Neighbor-aware Label Refinement: Enhancing Unreliable Instance-Dependent Partial Labels

Xijia Tang[1], Yuhua Qian[2], Chao Xu[1*], Chenping Hou[1*]
[1]College of Science, National University of Defense Technology
[2]Institute of Big Data Science and Industry, Shanxi University
No. 109 Deya Road
Changsha, Hunan, China
TXJnudt@hotmail.com, jinchengqyh@sxu.edu.cn, xcnudt@hotmail.com, hcpnudt@hotmail.com

Abstract

Partial Label Learning (PLL) aims to train multi-class classifiers from examples where each instance is associated with a set of candidate labels, among which the ground-truth label is assumed to be included. While most existing studies assume that partial labels are both instance-independent and reliable, such assumptions often break down in real-world scenarios, where candidate sets may depend on instance-specific features and even exclude the ground-truth label. In this work, we investigate a more realistic setting termed Unreliable Instance-Dependent Partial Label Learning (UIDPLL). To address the challenges in UIDPLL, we propose a novel framework named Neighborhood-guided Label Augmentation and Pruning (NLAP). NLAP exploits the structural consistency among neighboring instances to progressively refine candidate label sets and integrates classifier feedback to disambiguate labels during training. This progressive mechanism improves classification performance by tackling ambiguity caused by noise and instance dependency in partial labels. Furthermore, we provide theoretical guarantees for the proposed NLAP framework, demonstrating that label ambiguity can be effectively reduced through appropriate refinement and pruning procedures. Extensive experiments on both benchmark and real-world datasets demonstrate the robustness and effectiveness of the proposed method.

Code — https://github.com/TangXJ-895/NLAP

Introduction

Partial label learning (PLL) (Tian, Yu, and Fu 2023) is a weakly supervised learning framework(Simmler et al. 2021b) where each training instance is annotated with a candidate label set containing the ground-truth label. This learning paradigm naturally arises in a variety of real-world scenarios where obtaining precise annotations is costly or impractical, such as web mining(Luo and Orabona 2010), multimedia content analysis(Zeng et al. 2013) and ecoinformatics(Tang and Zhang 2017b), etc(Vahedi et al. 2024; Francis 2024). In recent years, PLL has received increasing attention due to its practical significance and the unique challenge of learning with ambiguous supervision(Xu et al. 2024; Jia, Si, and Zhang 2023; Lv et al. 2023; Wang et al. 2023, 2025).

To address the ambiguity in candidate labels, a variety of algorithms have been developed, ranging from traditional methods to deep learning-based models. Among them, identification-based approaches(Feng and An 2019; Xu, Lv, and Geng 2019; Lv et al. 2020a) aim to infer the true label from the candidate set, while average-based approaches(Zhang and Yu 2015; Zhang, Yu, and Tang 2017; Lv et al. 2023) treat all candidate labels equally and perform prediction averaging. With the development of deep neural networks, advanced tools such as contrastive learning(Wang et al. 2022, 2023), knowledge distillation(Wu, Wang, and Zhang 2024), etc have been introduced into PLL, bringing new perspectives and significantly enhancing performance.

However, most existing methods hinge on two unrealistic assumptions: (i) the ground-truth label is always included in the candidate label set, and (ii) candidate labels are generated via a uniform, instance-independent process. In real-world scenarios, both assumptions are frequently violated. Since the true label is unknown during annotation, its inclusion in the candidate set cannot be guaranteed. Moreover, incorrect labels are often semantically or statistically related to the instance, making the generation process inherently instance-dependent. As illustrated in Fig.1, an image of a goose may be mislabeled as "Swan", " Stork" or "Duck" with the correct label "Goose" potentially omitted due to limitations of annotators or automated labeling tools. In contrast, the conventional PLL setting shown in the second row of Fig.1 assumes that the ground-truth label "Goose" is always present and that candidate labels are generated in an instance-independent manner, often producing semantically irrelevant labels such as "Squirrel". While these assumptions simplify algorithm design, they diverge from practical annotation dynamics and thus limit the applicability of conventional PLL methods in real-world scenarios.

In this work, we investigate a more realistic setting named Unreliable and Instance-Dependent Partial Label Learning (UIDPLL), where the candidate label set is instance-dependent and the ground-truth label is not guaranteed to be included. To address the UIDPLL challenge, we propose a simple yet effective method, Neighborhood-guided Label Augmentation and Pruning (NLAP), which lever-

*Chenping Hou and Chao Xu are the corresponding authors of this paper.

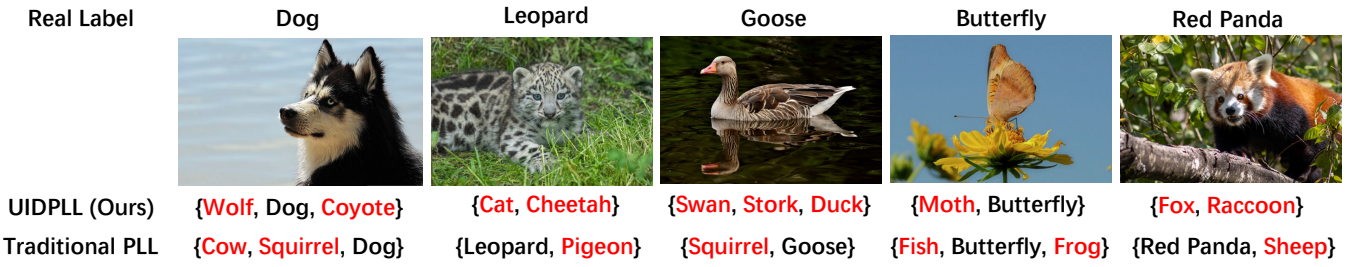

Figure 1: Illustration of the UIDPLL setting. The first row "Real Label" shows the ground-truth labels. The second row "UIDPLL" illustrates our setting, where candidate label sets may exclude the ground-truth and are biased toward semantically similar incorrect labels. The third row "Traditional PLL" represents the conventional PLL assumption that the ground-truth is always included and candidate labels are randomly sampled, independent of the instance.

ages neighborhood information to refine unreliable candidate sets. This neighbor-based refinement enhances label quality and supports more accurate disambiguation. NLAP iteratively improves candidate labels, enabling the model to handle both label absence and selection bias through a refinement-disambiguation loop. Furthermore, we provide theoretical guarantees for NLAP, showing that appropriate refinement and pruning can effectively reduce label ambiguity. Our main contributions can be summarized as follows:

- To the best of our knowledge, we are the first to study the UIDPLL setting, which is highly representative of real-world challenges.
- We propose NLAP, a neighbor-enhanced label refinement algorithm that corrects unreliable candidate sets and enables accurate disambiguation under UIDPLL.
- We theoretically analyze the effectiveness of NLAP and conduct extensive experiments on both benchmark and real-world PLL datasets, demonstrating its superior or comparable performance to existing SOTA approaches.

Related Work

In this section, we will provide a comprehensive overview of existing research on PLL. We categorize the literature into two major directions: traditional PLL approaches and deep PLL methods, both of which have been extensively studied.

Traditional PLL research primarily relied on linear models to address label ambiguity. These methods can be broadly categorized into two groups: averaging-based and identification-based disambiguation strategies. Averaging-based approaches(Hüllermeier and Beringer 2006; Zhang and Yu 2015; Zhang, Yu, and Tang 2017) treat all candidate labels equally and make predictions by averaging the model outputs over the candidate label set. In contrast, identification-based approaches(Chen et al. 2014; Zhang, Zhou, and Liu 2016; Tang and Zhang 2017a; Feng and An 2019; Xu, Lv, and Geng 2019) aim to heuristically identify the ground-truth label during training. By progressively refining the candidate label set, these methods seek to reduce label noise and improve label precision.

Deep neural networks have recently attracted widespread attention due to their strong representation power and flexibility. However, their training is highly sensitive to inaccurate or adversarial samples, which can degrade performance(Liu et al. 2024; Simmler et al. 2021a). PLL, as a classic weakly supervised paradigm, introduces label ambiguity that can similarly affect deep network training. Consequently, deep PLL has gained increasing interest in recent research, with numerous methods proposed to address these challenges. Lv et al. (Lv et al. 2020a) introduced PRODEN, a progressive identification approach that jointly addresses label disambiguation and classifier optimization by approximating the minimization of a consistency-based risk objective. (Wang et al. 2022) proposed an algorithm combining supervised contrastive learning with prototype-based label disambiguation to produce tightly clustered class representations and enhance disambiguation. Wu et al.(Wu, Wang, and Zhang 2024) proposed a distillation-based PLL framework combining confidence rectification and contrastive refinement to effectively enhance performance. Zhang et al.(Zhang et al. 2022) introduces class activation value(CAV), to identify true labels from candidate sets, and select labels based on maximum CAV for training.

Several recent studies have extended the classical PLL setting to enhance its robustness in more realistic and complex scenarios. Among these, Lv et al.(Lv et al. 2023) were the first to explore the Unreliable Partial Label Learning (UPLL) setting, where the candidate label set may not contain the ground-truth label. They present the first robustness analysis of average-based strategies in PLL, showing that average partial label losses with bounded losses are inherently robust. (Qiao et al. 2023) propose a theoretically guaranteed framework that simultaneously refines and disambiguates candidate labels to address the UPLL problem, ultimately training a classifier that approximates the Bayes optimal solution. (Wang et al. 2023) propose PiCO+, a framework for UPLL that combines prototype-based disambiguation and contrastive learning, and enhances robustness through clean sample selection and energy-based rejection. Xu et al.(Xu et al. 2021) first proposed the Instance-Dependent Partial Label Learning (IDPLL) framework, in which the construction of the candidate label set is influenced by the instance features. They developed a variational label enhancement framework to recover the latent label distribution and iteratively train the predictive model. Xu et al. (Xu et al. 2023) built on PRODEN to develop

an identification-based approach that iteratively removes false positives and updates the model to handle instance-dependent partial labels.

Although existing methods have made progress in PLL, key challenges in real-world scenarios remain. Most approaches disambiguate incorrect labels by removing or down-weighting them, but these strategies struggle in the UIDPLL setting where the ground-truth label may be absent, making it impossible to recover accurate supervision. Furthermore, instance-dependent candidate labels increase semantic overlap between true and false labels, complicating disambiguation and introducing misleading supervision.

Approach

Problem Setup

Partial Label Learning In the classical PLL setting, the true label of each instance is hidden among a set of candidate labels. Let $\mathcal{X} = \mathbb{R}^d$ be the d-dimensional feature space and $\mathcal{Y} = \{1, 2, \cdots, c\}$ be the label space with c class labels. We denote by $p(\boldsymbol{x}, y)$ the probability density defined on the product space $\mathcal{X} \times \mathcal{Y}$, corresponding to the clean data distribution. The PLL training dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{s}_i) \mid 1 \leq i \leq n\}$ consists of n pairs of instances $\boldsymbol{x}_i \in \mathcal{X}$ and their corresponding candidate label sets $\boldsymbol{s}_i \in \mathcal{S}$. Among them, $\mathcal{S} = 2^{\mathcal{Y}} \backslash \{\emptyset, \mathcal{Y}\}$. The goal of PLL is to train a mutil-class classifier $f : \mathcal{X} \mapsto \mathcal{Y}$ from $\mathcal{D}$ that can accurately predict the correct label for unseen instances.

Unreliable Instance-Dependent Partial Label Learning The classical PLL assumes that partial labels are instance-independent and reliable (i.e., noise-free). This implies that, apart from the latent true label y_i, the incorrect labels have equal probability of being included in the candidate label set $\boldsymbol{s}_i$, and the true label y_i is always contained in $\boldsymbol{s}_i$, i.e.,

$$\begin{aligned} p(y_i \in \boldsymbol{s}_i \mid \boldsymbol{x}_i, \boldsymbol{s}_i) = 1, \\ \forall y_j \neq y_i, \quad p(y_j \in \boldsymbol{s}_i) = r_p, \end{aligned} \tag{1}$$

where r_p is a constant in the range $(0, 1)$.

However, this assumption oversimplifies partial labels and is often violated in practice The probability of a label being included in the candidate set typically depends on instance-specific features, introducing inherent bias. Moreover, without ground-truth labels during annotation, the true label is not guaranteed to appear in the candidate set.

Prior PLL studies have not fully addressed these real-world challenges. To better reflect practical conditions and improve applicability, we introduce a more general and realistic UIDPLL setting, where candidate labels are instance-dependent and may exclude the ground-truth label. Specifically, for each training instance $\boldsymbol{x}_i \in \mathcal{X}$, the candidate label set $\boldsymbol{s}_i \in \mathcal{S}$ satisfies the following conditions:

$$\begin{aligned} p(y_i \in \boldsymbol{s}_i \mid \boldsymbol{x}_i, \boldsymbol{s}_i) = r_n, \forall (\boldsymbol{x}_i, y_i) \sim p(\boldsymbol{x}, \boldsymbol{y}), \forall \boldsymbol{s}_i \in \mathcal{S}, \\ \forall y_j \neq y_i, \quad p(y_j \in \boldsymbol{s}_i \mid \boldsymbol{x}_i) = h\big(p(y_j \mid \boldsymbol{x}_i)\big). \end{aligned} \tag{2}$$

where r_n is called the unreliability rate. $h(\cdot)$ is a monotonically increasing function, which means $y_j \in \boldsymbol{s}_i$ with higher probability if $p(y_j \mid \boldsymbol{x}_i)$ is large.

Compared to conventional settings, such structured and noisy PLs present greater challenges. This is because they can systematically mislead the model into learning incorrect decision boundaries, while the absence of the correct label further strengthens the models confidence in false labels, severely undermining the reliability of the supervision signal. To address this complex real-world problem, we propose a framework named NLAP. NLAP leverages the manifold assumption and the relational structure among samples to enhance true label signals and reduce the ambiguity caused by PLs. In particular, it adopts a neighborhood-based disambiguation strategy that propagates label information from semantically similar instances, improving supervision quality and mitigating the impact of UIDPLL.

The NLAP Framework

According to the research in (Yu et al. 2018; Lv et al. 2020b), for an ordinary multi-class classifier g, if the hypothesis class is sufficiently complex, given infinitely many data and a strictly proper loss function ℓ (such as cross-entropy loss or mean square loss), the optimal classifier $g^* = \arg\min_{g \in \mathcal{G}} \mathbb{E}_{(\boldsymbol{x}, y_{\boldsymbol{x}})}[\ell(f(\boldsymbol{x}), y_{\boldsymbol{x}})]$ can output the posterior probability, that is, for all $j \in \mathcal{Y}, g_j^*(\boldsymbol{x}) = \eta_j(\boldsymbol{x})$. In this study, we use a deep model with the softmax layer as our classifier but relax the approximation in UIDPLL by assuming that the probabilistic output of the classifier trained on a dataset with lower label ambiguity gets closer to the Bayesian posterior probability.

Theoretical Analysis of NLAP In this part, we carry out theoretical analysis of UIDPLL, including label redefinition and generalization error bound. Due to space limitations, the detailed proofs are presented in the supplementary material. Inspired by the investigation (Gong, Yuan, and Bao 2021; Qiao et al. 2023), we first introduce the definitions of Label Ambiguity and (α, ϵ, ρ)-Ambiguity Bounded Distribution before going into the details.

Definition 1 (Label Ambiguity)**.** *First, we denote*

$$U(\tilde{\mathcal{D}}) = \frac{1}{nc} \sum_{(\boldsymbol{x}, \boldsymbol{s}) \in \tilde{D}} \sum_{j \in Y} (\mathbb{I}[j = y_{\boldsymbol{x}}, j \notin \boldsymbol{s}] + \mathbb{I}[j \neq y_{\boldsymbol{x}}, j \in \boldsymbol{s}]) \tag{3}$$

as the label ambiguity of the dataset $\tilde{\mathcal{D}}$, where $\mathbb{I}$ is the indicator function.

According to Definition 1, $U(\tilde{\mathcal{D}}) \in [0, 1]$, refining one correct label from non-candidate labels to candidate labels or disambiguating one incorrect label from candidate labels to non-candidate labels, the label ambiguity reduces $\frac{1}{nc}$.

Definition 2 ((α, ϵ, ρ)-Ambiguity Bounded Distribution)**.** *A UIDPLL distribution $P[\boldsymbol{x}, \boldsymbol{s}]$ is bounded by (α, ϵ, ρ)-ambiguity if there exists a subset G of the support of $P[\boldsymbol{x}, \boldsymbol{s}]$ (i.e., $G \subseteq \mathcal{X} \times \mathcal{Y}$) with a probability mass of at least $1 - \rho$. Specifically, with respect to an appropriate underlying measure μ on $\mathcal{X} \times \mathcal{Y}$, it satisfies $\int_{(\boldsymbol{x}, \boldsymbol{s}) \in G} P[x, \boldsymbol{s}] d\mu(\boldsymbol{x}, \boldsymbol{s}) \geq 1 - \rho$. Moreover, for $f(\boldsymbol{x}) = \arg\min_{f \in \mathcal{F}} \hat{\mathcal{R}}_{PLL}(f)$ and any $\tilde{\mathcal{D}} \subseteq G$, the following condition holds:*

$$\sup_{(\boldsymbol{x}, \boldsymbol{s}) \in \tilde{\mathcal{D}},\, j \in Y} |f_j(\boldsymbol{x}) - \eta_j(\boldsymbol{x})| \leq \alpha U(\tilde{\mathcal{D}}) + \epsilon,$$

where $\hat{\mathcal{R}}_{PLL}(f)$ is the empirical risk, $\alpha \in (0,1)$ is used to resolve the scale problem, and $\epsilon \in (0,1)$ is a small value representing the inherent difference between f and η, which is influenced by factors such as the loss function, sample complexity, and optimization.

Definition 2 reveals that for a UIDPLL dataset $\tilde{\mathcal{D}} \subseteq G$ within the (α, ϵ, ρ)-ambiguity bounded distribution $P[\boldsymbol{x}, \boldsymbol{s}]$, the discrepancy between the classifier f and the posterior probability η is bounded by the label ambiguity of the entire dataset $\tilde{\mathcal{D}}$. If we can refine correct labels or disambiguate incorrect labels in $\tilde{\mathcal{D}}$, the boundary will be narrowed. From this point forward, we make the following assumptions.

Assumption 1. *The UIDPLL dataset $\tilde{\mathcal{D}}$ is always a subset of G within the (α, ϵ, ρ)-ambiguity bounded distribution $P[\boldsymbol{x}, \boldsymbol{s}]$.*

Theorem 1 (Refinement)**.** *Building on Assumption 1, for an instance $\boldsymbol{x}$ with the correct label $y_{\boldsymbol{x}} \notin \boldsymbol{s}$. Let $e = \frac{\sum_{k=1}^{K} \eta_j(\boldsymbol{x}_k)}{|\mathcal{D}_k|} - \alpha U(\tilde{\mathcal{D}}) - \epsilon$, then we can construct a local consistency level set $I(f,e) = \{(\boldsymbol{x}, j) | \frac{\sum_{k=1}^{K} f_j(\boldsymbol{x}_k)}{|\mathcal{D}_k|} > e, j \notin \boldsymbol{s}, \boldsymbol{x}_k \in \mathcal{D}_k\}$ such that $(\boldsymbol{x}, y_{\boldsymbol{x}}) \in I(f,e)$.*

Here $\mathcal{D}_k$ is the K nearest neighbors of $\boldsymbol{x}$. Theorem 1 provides a theoretical guarantee for the procedure process of NLAP. It shows that for an instance $\boldsymbol{x}$ with an unreliable label set, its correct label satisfies the local consistency condition $\frac{\sum_{k=1}^{K} f_{y_{\boldsymbol{x}}}(\boldsymbol{x}_k)}{|\mathcal{D}_k|} > e$ and $(\boldsymbol{x}, y_{\boldsymbol{x}})$ is included in the local consistency level set $I(f,e)$. Hence we can refine the correct label $y_{\boldsymbol{x}_i}$ for the instance $\boldsymbol{x}_i$ by performing the operation $\boldsymbol{s} \cup \{j\}$ on $(\boldsymbol{s}, j) \in \{(\boldsymbol{s}, j) \mid (\boldsymbol{x}, \boldsymbol{s}) \in \tilde{\mathcal{D}}, (\boldsymbol{x}, j) \in I(f,e)\}$. This process sieves labels from non-candidate labels to candidate labels, thereby restoring the reliability of candidate label sets that do not contain the correct labels.

After completing the purification, we will step into a typical PLL scenario. We can further analyze the generalization theoretical analysis of NLAP.

Theorem 2 (Refinement)**.** *Building on Assumption 1, denote $\mathcal{F}$ be a c-valued function class as the family of the hypothesis set. Suppose the loss function ℓ is M-bounded and L_ℓ-Lipschitz. Then, for any $\delta > 0$, with probability at least $1-\delta$ over a sample of size n, the following inequalities holds for all $f \in \mathcal{F}$,*

$$\hat{\mathcal{R}}_{PLL}(f) - \mathcal{R}_{PLL}(f^*) \leq 2\sqrt{2} c L_\ell \sum_{y=1}^{c} \mathfrak{R}_n(F) + M\sqrt{\frac{\log(2/\delta)}{2n}} + cL_\ell * (\alpha U(\tilde{\mathcal{D}}) + \epsilon). \tag{4}$$

Where $\mathfrak{R}_n(F)$ is the Rademacher complexity of $\mathcal{F}$ with sample size n (Bartlett and Mendelson 2002). Theorem 2 shows that the error gap $\hat{\mathcal{R}}_{PLL}(f) - \mathcal{R}_{PLL}(f^*)$ mainly contains two main parts, the first two terms originate from the classical generalization error bound, and the last term arises due to the label ambiguity of the dataset. As $n \to \infty$, $\mathfrak{R}_n(F) \to 0$ for all parametric models with a bounded norm. In addition, the label ambiguity can be continuously approached to 0 with a suitable Refinement and Purification procedure. Consequently, as the number of training data approaches infinity, $\hat{\mathcal{R}}_{PLL}(f)$ can be a appropriate approximation of $\mathcal{R}_{PLL}(f^*)$.

Practical Implementation NLAP progressively incorporates potential true labels to improve the reliability of candidate label sets. Based on the manifold assumption that nearby samples tend to share the same label, it leverages local neighbors in feature space to complete each instances candidate labels. Even when the true label is initially missing, it can often be recovered from neighboring labels, providing an effective cue for disambiguation.

In practice, NLAP first performs a warm-up phase using the observed partial labels to pretrain a randomly initialized classifier for several epochs. During this phase, we adopt a widely used and effective PLL loss (Lv et al. 2020a) to train the model:

$$\hat{\mathcal{R}}_{\text{PLL}}(f) = \sum_{i=1}^{n} \sum_{j=1}^{c} w_{ij} \ell\left(f_j\left(\boldsymbol{x}_i\right), \boldsymbol{s}_i\right). \tag{5}$$

In Eq.(5), $\ell(\cdot)$ denotes the cross-entropy loss, and w_{ij} is the weight assigned to the loss of sample $\boldsymbol{x}_i$ with respect to label j. The weights w_{ij} are initialized uniformly and updated during training based on the model's current predictions, assigning larger weights to labels with higher predicted probabilities:

$$w_{ij} = \begin{cases} f_j\left(\boldsymbol{x}_i\right) / \sum_{j \in \boldsymbol{s}_i} f_j\left(\boldsymbol{x}_i\right), & \text{if } j \in \boldsymbol{s}_i. \\ 0, & \text{otherwise}. \end{cases} \tag{6}$$

This allows the classifier to reach a reasonable state before overfitting noisy labels. After warm-up, NLAP iteratively refines each samples candidate label set by adding labels predicted with high confidence among its nearest neighbors under the current classifier. The classifier is then updated in the next epoch using the refined labels. Ideally, once training finishes, the model generalizes well to unseen instances for reliable prediction. The NLAP process is summarized in Algorithm 1.

Experiment

Benchmark Datasets

To evaluate the effectiveness of the proposed algorithm, we conduct experiments on five benchmark datasets commonly used in deep PLL, including MNIST(LeCun et al. 1998), Kuzushiji-MNIST(Clanuwat et al. 2018), Fashion-MNIST(Xiao, Rasul, and Vollgraf 2017), CIFAR 10, and CIFAR 100(Alex and Hinton 2009).

These datasets are manually corrupted into UIDPLL setting. Specifically, we use predictions from a well-trained neural network to assign flipping probabilities to incorrect labels based on their similarity to the instance. Given the original training set $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$, the corruption process of $\mathcal{D}$ involves two steps:

- To simulate unreliable candidate label sets that may not contain the ground-truth label, we randomly select $r_n \cdot N$ training samples according to a noise rate r_n, and uniformly flip their true labels to other classes $\tilde{y}$.

Algorithm 1: NLAP Algorithm

Input: the UIDPLL training set $\tilde{\mathcal{D}} = \{(\boldsymbol{x}_i, \boldsymbol{s}_i)\}_{i=1}^{n}$, the neighborhood size K, the initial threshold e_0, the threshold step e_{step}, the total epoch T
Output: classifier f

```
Initialize the model parameters via warm-up training.
for t = 1, ..., T do
    Train model f using dataset D̃
    Update threshold: e = e_0 − e_step · (t − 1)
    for x ∈ D̃ do
        Find K nearest neighbors D_k = {x_k}_{k=1}^K of x
        for j ∉ s_i do
            if (Σ_{k=1}^K f_j(x_k)) / |D_k| > e then
                s_j = s_j ∪ {j}
            end if
        end for
    end for
end for
```

- To generate instance-dependent partial label sets, we sample candidate labels based on the probability values output by a well-trained classifier $g(\cdot)$. We use the parameter r_p to control the number of candidate labels:

$$\bar{p}_j(\boldsymbol{x}_i) = \begin{cases} \frac{r_p \cdot g_j(\boldsymbol{x}_i)}{\max(g(\boldsymbol{x}_i) \setminus \{g_{\tilde{y}}(\boldsymbol{x}_i)\}) \cdot \sum_{k \neq j} g_k(\boldsymbol{x}_i)} & , j \neq \tilde{y}_i \\ 0, & j = \tilde{y}_i \end{cases} \tag{7}$$

  Let the $z_i \sim \text{Bernoulli}(\bar{p}_j(\boldsymbol{x}_i))$, The j-th label of label $\boldsymbol{s}_i$ can be represented as:

$$s_{ij} = \begin{cases} 1, & \text{if } j = \tilde{y}_i \\ z_{ij}, & \text{otherwise} \end{cases} \tag{8}$$

Baselines

We compare our proposed NLAP algorithm with eight state-of-the-art partial label learning baselines. These methods are: PRODEN(Lv et al. 2020a) proposes a classifier-consistent risk estimator based on the minimal loss over candidate labels and a strategy to progressively identify true labels. FREDIS(Qiao et al. 2023) combines disambiguation and refinement to correct label sets, with theoretical guarantees of converging to the Bayes optimal classifier. POP(Xu et al. 2023) is a progressive method that iteratively refines candidate labels and updates the model, with provable convergence to the Bayes-optimal classifier under instance-dependent partial labels. ABS-MAE(Lv et al. 2023) employs MAE loss to equally train on all candidate labels, enhancing average-based strategies via bounded loss robustness analysis. ABS-GCE(Lv et al. 2023) is the same as ABS-MAE but uses the GCE loss. CAVL(Zhang et al. 2022) selects the true label as the one with the highest class activation value from the candidate labels using model representations. PiCO+(Wang et al. 2023) disambiguates labels and reduces noise using prototype-based contrastive learning with clean sample selection and semi-supervised training. VALEN(Xu et al. 2021) uses variational inference to estimate true label distributions and trains the model with guidance from a lower bound on the data likelihood.

To ensure fairness, all baseline methods are implemented using the same batch size of 256 and model architecture as used in NLAP. For the MNIST dataset, we adopt LeNet as the classification model. For FMNIST and KMNIST, we use ResNet-18. For the CIFAR 10 and CIFAR 100 datasets, we employ ResNet-18 and ResNet-50 as backbones, respectively. All models are optimized using the Adam algorithm with a momentum of 1e-5. For each method, we adopt the available implementations and apply the recommended hyperparameter settings as specified in the original papers.

For model selection, most existing PLL methods rely on a fully labeled validation set for model selection, which contradicts the core PLL assumption of unavailable ground-truth labels. Inspired by (Wang et al. 2025), we split each UIDPLL training set into 90% training and 10% validation, selecting the model with the highest validation score on two partial label metrics Covering Rate (CR) and Approximated Accuracy (AA) for testing. To further ensure reliability, we additionally report Oracle Accuracy (OA), which uses ground-truth labels on the validation set for model selection. During training, we record the CR, AA, and OA values at each epoch, and after sufficient training, we report the results of the models that achieve the highest validation performance on each respective metric. Each experiment is repeated 5 times with different random seeds, and the average results are reported.

Experiment Results

Table 1 summarize the classification precision of each method on the synthetically UIDPLL datasets under model selection based on CR and OA. Some experimental results are presented in the supplementary material.

In our experiments, for the 10-class datasets MNIST, FMNIST, KMNIST, and CIFAR 10, we evaluate performance under five different (r_n, r_p) settings: $\{(0.1, 0.5), (0.5, 0.1), (0.3, 0.3), (0.3, 0.5), (0.5, 0.3)\}$, to explore performance at varying levels of noise and ambiguity. For the 100-class CIFAR100 dataset, (r_n, r_p) is chosen from $\{(0.1, 0.07), (0.3, 0.05), (0.1, 0.1), (0.3, 0.07), (0.1, 0.05)\}$. The reason for this distinction is that excessively high partial rates in the 100-class setting may render the methods ineffective, thereby undermining the validity of performance comparisons. Experimental results indicate that NLAP consistently outperforms or matches baseline methods across most scenarios, demonstrating its effectiveness and broad applicability. The strong performance under high r_n indicates that the method effectively recovers true labels from unreliable candidate sets, while the robustness under high r_p demonstrates its ability to leverage the enhanced label sets for accurate identification. These results confirm that the neighbor-aware augmentation is not arbitrary or disruptive; instead, it meaningfully guides label refinement and contributes positively to the final classification performance.

Further Analysis

Sensitivity Analysis We conduct a sensitivity analysis on two key hyperparameters: the number of nearest neighbors

Dataset	r_n	r_p	crt.	PRODEN	FREDIS	POP	ABS-MAE	ABS-GCE	CAVL	PiCO+	VALEN	NLAP
MNIST	0.1	0.5	CR	92.90(0.63)	92.77(0.63)	92.85(0.27)	21.68(8.66)	91.41(0.69)	90.82(2.72)	88.50(0.68)	53.87(4.18)	**93.31(0.61)**
			OA	93.21(0.25)	93.12(0.47)	93.10(0.22)	22.07(8.51)	91.70(0.50)	91.35(2.54)	88.83(0.46)	54.12(4.38)	**93.92(0.29)**
	0.5	0.1	CR	97.47(0.12)	97.40(0.26)	97.56(0.20)	97.93(0.17)	97.88(0.25)	95.38(0.93)	97.05(0.12)	97.25(0.25)	**98.00(0.23)**
			OA	97.42(0.20)	97.54(0.13)	97.53(0.18)	**97.97(0.13)**	97.95(0.18)	95.38(0.93)	97.27(0.15)	97.28(0.25)	97.96(0.22)
	0.3	0.3	CR	96.56(0.27)	96.56(0.34)	96.69(0.28)	83.87(11.57)	96.98(0.30)	96.60(0.19)	95.51(0.37)	93.90(0.28)	**97.24(0.16)**
			OA	96.78(0.26)	96.84(0.18)	96.77(0.23)	83.94(11.74)	97.17(0.31)	96.71(0.13)	95.64(0.31)	93.90(0.28)	**97.23(0.18)**
	0.3	0.5	CR	74.88(2.48)	73.79(1.35)	74.95(1.15)	12.34(3.58)	76.38(1.72)	74.14(4.55)	73.98(2.27)	32.14(3.22)	**77.65(0.89)**
			OA	78.21(0.81)	77.77(1.02)	78.33(0.74)	12.58(4.02)	**79.41(1.14)**	75.84(4.60)	79.00(0.98)	35.14(4.32)	78.41(0.91)
	0.5	0.3	CR	92.93(0.52)	92.76(0.70)	93.04(0.35)	44.15(25.47)	93.97(0.67)	92.98(0.89)	91.71(0.70)	85.06(2.67)	**94.38(0.65)**
			OA	93.36(0.51)	93.36(0.58)	93.50(0.47)	44.40(25.60)	94.55(0.28)	93.42(0.60)	92.66(0.66)	85.10(2.66)	**94.57(0.48)**
FMNIST	0.1	0.5	CR	82.52(2.59)	81.64(2.00)	81.74(1.24)	74.20(3.99)	71.67(3.91)	79.01(7.09)	72.38(1.53)	83.01(0.81)	**88.00(0.83)**
			OA	86.11(0.63)	86.19(0.98)	86.13(0.60)	74.59(3.83)	76.64(1.35)	79.14(7.18)	74.98(2.17)	84.24(0.29)	**89.36(0.11)**
	0.5	0.1	CR	86.89(0.61)	86.62(0.75)	86.48(0.40)	86.00(1.42)	87.07(0.87)	85.56(0.85)	85.78(0.84)	85.54(0.58)	**88.36(0.91)**
			OA	87.09(0.43)	87.14(0.23)	87.18(0.26)	86.17(1.60)	87.43(0.50)	85.70(0.66)	85.79(0.85)	85.49(0.57)	**88.50(0.96)**
	0.3	0.3	CR	78.63(2.09)	76.83(3.28)	79.23(1.03)	75.05(4.73)	77.44(3.03)	82.76(0.75)	76.65(1.88)	79.81(1.35)	**84.36(0.61)**
			OA	81.98(1.03)	82.98(1.15)	83.20(0.85)	76.30(5.30)	79.59(1.37)	83.44(0.65)	78.97(2.29)	82.18(0.71)	**86.68(0.39)**
	0.3	0.5	CR	52.72(4.50)	52.33(4.44)	53.58(4.21)	62.70(5.02)	55.28(7.99)	43.49(9.55)	61.45(4.93)	70.29(1.88)	**70.56(1.72)**
			OA	73.00(2.73)	73.51(2.01)	73.20(2.69)	63.63(5.86)	65.88(2.57)	49.47(6.92)	66.80(1.66)	73.96(0.93)	**79.90(1.09)**
	0.5	0.3	CR	56.61(6.37)	51.00(3.79)	59.43(6.95)	59.26(2.17)	56.44(6.21)	66.08(2.71)	61.66(2.33)	59.65(2.45)	**67.21(2.51)**
			OA	70.63(1.86)	70.32(1.25)	71.23(1.82)	61.63(4.10)	67.21(1.57)	70.06(1.49)	67.88(2.56)	68.73(1.13)	**74.52(1.27)**
KMNIST	0.1	0.5	CR	80.49(0.72)	**81.09(0.65)**	79.17(1.21)	61.37(16.98)	71.81(1.71)	73.65(6.29)	72.77(0.72)	63.25(0.76)	80.48(0.70)
			OA	81.15(0.27)	81.45(0.41)	80.23(0.34)	61.91(17.17)	71.91(1.46)	74.12(6.42)	73.48(0.79)	64.08(0.60)	**81.48(0.49)**
	0.5	0.1	CR	87.02(0.85)	86.76(1.42)	87.02(0.85)	90.58(0.95)	88.61(0.77)	83.36(1.78)	84.94(0.80)	82.94(0.47)	**92.35(0.67)**
			OA	86.60(1.51)	86.81(1.42)	86.60(1.51)	90.71(1.14)	88.74(0.38)	83.36(1.78)	84.94(0.80)	82.94(0.47)	**92.52(0.55)**
	0.3	0.3	CR	84.43(1.45)	84.43(1.51)	84.43(1.45)	89.51(0.56)	85.37(0.74)	82.58(0.94)	77.82(1.90)	77.19(1.92)	**90.02(0.70)**
			OA	84.69(0.75)	84.32(1.31)	84.69(0.75)	90.24(0.79)	85.37(0.74)	82.68(0.54)	78.30(2.06)	77.38(1.52)	**90.25(0.62)**
	0.3	0.5	CR	65.11(1.86)	64.45(0.90)	65.11(1.86)	47.95(8.32)	61.56(0.91)	54.91(8.58)	62.01(1.50)	56.27(1.11)	**69.23(1.10)**
			OA	65.85(1.25)	65.98(1.30)	65.85(1.25)	47.90(7.98)	61.40(1.54)	56.01(9.66)	65.20(1.63)	57.02(1.04)	**69.31(1.04)**
	0.5	0.3	CR	73.39(0.65)	74.09(0.94)	73.39(0.65)	77.42(4.67)	73.18(0.89)	71.70(1.22)	70.23(2.10)	66.84(1.69)	**78.80(1.00)**
			OA	73.91(0.69)	74.09(0.94)	73.91(0.69)	78.03(3.97)	73.52(1.54)	72.38(0.87)	71.29(1.25)	67.74(1.31)	**79.60(0.66)**
CIFAR10	0.1	0.5	CR	**71.48(1.50)**	69.91(0.82)	70.29(1.06)	28.53(3.00)	10.51(0.22)	49.97(1.75)	53.06(1.82)	43.17(1.78)	69.67(1.20)
			OA	**73.53(0.61)**	72.42(0.89)	71.80(0.96)	28.64(3.27)	10.98(0.51)	50.44(1.83)	55.85(0.87)	44.43(0.98)	69.98(1.75)
	0.5	0.1	CR	72.61(0.78)	72.42(0.41)	72.28(1.23)	36.87(2.62)	38.35(28.65)	65.75(1.74)	64.54(1.07)	68.64(1.00)	**78.16(1.64)**
			OA	73.36(0.34)	72.84(0.41)	73.19(0.63)	37.05(2.94)	38.98(28.04)	66.27(1.61)	64.98(0.70)	69.51(0.83)	**78.33(1.48)**
	0.3	0.3	CR	65.86(1.48)	64.41(0.75)	64.75(1.62)	26.79(1.39)	64.17(0.97)	54.92(0.72)	56.04(2.47)	50.96(2.63)	**70.01(1.14)**
			OA	68.03(0.75)	67.41(0.46)	67.59(0.84)	26.96(1.70)	69.51(0.96)	55.53(1.06)	57.74(1.13)	53.62(1.64)	**70.61(1.13)**
	0.3	0.5	CR	47.50(1.88)	46.81(1.68)	48.09(1.05)	26.35(1.46)	46.64(2.71)	41.29(4.94)	46.98(2.58)	37.18(1.70)	**52.54(1.16)**
			OA	53.88(1.53)	53.17(0.59)	53.58(0.91)	26.50(1.30)	53.42(1.18)	41.70(4.79)	48.62(1.22)	38.69(0.97)	**54.25(0.37)**
	0.5	0.3	CR	45.49(2.60)	**47.95(0.67)**	46.94(1.08)	24.78(0.51)	39.86(14.62)	42.98(1.72)	46.30(1.05)	41.42(1.92)	45.16(2.30)
			OA	50.77(1.24)	49.45(0.85)	50.68(1.11)	25.18(0.57)	42.84(16.18)	43.65(2.04)	49.45(0.90)	43.64(1.01)	**51.35(0.73)**
CIFAR100	0.1	0.07	CR	51.91(0.45)	51.62(0.65)	52.06(0.83)	12.48(0.66)	45.26(0.84)	39.17(1.10)	42.81(0.82)	32.20(0.37)	**54.17(0.62)**
			OA	51.88(0.76)	51.93(0.69)	52.08(0.94)	12.34(0.60)	45.14(1.04)	39.06(1.23)	43.82(0.60)	32.82(0.45)	**54.11(0.54)**
	0.3	0.05	CR	41.51(0.65)	42.13(0.64)	41.95(0.60)	12.25(1.72)	23.65(18.48)	35.42(1.39)	39.29(1.18)	32.02(0.79)	**48.25(1.27)**
			OA	42.61(0.67)	42.65(0.65)	42.67(0.84)	12.15(1.75)	30.04(16.76)	35.97(1.17)	41.30(0.61)	32.19(0.85)	**48.07(1.63)**
	0.1	0.1	CR	48.44(0.55)	47.84(0.53)	47.29(0.76)	11.67(1.49)	41.32(1.04)	33.01(0.59)	39.21(1.37)	28.94(0.90)	**51.51(0.75)**
			OA	48.38(0.62)	48.28(0.66)	47.56(0.59)	11.45(1.34)	41.57(0.92)	33.38(0.57)	41.57(0.75)	29.09(0.85)	**51.50(0.82)**
	0.3	0.07	CR	38.25(0.88)	37.48(1.15)	38.28(0.88)	10.62(1.09)	26.54(14.92)	30.14(0.79)	37.07(0.68)	28.35(0.79)	**40.03(0.51)**
			OA	39.38(0.67)	38.33(0.82)	39.66(0.23)	10.49(1.09)	35.88(0.43)	30.39(0.64)	39.29(0.94)	28.92(0.44)	**40.06(0.59)**
	0.1	0.05	CR	45.13(18.13)	45.24(18.56)	45.24(18.18)	10.84(4.19)	16.56(21.89)	45.63(0.93)	45.07(1.00)	36.05(0.80)	**56.06(0.46)**
			OA	45.34(18.23)	45.38(18.62)	44.98(18.05)	10.91(4.09)	17.12(22.70)	45.63(0.80)	46.50(0.63)	36.62(1.30)	**56.39(0.63)**

Table 1: Average accuracy(%) with standard deviations over 5 trials on 5 benchmark datasets under different levels of r_n and r_p. NLAP is compared with 8 PLL methods. The best accuracy are highlighted in boldface.

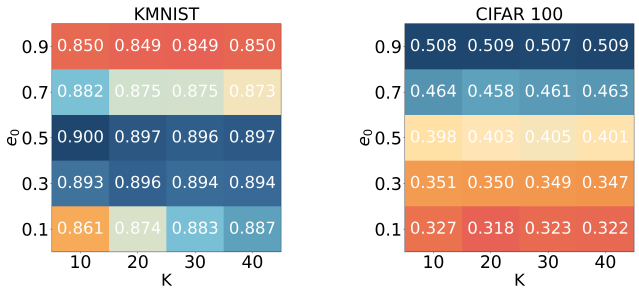

Figure 2: Sensitivity analysis of hyperparameters e_0 and K with $\{r_n, r_p\} = \{0.3, 0.3\}$ ($\{0.1, 0.1\}$ for CIFAR100).

K and the initial threshold e_0, which determine the strength of augmenting candidate label sets via neighboring samples. We perform three runs with varying parameter settings on all benchmark datasets, and report the average results on KMNIST and CIFAR100 in Fig.2. Due to space limitations, results on the remaining datasets are provided in the supplementary material.

As shown in Fig.2, larger e_0 values improve performance on CIFAR 100, while the optimal e_0 on KMNIST lies between 0.3 and 0.5, likely due to differences in data complexity. Overall, performance is more sensitive to e_0 than to K. A K between 20 and 30 yields stable results. We attribute this to the fact that too small a K may fail to provide sufficient samples for label augmentation, while too large a K may include irrelevant neighbors that violate the manifold assumption. These observations provide useful guidance for hyperparameter tuning.

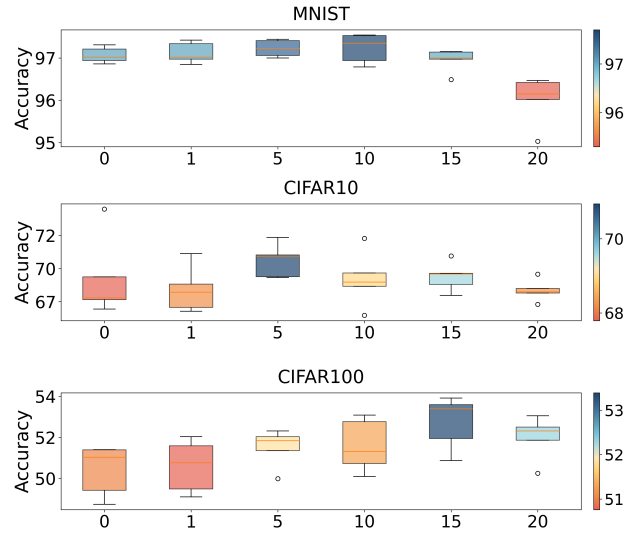

Figure 3: Boxplots of performance accuracy(%) over different warm-up rounds for MNIST, CIFAR10 and CIFAR100.

The Impact of Warm-up To evaluate the effect of the warm-up phase in NLAP, we varied the number of warm-up epochs and performed five independent experiments. The average results on MNIST, CIFAR10 of $\{r_n, r_p\} = \{0.3, 0.3\}$ and CIFAR100 of $\{r_n, r_p\} = \{0.1, 0.1\}$ are shown in Fig.3, while results on other datasets are provided in the supplementary material. Experimental results show that the number of warm-up epochs has little impact on overall performance. However, excessively long warm-up periods may degrade performance on certain datasets, likely due to the model overfitting to wrong labels before effective refinement is applied.

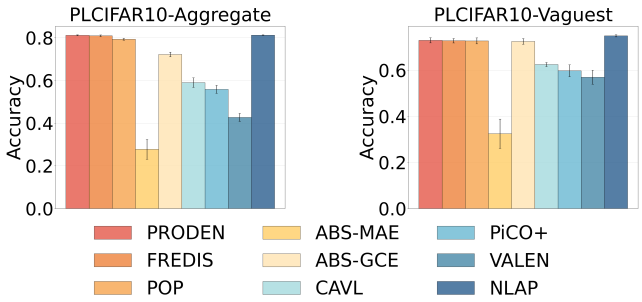

Figure 4: Bar chart of accuracy for 9 methods on PLCIFAR10-Aggregate and PLCIFAR10-Vaguest.

Experimental on Real-world Datasets

Experiments on real-world datasets help demonstrate and advance the evaluation of methods in practical scenarios(Wang et al. 2025; Liu et al. 2025; Song, Kim, and Lee 2019). To evaluate the effectiveness of our proposed method in real-world scenarios, we conducted experiments on an image dataset with manually annotated partial labels introduced in (Wang et al. 2025). This dataset is built upon CIFAR 10 and has two versions, PLCIFAR10-Aggregate and PLCIFAR10-Vaguest, reflecting different strategies for constructing partial labels from human annotations. PLCIFAR10-Aggregate assigns the union of all annotators partial labels to each image, while PLCIFAR10-Vaguest assigns the largest candidate label set provided by any single annotator.

We evaluate our NLAP alongside eight representative baselines on both PLCIFAR10-Aggregate and PLCIFAR10-Vaguest. We ran 5 experiments with different random seeds and present the results as box plots in Fig.4. The results show that our method achieves comparable or superior performance under real annotation conditions, highlighting its robustness and practicality in real-world scenarios.

Conclusion

This paper investigates a novel UIDPLL setting and proposes a new method named NLAP. NLAP is theoretically grounded and trains classifiers by progressively refining candidate label sets via structural similarities among neighboring instances. The analysis shows that appropriate refinement and purification effectively reduce label ambiguity. Extensive experiments on benchmark and real-world datasets demonstrate the methods effectiveness. The exploration of more realistic PLL scenarios contributes to improving the practical applicability of PLL algorithms and reducing the reliance on precisely annotated data in real-world problems.

Acknowledgements

This work was partially supported by the NSF for Distinguished Young Scholars under Grant No. 62425607, the Key NSF of China under Grant No. 62136005. This paper was also partially supported by Postgraduate Scientific Research Innovation Project of Hunan Province under Grant QL20230002.

References

Alex, K.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4).

Bartlett, P. L.; and Mendelson, S. 2002. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. *J. Mach. Learn. Res.*, 3: 463–482.

Chen, Y.; Patel, V. M.; Chellappa, R.; and Phillips, P. J. 2014. Ambiguously Labeled Learning Using Dictionaries. *IEEE Trans. Inf. Forensics Secur.*, 9(12): 2076–2088.

Clanuwat, T.; Bober-Irizar, M.; Kitamoto, A.; Lamb, A.; Yamamoto, K.; and Ha, D. 2018. Deep Learning for Classical Japanese Literature. *CoRR*, abs/1812.01718.

Feng, L.; and An, B. 2019. Partial Label Learning with Self-Guided Retraining. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, 3542–3549. AAAI Press.

Francis, A. 2024. Sensor Independent Cloud and Shadow Masking with Partial Labels and Multimodal Inputs. *IEEE Transactions on Geoscience and Remote Sensing*.

Gong, X.; Yuan, D.; and Bao, W. 2021. Understanding Partial Multi-Label Learning via Mutual Information. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 4147–4156.

Hüllermeier, E.; and Beringer, J. 2006. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5): 419–439.

Jia, Y.; Si, C.; and Zhang, M.-L. 2023. Complementary Classifier Induced Partial Label Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 974–983.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Liu, Y.; Li, W.; Liu, L.; Zhou, J.; Peng, B.; Song, Y.; Xiong, X.; Yang, W.; Liu, T.; Liu, Z.; et al. 2025. ATRNet-STAR: A Large Dataset and Benchmark Towards Remote Sensing Object Recognition in the Wild. *arXiv preprint arXiv:2501.13354*.

Liu, Y.; Peng, B.; Liu, L.; and Li, X. 2024. S^4ST: A Strong, Self-transferable, faSt, and Simple Scale Transformation for Transferable Targeted Attack. *arXiv preprint arXiv:2410.13891*.

Luo, J.; and Orabona, F. 2010. Learning from candidate labeling sets. *Advances in neural information processing systems*, 23.

Lv, J.; Liu, B.; Feng, L.; Xu, N.; Xu, M.; An, B.; Niu, G.; Geng, X.; and Sugiyama, M. 2023. On the robustness of average losses for partial-label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5): 2569–2583.

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

Lv, J.; Xu, M.; Feng, L.; Niu, G.; Geng, X.; and Sugiyama, M. 2020b. Progressive Identification of True Labels for Partial-Label Learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 6500–6510. PMLR.

Qiao, C.; Xu, N.; Lv, J.; Ren, Y.; and Geng, X. 2023. FREDIS: A Fusion Framework of Refinement and Disambiguation for Unreliable Partial Label Learning. In *International Conference on Machine Learning, ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, 28321–28336. PMLR.

Simmler, N.; Sager, P.; Andermatt, P.; Chavarriaga, R.; Schilling, F.; Rosenthal, M.; and Stadelmann, T. 2021a. A Survey of Un-, Weakly-, and Semi-Supervised Learning Methods for Noisy, Missing and Partial Labels in Industrial Vision Applications. In *8th Swiss Conference on Data Science, SDS 2021, Lucerne, Switzerland, June 9, 2021*, 26–31. IEEE.

Simmler, N.; Sager, P.; Andermatt, P.; Chavarriaga, R.; Schilling, F.-P.; Rosenthal, M.; and Stadelmann, T. 2021b. A survey of un-, weakly-, and semi-supervised learning methods for noisy, missing and partial labels in industrial vision applications. In *2021 8th Swiss conference on data science (SDS)*, 26–31. IEEE.

Song, H.; Kim, M.; and Lee, J.-G. 2019. Selfie: Refurbishing unclean samples for robust deep learning. In *International conference on machine learning*, 5907–5915. PMLR.

Tang, C.; and Zhang, M. 2017a. Confidence-Rated Discriminative Partial Label Learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2611–2617. AAAI Press.

Tang, C.-Z.; and Zhang, M.-L. 2017b. Confidence-rated discriminative partial label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Tian, Y.; Yu, X.; and Fu, S. 2023. Partial label learning: Taxonomy, analysis and outlook. *Neural Networks*, 161: 708–734.

Vahedi, B.; Lucas, B.; Banaei-Kashani, F.; Barrett, A. P.; Meier, W. N.; Khalsa, S. J. S.; and Karimzadeh, M. 2024. Partial label learning with focal loss for sea ice classification based on ice charts. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 17: 13616–13633.

Wang, H.; Xiao, R.; Li, Y.; Feng, L.; Niu, G.; Chen, G.; and Zhao, J. 2022. PiCO: Contrastive Label Disambiguation

for Partial Label Learning. In *The Tenth International Conference on Learning Representations, ICLR 2022*. OpenReview.

Wang, H.; Xiao, R.; Li, Y.; Feng, L.; Niu, G.; Chen, G.; and Zhao, J. 2023. Pico+: Contrastive label disambiguation for robust partial label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5): 3183–3198.

Wang, W.; Wu, D.-D.; Wang, J.; Niu, G.; Zhang, M.-L.; and Sugiyama, M. 2025. Realistic Evaluation of Deep Partial-Label Learning Algorithms. In *The Thirteenth International Conference on Learning Representations*.

Wu, D.; Wang, D.; and Zhang, M. 2024. Distilling reliable knowledge for instance-dependent partial label learning. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024*, volume 38, 15888–15896.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *CoRR*, abs/1708.07747.

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*, 38551–38565. PMLR.

Xu, N.; Lv, J.; and Geng, X. 2019. Partial Label Learning via Label Enhancement. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, 5557–5564. AAAI Press.

Xu, N.; Qiao, C.; Geng, X.; and Zhang, M. 2021. Instance-Dependent Partial Label Learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021*, 27119–27130.

Xu, N.; Qiao, C.; Zhao, Y.; Geng, X.; and Zhang, M.-L. 2024. Variational Label Enhancement for Instance-Dependent Partial Label Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yu, X.; Liu, T.; Gong, M.; and Tao, D. 2018. Learning with Biased Complementary Labels. In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., eds., *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I*, volume 11205 of *Lecture Notes in Computer Science*, 69–85. Springer.

Zeng, Z.; Xiao, S.; Jia, K.; Chan, T.-H.; Gao, S.; Xu, D.; and Ma, Y. 2013. Learning by associating ambiguously labeled images. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, 708–715.

Zhang, F.; Feng, L.; Han, B.; Liu, T.; Niu, G.; Qin, T.; and Sugiyama, M. 2022. Exploiting Class Activation Value for Partial-Label Learning. In *The Tenth International Conference on Learning Representations, 2022*. OpenReview.net.

Zhang, M.; Zhou, B.; and Liu, X. 2016. Partial Label Learning via Feature-Aware Disambiguation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1335–1344. ACM.

Zhang, M.-L.; and Yu, F. 2015. Solving the partial label learning problem: An instance-based approach. In *IJCAI*, 4048–4054.

Zhang, M.-L.; Yu, F.; and Tang, C.-Z. 2017. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, 29(10): 2155–2167.