive thresholding for semi-supervised learning. In *Proceedings of the International Conference on Learning Representations*, 2023. 2

[18] Hongwei Wen, Jingyi Cui, Hanyuan Hang, Jiabin Liu, Yisen Wang, and Zhouchen Lin. Leveraged weighted loss for partial label learning. In *Proceedings of the International Conference on Machine Learning*, 2021. 3, 7

[19] Dong-Dong Wu, Deng-Bao Wang, and Min-Ling Zhang. Revisiting consistency regularization for deep partial label learning. In *Proceedings of the International Conference on Machine Learning*, pages 24212–24225. PMLR, 2022. 2, 3, 4, 5, 7

[20] S Xia, Jiaqi Lv, Ning Xu, and Xin Geng. Ambiguity-induced contrastive learning for instance-dependent partial label learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 3615–3621, 2022. 3, 6

[21] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6

[22] Ning Xu, Jiaqi Lv, and Xin Geng. Partial label learning via label enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5557–5564, 2019. 1, 2, 3

[23] Ning Xu, Congyu Qiao, Xin Geng, and Min-Ling Zhang. Instance-dependent partial label learning. volume 34, 2021. 1, 2, 3, 6

[24] Fei Zhang, Lei Feng, Bo Han, Tongliang Liu, Gang Niu, Tao Qin, and Masashi Sugiyama. Exploiting class activation value for partial-label learning. In *Proceedings of the International Conference on Learning Representations*, 2021. 2, 3

[25] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of the International Conference on Learning Representations*, 2018. 5

[26] Min-Ling Zhang, Bin-Bin Zhou, and Xu-Ying Liu. Partial label learning via feature-aware disambiguation. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1335–1344, 2016. 3

[27] Yu Zhou, Jianjun He, and Hong Gu. Partial label learning via gaussian processes. *IEEE Transactions on Cybernetics*, 47(12):4443–4450, 2016. 3

[28] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018. 1