Variational Label Enhancement for Instance-Dependent Partial Label Learning

Ning Xu, *Member, IEEE,* Congyu Qiao, Yuchen Zhao, Xin Geng*, *Senior Member, IEEE,*
and Min-Ling Zhang, *Senior Member, IEEE*

Abstract—Partial label learning (PLL) is a form of weakly supervised learning, where each training example is linked to a set of candidate labels, among which only one label is correct. Most existing PLL approaches assume that the incorrect labels in each training example are randomly picked as the candidate labels. However, in practice, this assumption may not hold true, as the candidate labels are often instance-dependent. In this paper, we address the instance-dependent PLL problem and assume that each example is associated with a latent *label distribution* where the incorrect label with a high degree is more likely to be annotated as a candidate label. Motivated by this consideration, we propose two methods VALEN and MILEN, which train the predictive model via utilizing the latent label distributions recovered by the label enhancement process. Specifically, VALEN recovers the latent label distributions via inferring the variational posterior density parameterized by an inference model with the deduced evidence lower bound. MILEN recovers the latent label distribution by adopting the variational approximation to bound the mutual information among the latent label distribution, observed labels and augmented instances. Experiments on benchmark and real-world datasets validate the effectiveness of the proposed methods.

Index Terms—Label enhancement, partial-label learning, instance-dependent partial-label learning.

1 INTRODUCTION

PARTIAL label learning (PLL) deals with the problem where each training example is associated with a set of candidate labels, among which only one label is valid [5], [7], [50]. Due to the difficulty in collecting exactly labeled data in many real-world scenarios, PLL leverages inexact supervision instead of exact labels. The need to learn from the inexact supervision leads to a wide range of applications for PLL techniques, such as web mining [30], multimedia content analysis [4], [37], ecoinformatics [29], [36], etc.

To accomplish the task of learning from partial label data, many approaches have been proposed. Identification-based PLL approaches [5], [21], [29], [33], [50] regard the ground-truth label as a latent variable and try to identify it. Average-based approaches [7], [20], [53] treat all the candidate labels equally and average the modeling outputs as the prediction. For confidence-based approaches [10], [44], [55], the confidence of each label is estimated instead of identifying the ground-truth label. These approaches always adopt the randomly picked candidate labels to corrupt benchmark data into partially labeled versions despite having no explicit generation process of candidate label sets. To depict the instance-independent generation process of candidate label sets, Feng [11] proposes a statistical model and deduces a risk-consistent method and a classifier-consistent method. Under the same generation process, another classifier-consistent risk estimator is proposed for deep model and stochastic optimizers [31].

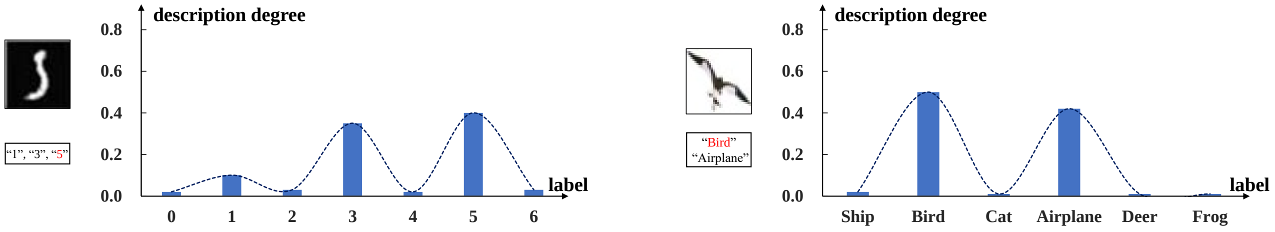
The previous methods assume that the candidate labels are randomly sampled with the uniform generating procedure [11], [31], which is commonly adopted to corrupt benchmark datasets into partially labeled versions in their experiments. However, the candidate labels are always instance-dependent (feature-dependent) in practice as the incorrect labels related to the feature are more likely to be picked as the candidate label set for each instance. These methods usually do not perform as well as expected due to the unrealistic assumption of the generating procedure of candidate label sets.

In this paper, we consider instance-dependent PLL and assume that each instance in PLL is associated with a latent *label distribution* [14], [43], [46] constituted by the real number of each label, representing the degree to each label describing the feature. Then, the incorrect label with a high degree in the latent label distribution is more likely to be annotated as the candidate label. For example, the candidate label set of the handwritten digits in Figure 1(a) contains “1”, “3” and “5”, where “1” and “3” are not ground-truth but selected as candidate labels due to their high degrees in the latent label distribution of the instance. The object in Figure 1(b) is annotated with “bird” and “airplane” as the degrees of these two labels are much higher than others in the label distribution. The intrinsic ambiguity increases the difficulty of annotating, which leads to the result that annotators pick the candidate labels with high degrees in the latent label distribution of each instance instead of annotating the ground-truth label directly in PLL. Therefore, the latent label distribution is the essential labeling information in partially labeled examples and worth being leveraged for predictive model training.

Motivated by the above consideration, we deal with the PLL problem from two aspects. First, we enhance the labeling information by recovering the latent label distribution for each training example as a label enhancement process [43], [46]. Second, we run

- *Ning Xu, Congyu Qiao, Yuchen Zhao and Xin Geng are with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, and the Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China.*
E-mail: {xning, qiaocy, yczhao, xgeng}@seu.edu.cn.
- *Min-Ling Zhang is with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, and the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China.*
E-mail: zhangml@seu.edu.cn.

**Corresponding author.*



(a) Handwritten digits in MNIST [26]

(b) Color image in CIFAR-10 [25]

Fig. 1. The examples about the latent label distributions for partial label learning. The candidate labels are in the box and the red one is valid.

label enhancement and train the predictive model with recovered label distributions iteratively. Given that the label distribution for each example is latent and the data exhibits high dimensionality, calculating the exact posterior density of the label distribution becomes intractable. This complexity arises from the latent nature of the label distributions, which are not directly observable and must be inferred from the data, and the high-dimensional space of the data, which significantly increases the computational complexity and resource requirements for exact inference. As a result, exact computation of the posterior density is not feasible, necessitating the use of approximation methods such as variational inference to estimate the posterior distribution in a computationally efficient manner. We could approximate the true posterior distribution of a model's latent variables given observed data with a simpler, parameterized family of distributions by choosing a distribution from this family (the variational distribution) that is as close as possible to the true posterior distribution.

The proposed method named VALEN, i.e., *VAriational-inference-based Label ENhancement for instance-dependent partial label learning*, uses the candidate labels to initialize the predictive model in the warm-up training stage, then recovers the latent label distributions via inferring the variational posterior density parameterized by an inference model with the deduced evidence lower bound, and trains the predictive model with a risk estimator by leveraging the candidate labels as well as the label distributions. In addition, by adopting the data augmentation on image datasets, MILEN, i.e., *variational-Mutual-Information-based Label ENhancement for instance-dependent partial label learning* is proposed to leverage the mutual information in the label enhancement process. During the label enhancement process, the latent label distribution is estimated by adopting the variational approximation to bound the mutual information among the latent label distribution, observed labels and augmented instances with data augmentation. Our contributions can be summarized as follows:

- We for the first time consider the instance-dependent PLL and assume that each partially labeled example is associated with a latent label distribution, which is the essential labeling information and worth being recovered for predictive model training.
- We propose two methods to recover the latent label distribution for instance-dependent PLL. VALEN infers the posterior density of the latent label distribution via deducing the evidence lower bound for the approximate Dirichlet density, in which the topological information and the features extracted from the predictive model are leveraged. MILEN recovers the latent label distribution by preserving the label-relevant information while discarding the label-irrelevant information, in which the mutual information among the augmented data is leveraged.
- We train the predictive model with a proposed empirical risk estimator by leveraging the candidate labels as well as the label distributions. We iteratively recover the latent label distributions and train the predictive model in every epoch. After the network has been fully trained, the predictive model can perform predictions for future test examples alone.

Experiments on the corrupted benchmark datasets and real-world PLL datasets validate the effectiveness of the proposed method.

Preliminary results of this paper have been previously presented in a shorter conference version [45]. However, in that version, only the variational lower bound was utilized to recover label distributions. In this extended paper, we aim to further investigate the mutual information and propose an alternative method that incorporates label enhancement and classifier training iteratively in each epoch. This new method takes into account data augmentation from the perspective of mutual information. Additionally, we have included more datasets and compared them with additional algorithms in our experiments. Furthermore, we have conducted an active learning experiment on instance-dependent partial label learning datasets to demonstrate the classifier could be improved with human interaction.

The rest of this paper is organized as follows. In Section 2, we provide a concise review and discussion of related work in the field. Section 3 presents the technical details of the proposed methods. This includes a comprehensive explanation of the methods and their implementation. In Section 4, we present the experimental results and provide further analysis and insights based on these results. Finally, in Section 5, we conclude this paper by summarizing the key findings and contributions and discussing potential directions for future research.

2 Related Work

As discussed in Section 1, the supervision information conveyed by partially labeled training examples is implicit due to the hidden ground-truth label within the candidate label set. Consequently, partial label learning can be considered as a learning framework with *weak supervision* [22], where the labeling information is implicitly provided. The primary approach for addressing partial label learning is disambiguation, which involves determining the ground-truth label from the candidate label set associated with each training example. Existing strategies for disambiguation include identification-based and averaging-based approaches. In identification-based disambiguation, the ground-truth label is treated as a latent variable and identified during the learning process [5], [21], [29], [33], [50]. On the other hand, averaging-based disambiguation treats all candidate labels equally and makes predictions by averaging the model's outputs [7], [20], [53].

Many existing algorithms attempt to adapt common learning techniques to handle partial label data and accomplish the learning task. For maximum likelihood techniques, the likelihood of observing each partially labeled training example is defined over its candidate label set rather than the unknown ground-truth label [21], [29]. K-nearest neighbor techniques determine the class label of unseen instances by voting among the candidate labels of its neighboring examples [20], [53]. Maximum margin techniques define classification margins over the partially labeled training examples based on discriminative modeling outputs from candidate labels and non-candidate labels [33], [50]. Boosting techniques update the weight of each partially labeled training example and the confidence of candidate labels in each boosting round [36]. Disambiguation-free strategies estimate the generalized description degree using a graph Laplacian and perform multi-output regression [44]. The confidence of each candidate label is estimated by leveraging the manifold structure of the feature space [55]. However, these methods primarily focus on estimating soft labeling information and training predictive models in separate stages, without considering the feedback from the predictive models.

The aforementioned approaches addressed the problem in specific, low-efficiency manners, which were not compatible with high-efficient stochastic optimization. To tackle large-scale datasets, recent works have employed deep networks with an entropy-based regularizer to maximize the margin between potentially correct labels and unlikely ones [49]. Another approach proposed by [31] introduced a classifier-consistent risk estimator and a progressive identification method, which are compatible with deep models and stochastic optimizers. [11] presented a statistical model to describe the generation process of candidate label sets, leading to the development of risk-consistent and classifier-consistent methods. In the work by [40], label-specific sampling probabilities of candidate label sets were considered, and leveraged weight loss functions were proposed to address the partial label learning problem. [52] differentiated the true label from the candidate set by utilizing class activation values. [39] introduced contrastive representation learning and performed disambiguation based on prototypes. Additionally, [41] designed a regularization loss by revisiting the manifold consistency. These recent approaches aimed to enhance the efficiency and effectiveness of partial label learning.

Previous methods have typically assumed that candidate labels are randomly sampled using a uniform generating procedure. However, in practical scenarios, candidate labels are often instance-dependent or feature-dependent. This means that the incorrect labels related to a specific feature are more likely to be selected as the candidate label set for each instance. In this paper, we focus on instance-dependent partial label learning, where we assume that each instance is associated with a latent *label distribution* [9], [14], [43], [46]. This label distribution consists of real-value numbers for each label, representing the degree to which each label describes the feature. Label enhancement (LE) methods [15], [43], [46] are employed to recover the latent label distribution from observed logical labels. The recovered label distribution serves as a pseudo label [27], [35].

3 Proposed Methods

First of all, we briefly introduce some necessary notations. Let $\mathcal{X} = \mathbb{R}^q$ be the q-dimensional instance space and $\mathcal{Y} = \{y_1, y_2, ..., y_c\}$ be the label space with c class labels. Given the PLL training set $\mathcal{D} = \{(\boldsymbol{x}_i, S_i) | 1 \leq i \leq n\}$ where $\boldsymbol{x}_i$ denotes the q-dimensional instance and $S_i \subseteq \mathcal{Y}$ denotes the candidate label set associated with $\boldsymbol{x}_i$. Note that S_i contains the correct label $y_{\boldsymbol{x}_i}$ of $\boldsymbol{x}_i$ and the task of PLL is to induce a multi-class classifier $f : \mathcal{X} \mapsto \mathcal{Y}$ from $\mathcal{D}$. For each PLL training example $(\boldsymbol{x}_i, S_i)$, we use the logical label vector $\boldsymbol{l}_i = [l_i^{y_1}, l_i^{y_2}, \ldots, l_i^{y_c}]^\top \in \{0,1\}^c$ to represent whether y_j is the candidate label, i.e., $l_i^{y_j} = 1$ if $y_j \in S_i$, otherwise $l_i^{y_j} = 0$. The label distribution of $\boldsymbol{x}_i$ is denoted by $\boldsymbol{d}_i = [d_i^{y_1}, d_i^{y_2}, \ldots, d_i^{y_c}]^\top \in [0,1]^c$ where $\sum_{j=1}^{c} d_i^{y_j} = 1$. Then $\mathbf{L} = [\boldsymbol{l}_1, \boldsymbol{l}_2, \ldots, \boldsymbol{l}_n]$ and $\mathbf{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n]$ represent the logical label matrix and label distribution matrix, respectively. Let $\boldsymbol{z}_i$ denote the latent feature which is the latent representation of observed instance $\boldsymbol{x}_i$. The representation captures the intrinsic patterns and relationships within the instance, providing a more meaningful dimension of information for the training. Then $\mathbf{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_n]$ represents the latent feature matrix. The instance-dependent partial labels of $\boldsymbol{x}_i$ are generated by the process that the ground-truth label is selected as a partial label and each incorrect label is selected as a partial label with a certain probability.

3.1 Overview

To address the instance-dependent partial label learning problem, the proposed approaches, Valen and Milen, follow an iterative process. They aim to recover the latent label distribution for each example $\boldsymbol{x}$ and subsequently train the predictive model using the recovered label distribution.

Before the main iterative process, a warm-up period is introduced, as discussed in Subsection 3.2. During this phase, the predictive model is trained using the PLL minimal loss [31]. This warm-up period helps to establish a reasonable predictive model before it starts fitting incorrect labels. Once the warm-up period is completed, the features extracted from the predictive model are utilized to recover the latent label distribution for each example, thereby enhancing the overall performance of the partial label learning process.

The details of the label enhancement process for instance-dependent PLL are proposed in Subsection 3.3. Valen recovers the latent label distributions via inferring the variational posterior density parameterized by an inference model with the deduced evidence lower bound. Then Valen derives the evidence lower bound for optimizing the inference model, and the label distributions can be generated from the variational posterior using this optimized model. Milen recovers the latent label distribution by adopting the variational approximation to bound the mutual information among the latent label distribution, observed labels, and augmented instances with data augmentation. During the label enhancement process, the primary objective is to maximize the preservation of label-relevant information within the recovered label distribution while discarding label-irrelevant information effectively.

In classifier training, the predictive model is trained by leveraging the recovered label distributions and candidate labels with the proposed empirical risk estimator in Subsection 3.4. We implement label enhancement and classifier training iteratively in every epoch. Once the models have been fully trained, the predictive model can make predictions for future test instances independently.

3.2 Warm-up Training

The predictive model $\boldsymbol{\theta}$ is trained on partially labeled examples by minimizing the PLL minimal loss function [31] defined as follows:

$$\mathcal{L}_{min} = \sum_{i=1}^{n} \min_{y_j \in S_i} \ell(f(\boldsymbol{x}_i), \boldsymbol{e}^{y_j}), \tag{1}$$

where ℓ is cross-entropy loss and $\boldsymbol{e}^{\mathcal{Y}} = \{\boldsymbol{e}^{y_j} : y_j \in \mathcal{Y}\}$ denotes the standard canonical vector in $\mathcal{R}^c$, i.e., the j-element in $\boldsymbol{e}^{y_j}$ equals 1 and others equal 0. Similar to [31], the $\mathtt{min}$ operator in Eq. (1) is replaced by using the current predictions for weighting on the possible labels in warm-up training. Then we could extract the feature $\boldsymbol{\phi}$ of each $\boldsymbol{x}$ via using the predictive model.

3.3 Variational Label Enhancement

3.3.1 VALEN

We assume that the prior density $p(\boldsymbol{d})$ is a Dirichlet with $\hat{\boldsymbol{\alpha}}$, i.e., $p\left(\boldsymbol{d}\right) = Dir\left(\boldsymbol{d} \mid \hat{\boldsymbol{\alpha}}\right)$ where $\hat{\boldsymbol{\alpha}} = [\varepsilon, \varepsilon, \ldots, \varepsilon]^\top$ is a c-dimensional vector with a minor value ε. Then we let the prior density $p(\mathbf{D})$ be the product of each Dirichlet

$$p(\mathbf{D}) = \prod_{i=1}^{n} Dir(\boldsymbol{d}_i|\hat{\boldsymbol{\alpha}}). \tag{2}$$

Similarly, the prior density $p(\boldsymbol{z})$ is assumed to be a standard Gaussian with mean $\hat{\boldsymbol{\mu}}$ and standard deviation $\hat{\boldsymbol{\sigma}}$ where $\hat{\boldsymbol{\mu}} = [0, 0, \ldots, 0]$ and $\hat{\boldsymbol{\sigma}} = [1, 1, \ldots, 1]$. The prior density for the entire latent feature $\mathbf{Z}$ is then represented as the product of each Gaussian distribution for each example $\boldsymbol{z}_i$

$$p(\mathbf{Z}) = \prod_{i=1}^{n} Gau(\boldsymbol{z}_i|\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}). \tag{3}$$

Next, we consider the topological information of the feature space using the affinity graph $G = (V, E, \mathbf{A})$. The vertex set V corresponds to the feature vectors of each example and contains n elements: $V = \boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_n$. The edge set E corresponds to pairs of feature vectors that represent the relationships between different examples. Each edge is represented as $E = (\boldsymbol{\phi}_i, \boldsymbol{\phi} j) \mid 1 \leq i \neq j \leq n$. The sparse adjacency matrix $\mathbf{A}$ is used to represent the edge connections in the graph, where $aij = 1$ if $\boldsymbol{\phi}_i$ is in the set $\mathcal{N}(\boldsymbol{\phi}_j)$, which contains the k-nearest neighbors of $\boldsymbol{\phi}_j$. The diagonal elements of $\mathbf{A}$ are set to 1 to indicate self-connections.

We consider the topological information of the feature space, which is represented by the affinity graph $G = (V, E, \mathbf{A})$. Here, the feature vector $\boldsymbol{\phi}_i$ of each example could be extracted from the predictive model $\boldsymbol{\theta}$ in the current epoch, $V = \{\boldsymbol{\phi}_i \mid 1 \leq i \leq n\}$ denotes the vertex set V corresponds to the feature vectors of each example, each edge is represented as $E = \{(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) \mid 1 \leq i \neq j \leq n\}$, and a sparse adjacency matrix $\mathbf{A} = [a_{ij}]_{n \times n}$ can be obtained by

$$a_{ij} = \left\{ \begin{array}{ll} 1 & \text{if } \boldsymbol{\phi}_i \in \mathcal{N}(\boldsymbol{\phi}_j) \\ 0 & \text{otherwise} \end{array} \right., \tag{4}$$

where $\mathcal{N}(\boldsymbol{\phi}_j)$ is the set for k-nearest neighbors of $\boldsymbol{\phi}_j$ and the diagonal elements of $\mathbf{A}$ are set to 1.

The goal of the VALEN is to approximate the posterior density $p(\mathbf{D}, \mathbf{Z}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})$, where $\boldsymbol{\Phi}$, $\mathbf{A}$, and $\mathbf{L}$ represent observed feature matrix, adjacency matrix, and logical labels, respectively. Due to computational complexity, the exact posterior density is challenging to compute. To address this, the posterior density is decomposed as

$$p(\mathbf{D}, \mathbf{Z}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A}) = p(\mathbf{D}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A}) p(\mathbf{Z}|\mathbf{D}, \boldsymbol{\Phi}). \tag{5}$$

Here, we assume that $\mathbf{Z}$ is independent of $\mathbf{L}$ and $\mathbf{A}$ when the latent variable $\mathbf{D}$ is given in the condition, allowing us to remove $\mathbf{L}$ and $\mathbf{A}$ from the conditional distribution $p(\mathbf{Z}|\mathbf{D}, \mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})$. This independence assumption simplifies the modeling process and makes the inference more tractable.

The fixed-form density $q(\mathbf{D}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})$ and $q(\mathbf{Z}|\mathbf{D}, \boldsymbol{\Phi})$ are employed to approximate the true posterior $p(\mathbf{D}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})$ and $p(\mathbf{Z}|\mathbf{D}, \boldsymbol{\Phi})$ respectively. We let the approximate posterior $q_{\boldsymbol{w}_1}(\mathbf{D}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})$ be the product of each Dirichlet parameterized by a vector $\boldsymbol{\alpha}_i = [\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^c]^\top$ and let the $q_{\boldsymbol{w}_2}(\mathbf{Z}|\mathbf{D}, \boldsymbol{\Phi})$ be the product of each Gaussian parameterized by the mean vector $\boldsymbol{\mu}_i = [\mu_i^1, \mu_i^2, \ldots, \mu_i^J]$ and standard deviation vector $\boldsymbol{\sigma}_i = [\sigma_i^1, \sigma_i^2, \ldots, \sigma_i^J]$ where J is the dimension of the latent feature $\mathbf{Z}$:

$$q_{\boldsymbol{w}_1}(\mathbf{D} \mid \mathbf{L}, \boldsymbol{\Phi}, \mathbf{A}) = \prod_{i=1}^{n} Dir\left(\boldsymbol{d}_i|\boldsymbol{\alpha}_i\right), \tag{6}$$

$$q_{\boldsymbol{w}_2}(\mathbf{Z}|\mathbf{D}, \boldsymbol{\Phi}) = \prod_{i=1}^{n} Gau(\boldsymbol{z}_i|\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i). \tag{7}$$

Here, the parameters $\boldsymbol{\Delta} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_n]$ are outputs of the inference model parameterized by $\boldsymbol{w}_1$, which is defined as a two-layer GCN [24] by $\mathrm{GCN}(\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A}) = \tilde{\mathbf{A}}\,\mathrm{ReLU}\left(\tilde{\mathbf{A}}\mathbf{U}\mathbf{W}_0\right)\mathbf{W}_1$, with $\mathbf{U} = [\boldsymbol{\Phi}; \mathbf{L}]$ and weights $\mathbf{W}_0$, $\mathbf{W}_1$. Here $\tilde{\mathbf{A}} = \hat{\mathbf{A}}^{-\frac{1}{2}}\mathbf{A}\hat{\mathbf{A}}^{-\frac{1}{2}}$ is the symmetrically normalized weight matrix where $\hat{\mathbf{A}}$ is the degree matrix of $\mathbf{A}$. The parameters $\boldsymbol{\Lambda} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_n, \boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \ldots, \boldsymbol{\sigma}_n]$ are outputs of a inference model parameterized by $\boldsymbol{w}_2$, which is defined as a 4-hidden-layer convolutional networks.

By employing Variational Bayes techniques, we can derive a lower bound on the marginal likelihood of the model, ensuring that the approximate posterior distribution $q_{\boldsymbol{w}}(\mathbf{D}, \mathbf{Z}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})$ closely approximates the true posterior $p(\mathbf{D}, \mathbf{Z}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})$. Given the logical label matrix $\mathbf{L}$, feature matrix $\boldsymbol{\Phi}$, and the corresponding $\mathbf{A}$, the evidence lower bound (ELBO) can be derived as follows:

$$\begin{aligned} \mathcal{L}_{ELBO} &= \mathbb{E}_{q_{\boldsymbol{w}}(\mathbf{D}, \mathbf{Z}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})}[\log p(\boldsymbol{\Phi}|\mathbf{D}, \mathbf{Z})] \\ &+ \mathbb{E}_{q_{\boldsymbol{w}}(\mathbf{D}, \mathbf{Z}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})}[\log p(\mathbf{L}, \mathbf{A}|\mathbf{D})] \\ &- \mathrm{KL}[q_{\boldsymbol{w}_1}(\mathbf{D}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})||p(\mathbf{D})] \\ &- \mathrm{KL}[q_{\boldsymbol{w}_2}(\mathbf{Z}|\mathbf{D}, \boldsymbol{\Phi})||p(\mathbf{Z})]. \end{aligned} \tag{8}$$

According to Eq. (2) and Eq. (6) , the first KL divergence in Eq. (8) can be analytically calculated as follows:

$$\begin{aligned} \mathrm{KL}\left(q_{\boldsymbol{w}_1}(\mathbf{D}|\mathbf{L}, \boldsymbol{\Phi}, \mathbf{A})|p(\mathbf{D})\right) &= \sum_{i=1}^{n} \Bigg(\log\Gamma(\sum_{j=1}^{c} \alpha_i^j) \\ &- \sum_{j=1}^{c} \log\Gamma(\alpha_i^j) - \log\Gamma(c \cdot \varepsilon) + c\log\Gamma(\varepsilon) \\ &+ \sum_{j=1}^{c} (\alpha_i^j - \varepsilon)(\psi(\alpha_i^j) - \psi(\sum_{j=1}^{c} \alpha_i^j)) \Bigg), \end{aligned} \tag{9}$$

where $\Gamma(\cdot)$ and $\psi(\cdot)$ are Gamma function and Digamma function, respectively.

According to Eq. (3) and Eq. (7), the second KL divergence in Eq. (8) can be analytically calculated as follows:

$$\begin{aligned}\mathrm{KL}(q_{\boldsymbol{w}_2}(\mathbf{Z}|\mathbf{D},\mathbf{\Phi})||p(\mathbf{Z})) = \sum_{i=1}^{n}\sum_{j=1}^{J}\Big(1+\log((\sigma_i^j))\\ -(\mu_i^j)^2-(\sigma_i^j)^2\Big).\end{aligned} \tag{10}$$

As the first term in Eq. (8) is computationally intractable, we employ the implicit reparameterization trick [12] to approximate it using Monte Carlo (MC) estimation. $p(\boldsymbol{\phi}_i|\mathbf{D},\mathbf{Z})$ is assumed to be product of each Gaussian parameterized by the mean vector $\bar{\boldsymbol{\mu}}_i^{(m)} = [\bar{\mu}_i^{1(m)}, \bar{\mu}_i^{2(m)}, \ldots, \bar{\mu}_i^{q(m)}]$ and standard deviation vector $\bar{\boldsymbol{\sigma}}_i = [\bar{\sigma}_i^{1(m)}, \bar{\sigma}_i^{2(m)}, \ldots, \bar{\sigma}_i^{q(m)}]$ which are computed from m-th sampling $\mathbf{D}^{(m)}$ and $\mathbf{Z}^{(m)}$ with a 4-hidden-layer convolutional network $p_{\boldsymbol{\eta}_1}(\mathbf{\Phi}|\mathbf{D},\mathbf{Z})$ parameterized by $\boldsymbol{\eta}_1$ where q is the dimension of $\boldsymbol{\phi}$, then the first term of Eq. (8) can be calculated:

$$\begin{aligned}\mathbb{E}_{q_{\boldsymbol{w}}(\mathbf{D},\mathbf{Z}|\mathbf{L},\mathbf{\Phi},\mathbf{A})}[\log p(\mathbf{\Phi}|\mathbf{D},\mathbf{Z})] = \frac{1}{M}\sum_{i}^{n}\sum_{j}^{q}\sum_{m}^{M}\\ \Big(-\frac{1}{2}\log 2\pi - \log\bar{\sigma}_i^{j(m)} - \frac{(\phi_i^j-\bar{\mu}_i^{j(m)})^2}{2\bar{\sigma}_i^{j(m)}}\Big).\end{aligned} \tag{11}$$

For the second term of Eq. (8), we decompose it as:

$$\begin{aligned} p(\mathbf{L}\mid\mathbf{D}) &= \prod_{i=1}^{n} p\left(\boldsymbol{l}_i\mid\mathbf{D}\right),\\ p(\mathbf{A}\mid\mathbf{D}) &= \prod_{i=1}^{n}\prod_{j=1}^{n} p\left(a_{ij}\mid\boldsymbol{d}_i,\boldsymbol{d}_j\right),\\ \text{with } & p\left(a_{ij}=1\mid\boldsymbol{d}_i,\boldsymbol{d}_j\right) = s\left(\boldsymbol{d}_i^{\top}\boldsymbol{d}_j\right).\end{aligned} \tag{12}$$

Here, the function $s(\cdot)$ refers to the logistic sigmoid function. Additionally, we assume that $p\left(\boldsymbol{l}_i|\mathbf{D}\right)$ follows a multivariate Bernoulli distribution with probabilities $\boldsymbol{\tau}_i$. To simplify the observation model, we compute $\mathbf{T}^{(m)} = [\boldsymbol{\tau}_1^{(m)}, \boldsymbol{\tau}_2^{(m)}, \ldots, \boldsymbol{\tau}_n^{(m)}]$ from the m-th sampling of $\mathbf{D}^{(m)}$ using a three-layer MLP parameterized by $\boldsymbol{\eta}_2$. Consequently, we can calculate the second term of Eq. (8) as follows:

$$\begin{aligned}\mathbb{E}_{q_{\boldsymbol{w}}(\mathbf{D},\mathbf{Z}|\mathbf{L},\mathbf{\Phi},\mathbf{A})}[\log p_{\boldsymbol{\eta}_2}(\mathbf{L},\mathbf{A}|\mathbf{D})] =\\ \frac{1}{M}\sum_{m=1}^{M}\Bigg(\mathrm{tr}\left((\mathbf{I}-\mathbf{L})^{\top}\log\left(\mathbf{I}-\mathbf{T}^{(m)}\right)\right)\\ +\mathrm{tr}\left(\mathbf{L}^{\top}\log\mathbf{T}^{(m)}\right) - \|\mathbf{A}-S\left(\mathbf{D}^{(m)}\mathbf{D}^{(m)\top}\right)\|_F^2\Bigg).\end{aligned} \tag{13}$$

It is worth mentioning that during the training process, we can use only one Monte Carlo (MC) sample in Eq. (13), as suggested in [23], [46]. This can significantly reduce the computational burden and make the training more efficient.

Moreover, VALEN enhances the label enhancement process by incorporating the compatibility loss, which ensures that the reconstructed label distributions should not deviate significantly from the estimated confidences $\zeta(\boldsymbol{x}_i)$ [11], [31] obtained from the current prediction $f(\boldsymbol{x}_i;\boldsymbol{\theta})$:

$$\mathcal{L}_o = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c}\zeta_j(\boldsymbol{x}_i)\log d_i^{y_j}, \tag{14}$$

where

$$\zeta_j(\boldsymbol{x}_i) = \begin{cases}\frac{f_j(\boldsymbol{x}_i;\boldsymbol{\theta})}{\sum_{y_k\in S_i} f_k(\boldsymbol{x}_i;\boldsymbol{\theta})} & \text{if } y_j\in S_i\\ 0 & \text{otherwise.}\end{cases} \tag{15}$$

Algorithm 1 VALEN Algorithm

```
Input: The PLL training set $\mathcal{D} = \{(\boldsymbol{x}_i, S_i)\}_{i=1}^n$, epoch $T$ and iteration $I$;
Initialize the predictive model $\boldsymbol{\theta}$ by warm-up training. The observation model $\boldsymbol{\eta} = [\boldsymbol{\eta}_1, \boldsymbol{\eta}_2]$ and reference model $\boldsymbol{w} = [\boldsymbol{w}_1, \boldsymbol{w}_2]$ are initialized with some initial values.
Extract the features $\mathbf{\Phi}$ by using the predictive model $\boldsymbol{\theta}$ after warm-up training and then calculate $\mathbf{A}$;
for $t = 1, \ldots, T$ do
    Shuffle the training set $\mathcal{D} = \{(\boldsymbol{x}_i, S_i)\}_{i=1}^n$ into $I$ mini-batches;
    for $k = 1, \ldots, I$ do
        Obtain label distribution $\boldsymbol{d}_i$ corresponding to each example $\boldsymbol{x}_i$ by Eq. (6);
        Update $\boldsymbol{\theta}$, $\boldsymbol{w}$ and $\boldsymbol{\eta}$ by forward computation and back-propagation by fusing Eq. (16) and Eq. (28);
    end for
end for
Output: The predictive model $\boldsymbol{\theta}$.
```

Now, we can straightforwardly derive the objective function for label enhancement as

$$\mathcal{L}_{LE} = \lambda\mathcal{L}_o - \mathcal{L}_{ELBO}, \tag{16}$$

where λ is a hyper-parameter. The label distribution matrix $\mathbf{D}$ is sampled from $q(\mathbf{D}|\mathbf{L},\mathbf{\Phi},\mathbf{A})$, i.e., $\boldsymbol{d}_i \sim Dir(\boldsymbol{\alpha}_i)$. Importantly, the implicit reparameterization gradient [12] is employed, allowing the gradients to be computed analytically in the backward pass. This avoids the inversion of the standardization function and facilitates efficient optimization during training.

Finally, VALEN trains the classifier in Subsection 3.4 via using the recovered label distributions. The algorithmic description of the VALEN is shown in Algorithm 1.

3.3.2 MILEN

In this subsection, we propose a novel label enhancement approach for instance-dependent PLL named MILEN, which leverages the mutual information [1], [17], [57] among augmented data through data augmentation [8], [39]. Let $\mathcal{A}(\boldsymbol{x}) = \{\mathrm{Aug}_k(\boldsymbol{x})|1 \le k \le K\}$ be the set of randomly augmented instances, where $\mathrm{Aug}_k(\boldsymbol{x})$ denotes k-th random augmentation of the original instance, and K denotes the number of augmentations. The k-th augmented instance matrix is denoted by $\mathbf{X}^{\mathrm{Aug}_k} = [\mathrm{Aug}_k(\boldsymbol{x}_1), \mathrm{Aug}_k(\boldsymbol{x}_2), \ldots, \mathrm{Aug}_k(\boldsymbol{x}_n)]$, and the label distribution matrix corresponding to $\mathbf{X}^{\mathrm{Aug}_k}$ is denoted by $\mathbf{D}^{\mathrm{Aug}_k}$. For convenience, let $\mathbf{X}^{\mathrm{Aug}_0}$ denote the original instance $\mathbf{X}$, and $\mathbf{D}^{\mathrm{Aug}_0}$ denote the latent label distribution matrix corresponding to the original instance $\mathbf{X}$.

Let $I(\mathbf{X},\mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y})$ denote the mutual information, a Shannon entropy-based measure of relevant information between two random variables $\mathbf{X}$ and $\mathbf{Y}$, where $H(\mathbf{X}) = -\int \mathrm{d}\boldsymbol{x}p(\boldsymbol{x})\log p(\boldsymbol{x})$ is the Shannon entropy for the variable $\mathbf{X}$, and $H(\mathbf{X}|\mathbf{Y}) = -\int \mathrm{d}(\boldsymbol{x},\boldsymbol{y})p(\boldsymbol{x},\boldsymbol{y})\log p(\boldsymbol{x}|\boldsymbol{y})$ is the conditional entropy for $\mathbf{X}$ given $\mathbf{Y}$. Also, let $I(\mathbf{X},\mathbf{Y}|\mathbf{Z}) = H(\mathbf{X}|\mathbf{Z}) - H(\mathbf{X}|\mathbf{Y},\mathbf{Z})$ denote the conditional mutual information,

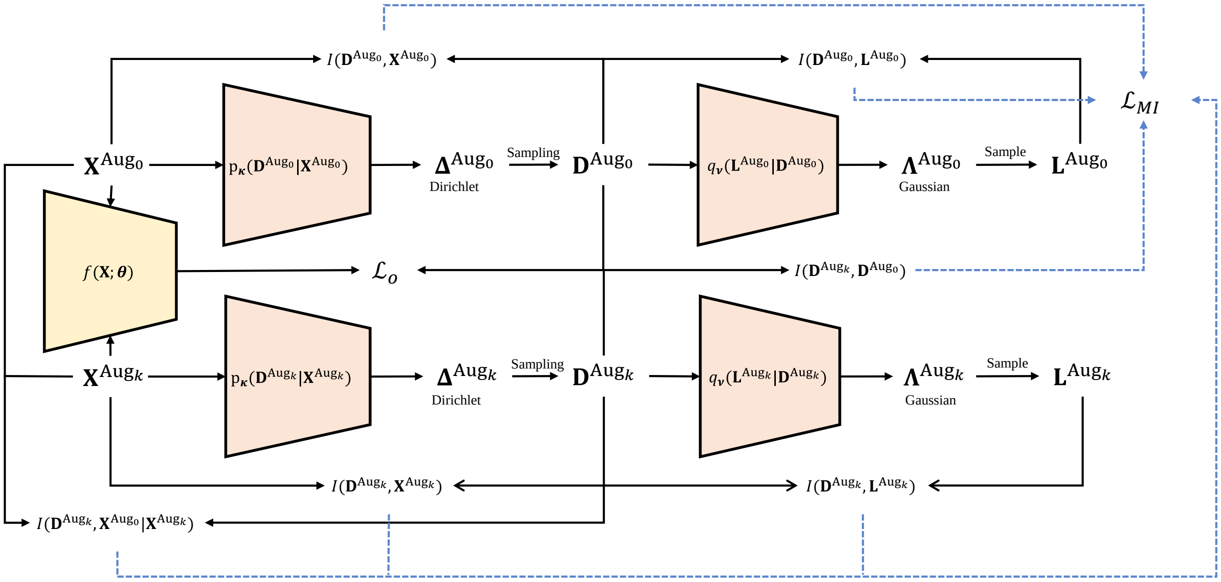

$I(\mathbf{D}^{\text{Aug}_0}, \mathbf{X}^{\text{Aug}_0})$
$I(\mathbf{D}^{\text{Aug}_0}, \mathbf{L}^{\text{Aug}_0})$
$\mathcal{L}_{MI}$
$\mathbf{X}^{\text{Aug}_0}$
$p_{\boldsymbol{\kappa}}(\mathbf{D}^{\text{Aug}_0}|\mathbf{X}^{\text{Aug}_0})$
$\boldsymbol{\Delta}^{\text{Aug}_0}$
Dirichlet
Sampling
$\mathbf{D}^{\text{Aug}_0}$
$q_{\boldsymbol{\nu}}(\mathbf{L}^{\text{Aug}_0}|\mathbf{D}^{\text{Aug}_0})$
$\boldsymbol{\Lambda}^{\text{Aug}_0}$
Gaussian
Sample
$\mathbf{L}^{\text{Aug}_0}$
$f(\mathbf{X};\boldsymbol{\theta})$
$\mathcal{L}_o$
$I(\mathbf{D}^{\text{Aug}_k}, \mathbf{D}^{\text{Aug}_0})$
$\mathbf{X}^{\text{Aug}_k}$
$p_{\boldsymbol{\kappa}}(\mathbf{D}^{\text{Aug}_k}|\mathbf{X}^{\text{Aug}_k})$
$\boldsymbol{\Delta}^{\text{Aug}_k}$
Dirichlet
Sampling
$\mathbf{D}^{\text{Aug}_k}$
$q_{\boldsymbol{\nu}}(\mathbf{L}^{\text{Aug}_k}|\mathbf{D}^{\text{Aug}_k})$
$\boldsymbol{\Lambda}^{\text{Aug}_k}$
Gaussian
Sample
$\mathbf{L}^{\text{Aug}_k}$
$I(\mathbf{D}^{\text{Aug}_k}, \mathbf{X}^{\text{Aug}_k})$
$I(\mathbf{D}^{\text{Aug}_k}, \mathbf{L}^{\text{Aug}_k})$
$I(\mathbf{D}^{\text{Aug}_k}, \mathbf{X}^{\text{Aug}_0}|\mathbf{X}^{\text{Aug}_k})$

Fig. 2. An illustration of the optimization procedure in our MILEN approach, where the inference model $\boldsymbol{\kappa}$ and $\boldsymbol{\nu}$ are respectively instantiated by MLPs to recover the label distributions from the original training data and augmented training data.

a measure of relevant information between two random variables $\mathbf{X}$ and $\mathbf{Y}$ given $\mathbf{Z}$. According to the definition of mutual information, the larger $I(\mathbf{X},\mathbf{Y})$ is, the more relevant information in $\mathbf{X}$ is provided by $\mathbf{Y}$.

The framework of MILEN is illustrated in Fig 2. Firstly, we consider the recovered label distribution in one augmentation. As we mention in Section 1, a label with a high degree in the latent label distribution is more likely to be annotated as the candidate label. This suggests that the recovered label distribution $\mathbf{D}^{\text{Aug}_k}$ should be highly relevant to the logical label matrix $\mathbf{L}^{\text{Aug}_k}$ in one augmentation, i.e., $\mathbf{D}^{\text{Aug}_k}$ should preserve label-relevant information. Meanwhile, since the instance contains label-irrelevant information, e.g. object detection from a complex background image, the recovered label distribution $\mathbf{D}^{\text{Aug}_k}$ should discard the label-irrelevant information in the instance matrix $\mathbf{X}^{\text{Aug}_k}$.

In addition, we consider the recovered label distribution between the original instance and its augmentations. Since the instance matrix $\mathbf{X}^{\text{Aug}_k}$ of all augmentation is augmented from $\mathbf{X}^{\text{Aug}_0}$, their corresponding latent label distributions should be closely relevant. Besides, the recovered label distribution is encouraged to contain augmentation-specific information.

Based on the above consideration and the definition of mutual information, i.e., the larger $I(\mathbf{X},\mathbf{Y})$ is, the more relevant information in $\mathbf{X}$ is provided by $\mathbf{Y}$, we have the following optimization objective to estimate the latent label distributions in each augmentation:

$$\begin{aligned}\min_{\mathbf{D}^{\text{Aug}_0},\mathbf{D}^{\text{Aug}_1},\ldots,\mathbf{D}^{\text{Aug}_k}} &\sum_{k=0}^{K}\big[-\beta_1 I(\mathbf{D}^{\text{Aug}_k},\mathbf{L})+ \\ I(\mathbf{D}^{\text{Aug}_k},\mathbf{X}^{\text{Aug}_k})\big] &+ \sum_{k=1}^{K}\big[-\beta_2 I(\mathbf{D}^{\text{Aug}_k},\mathbf{D}^{\text{Aug}_0}) \\ &+ I(\mathbf{D}^{\text{Aug}_k},\mathbf{X}^{\text{Aug}_0}|\mathbf{X}^{\text{Aug}_k})\big].\end{aligned} \tag{17}$$

Here, β_1 and β_2 are the hyperparameters that control the importance of the mutual information term.

For the first term of Eq. (17), we introduce the variational approximation $q(\boldsymbol{l}|\boldsymbol{d}^{\text{Aug}_k})$ to $p(\boldsymbol{l}|\boldsymbol{d}^{\text{Aug}_k})$. Then by assuming that the Markov chain $\mathbf{L} \leftrightarrow \mathbf{X}^{\text{Aug}_k} \leftrightarrow \mathbf{D}^{\text{Aug}_k}$ holds, we can obtain an upper bound on the first term of Eq. (17):

$$\begin{aligned}-\beta_1 I(\mathbf{D}^{\text{Aug}_k},\mathbf{L}) \leq -\beta_1 \int \mathrm{d}(\text{Aug}_k(\boldsymbol{x}),\, \boldsymbol{l},\, \boldsymbol{d}^{\text{Aug}_k}) \\ p(\text{Aug}_k(\boldsymbol{x}),\boldsymbol{l})p(\boldsymbol{d}^{\text{Aug}_k}|\text{Aug}_k(\boldsymbol{x}))\log q(\boldsymbol{l}|\boldsymbol{d}^{\text{Aug}_k}).\end{aligned} \tag{18}$$

MILEN assumes that

$$p_{\boldsymbol{\kappa}}(\mathbf{D}^{\text{Aug}_k}|\mathbf{X}^{\text{Aug}_k}) = \prod_{i=1}^{n} Dir(\boldsymbol{d}_i^{\text{Aug}_k}|\boldsymbol{\alpha}_i^{\text{Aug}_k}), \tag{19}$$

where the parameters $\boldsymbol{\Delta}^{\text{Aug}_k} = [\boldsymbol{\alpha}_1^{\text{Aug}_k},\boldsymbol{\alpha}_2^{\text{Aug}_k},\ldots,\boldsymbol{\alpha}_n^{\text{Aug}_k}]$ are outputs of the model parameterized by $\boldsymbol{\kappa}$, and

$$q_{\boldsymbol{\nu}}(\mathbf{L}|\mathbf{D}^{\text{Aug}_k}) = \prod_{i=1}^{n} Gau(\boldsymbol{l}_i|\boldsymbol{\mu}_i^{\text{Aug}_k},\boldsymbol{I}), \tag{20}$$

where the parameters $\boldsymbol{\Lambda}^{\text{Aug}_k} = [\boldsymbol{\mu}_1^{\text{Aug}_k},\boldsymbol{\mu}_2^{\text{Aug}_k},\ldots,\boldsymbol{\mu}_n^{\text{Aug}_k}]$ are outputs of the model parameterized by $\boldsymbol{\nu}$. Note that the label distribution matrix of augmented instances $\mathbf{D}^{\text{Aug}_k}$ is sampled from $p(\mathbf{D}^{\text{Aug}_k}|\mathbf{X}^{\text{Aug}_k})$, i.e., $\boldsymbol{d}_i^{\text{Aug}_k} \sim Dir(\boldsymbol{\alpha}_i^{\text{Aug}_k})$, where the implicit reparameterization gradient [12] is also employed to allow the gradients to be computed analytically in backward pass.

For the second term of Eq. (17), we also introduce the variational approximation $q(\boldsymbol{d}^{\text{Aug}_k})$ to $p(\boldsymbol{d}^{\text{Aug}_k})$. Then we can obtain an upper bound on the second term of Eq. (17):

$$\begin{aligned}I(\mathbf{D}^{\text{Aug}_k},\mathbf{X}^{\text{Aug}_k}) \leq \int \mathrm{d}(\text{Aug}_k(\boldsymbol{x}),\, \boldsymbol{d}^{\text{Aug}_k}) \\ p(\text{Aug}_k(\boldsymbol{x}),\boldsymbol{d}^{\text{Aug}_k})\log\frac{p(\boldsymbol{d}^{\text{Aug}_k}|\text{Aug}_k(\boldsymbol{x}))}{q(\boldsymbol{d}^{\text{Aug}_k})}.\end{aligned} \tag{21}$$

Here, to optimize Eq. (21), we assume that $q(\mathbf{D}^{\text{Aug}_k}) = \prod_{i=1}^{n} Dir(\boldsymbol{d}_i^{\text{Aug}_k}|\hat{\boldsymbol{\alpha}})$.

For the third term of Eq. (17), we could define an infoNCE MI estimator, following the formulation of [34],

$$\begin{aligned}
&-\beta_2 I(\mathbf{D}^{\text{Aug}_k}, \mathbf{D}^{\text{Aug}_0}) \leq -\beta_2 \widehat{I}^{\text{infoNCE}}(\mathbf{D}^{\text{Aug}_k}, \mathbf{D}^{\text{Aug}_0}) \\
&:= \beta_2 \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\Big[\mathcal{B}(\boldsymbol{d}^{\text{Aug}_k}, \boldsymbol{d}^{\text{Aug}_0}) - \\
&\mathbb{E}_{\boldsymbol{x}'\sim p(\boldsymbol{x})}\Big[\log\sum_{\boldsymbol{x}'} e^{\mathcal{B}(\boldsymbol{d}^{\text{Aug}_k}, \boldsymbol{d}'^{\text{Aug}_0})}\Big]\Big],
\end{aligned} \tag{22}$$

where $\mathcal{B}$ is a function to calculate the cosine similarity.

For the fourth term of Eq. (17), we optimize the upper bound according to that the entropy is non-negative:

$$\begin{aligned}
&I(\mathbf{D}^{\text{Aug}_k}, \mathbf{X}^{\text{Aug}_0} | \mathbf{X}^{\text{Aug}_k}) = H(\mathbf{D}^{\text{Aug}_k} | \mathbf{X}^{\text{Aug}_k}) \\
&- H(\mathbf{D}^{\text{Aug}_k} | \mathbf{X}^{\text{Aug}_0}, \mathbf{X}^{\text{Aug}_k}) \\
&\leq H(\mathbf{D}^{\text{Aug}_k} | \mathbf{X}^{\text{Aug}_k}) \\
&= -\int \mathrm{d}(\text{Aug}_k(\boldsymbol{x}),\, \boldsymbol{d}^{\text{Aug}_k}) p(\text{Aug}_k(\boldsymbol{x})) \\
&p(\boldsymbol{d}^{\text{Aug}_k} | \text{Aug}_k(\boldsymbol{x})) \log p(\boldsymbol{d}^{\text{Aug}_k} | \text{Aug}_k(\boldsymbol{x})).
\end{aligned} \tag{23}$$

According to 18, 21, 22, 23, we can calculate the optimization objective related to mutual information in 17 as follows:

$$\begin{aligned}
&\mathcal{L}_{MI} = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=0}^{K}\Big[\frac{1}{2}\beta_1 \|\boldsymbol{\mu}_i^{\text{Aug}_k} - \boldsymbol{l}_i\|_2^2 \\
&+ \Big(\log\Gamma(\sum_{j=1}^{c}\alpha_i^{j,\text{Aug}_k}) - \sum_{j=1}^{c}\log\Gamma(\alpha_i^{j,\text{Aug}_k}) \\
&- \log\Gamma(c\cdot\varepsilon) + c\log\Gamma(\varepsilon) \\
&+ \sum_{j=1}^{c}(\alpha_i^{j,\text{Aug}_k} - \varepsilon)(\psi(\alpha_i^{j,\text{Aug}_k}) - \psi(\sum_{j=1}^{c}\alpha_i^{j,\text{Aug}_k}))\Big)\Big] \\
&- \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\Big[\beta_2 \log\frac{e^{\mathcal{B}(\boldsymbol{d}^{\text{Aug}_k}, \boldsymbol{d}^{\text{Aug}_0})}}{\sum_{\boldsymbol{x}'} e^{\mathcal{B}(\boldsymbol{d}^{\text{Aug}_k}, \boldsymbol{d}'^{\text{Aug}_0})}} \\
&+ \Big(\log\Gamma(\sum_{j=1}^{c}\alpha_i^{j,\text{Aug}_k}) - \sum_{j=1}^{c}\log\Gamma(\alpha_i^{j,\text{Aug}_k}) \\
&+ \sum_{j=1}^{c}(\alpha_i^{j,\text{Aug}_k} - 1)(\psi(\alpha_i^{j,\text{Aug}_k}) - \psi(\sum_{j=1}^{c}\alpha_i^{j,\text{Aug}_k}))\Big)\Big].
\end{aligned} \tag{24}$$

Similar to VALEN, MILEN enforces the recovered label distributions of each augmented data point not to be completely different from the confidence $\zeta(\text{Aug}_k(\boldsymbol{x}_i))$ [11], [31] estimated by current prediction $f(\text{Aug}_k(\boldsymbol{x}_i); \boldsymbol{\theta})$:

$$\mathcal{L}_o^{\text{Aug}} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c}\sum_{k=0}^{K}\zeta_j(\text{Aug}_k(\boldsymbol{x}_i))\log d_i^{y_j,\text{Aug}_k}, \tag{25}$$

where

$$\zeta_j(\text{Aug}_k(\boldsymbol{x}_i)) = \begin{cases} \frac{f_j(\text{Aug}_k(\boldsymbol{x}_i);\boldsymbol{\theta})}{\sum_{y_r\in S_i} f_r(\text{Aug}_k(\boldsymbol{x}_i);\boldsymbol{\theta})} & \text{if } y_j \in S_i \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

Then, we can formulate the MILEN optimization objective $\mathcal{L}_{LE}$ as follows:

$$\mathcal{L}_{LE} = \lambda\mathcal{L}_o^{\text{Aug}} - \mathcal{L}_{MI}, \tag{27}$$

where λ is a hyper-parameter.

Finally, MILEN trains the classifier in Subsection 3.4 by using the recovered label distributions. The algorithmic description of the MILEN is shown in Algorithm 2.

Algorithm 2 MILEN Algorithm

```
Input: The PLL training set $\mathcal{D} = \{(\boldsymbol{x}_i, S_i)\}_{i=1}^n$, epoch $T$ and iteration $I$;
Initialize the predictive model $\boldsymbol{\theta}$ by warm-up training, the mutual information models $\boldsymbol{\kappa}$ and $\boldsymbol{\nu}$;
for $t = 1, \dots, T$ do
    Augment training set $\mathcal{D} = \{(\boldsymbol{x}_i, S_i)\}_{i=1}^n$ into $\mathcal{D}^{\text{Aug}} = \{(\mathcal{A}(\boldsymbol{x}_i), S_i)\}_{i=1}^n$;
    Shuffle augmented training set $\mathcal{D}^{\text{Aug}} = \{(\mathcal{A}(\boldsymbol{x}_i), S_i)\}_{i=1}^n$ into $B$ mini-batches;
    for $b = 1, \dots, B$ do
        for $k = 0, \dots, K$ do
            Obtain label distribution $\boldsymbol{d}_i^{\text{Aug}_k}$ for each example $\boldsymbol{x}_i^{\text{Aug}_k}$ in $\mathcal{A}(\boldsymbol{x}_i)$ by Eq. (19);
        end for
        Update $\boldsymbol{\theta}$, $\boldsymbol{\kappa}$ and $\boldsymbol{\nu}$ by forward computation and back-propagation by fusing Eq. (27) and Eq. (28);
    end for
end for
Output: The predictive model $\boldsymbol{\theta}$.
```

3.4 Classifier Training

To train the predictive model, we minimize the following empirical risk estimator by levering the recovered label distributions:

$$\widehat{R}_V(f) = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{y_j\in S_i}\frac{d_i^{y_j}}{\sum_{y_j\in S_i} d_i^{y_j}}\ell(f(\boldsymbol{x}_i), \boldsymbol{e}^{y_j})\right). \tag{28}$$

Here we adopt the average value of $\boldsymbol{d}_i$ sampled by $\boldsymbol{d}_i \sim Dir(\boldsymbol{\alpha}_i)$. We can use any deep neural network as the predictive model, and then equip it with the VALEN or MILEN framework to deal with PLL. VALEN trains the predictive model and updates the label distributions in a principled end-to-end manner by fusing the objective Eq. (16) and Eq. (28). Besides, MILEN trains the predictive model and updates the label distributions in a principled end-to-end manner by fusing the objective Eq. (27) and Eq. (28).

Let $\widehat{f}_V = \min_{f\in\mathcal{F}}\widehat{R}_V(f)$ be the empirical risk minimizer and $f^\star = \min_{f\in\mathcal{F}} R_V(f)$ be the optimal risk minimizer where $R_V(f)$ is the risk estimator. Besides, we define the function space $\mathcal{H}_{y_j}$ for the label $y_j \in \mathcal{Y}$ as $\{h : \boldsymbol{x} \mapsto f_{y_j}(\boldsymbol{x}) \mid f \in \mathcal{F}\}$. Let $\mathfrak{R}_n(\mathcal{H}_{y_j})$ be the expected Rademacher complexity [2] of $\mathcal{H}_{y_j}$ with sample size n, then we have the following theorem.

Theorem 1. Assume the loss function $\ell(f(\boldsymbol{x}), \boldsymbol{e}^{y_j})$ is L-Lipschitz with respect to $f(\boldsymbol{x})(0 < L < \infty)$ for all $y_j \in \mathcal{Y}$ and upper-bounded by M, i.e., $M = \sup_{x\in\mathcal{X}, f\in\mathcal{F}, y_j\in\mathcal{Y}}\ell(f(\boldsymbol{x}), \boldsymbol{e}^{y_j})$. Then, for any $\delta > 0$, with probability at least $1 - \delta$,

$$R\left(\widehat{f}_V\right) - R(f^\star) \leq 4\sqrt{2}L\sum_{j=1}^{c}\mathfrak{R}_n\left(\mathcal{H}_{y_j}\right) + M\sqrt{\frac{\log\frac{2}{\delta}}{2n}}.$$

The proof of Theorem 1 is provided in Appendix. Theorem 1 shows that the empirical risk minimizer f_V converges to the optimal risk minimizer $f^\star$ as $n \to \infty$ and $\mathfrak{R}_n\left(\mathcal{H}_{y_j}\right) \to 0$ for all parametric models with a bounded norm. Note that we pre-limit the predictive model f by clamping their output to $[-A, A]$, and thus the loss function will be bounded. In practice, with the assistance of Pytorch, this operation is implemented by the `torch.clamp` function applied to the last linear layer of our neural network. Then, we

TABLE 1
Characteristic of the benchmark datasets.

Dataset	#Train	#Validation	#Test	#Features	#Class Labels	avg. #CLs
MNIST	54,000	6000	10,000	784	10	8.71
Fashion-MNIST	54,000	6000	10,000	784	10	3.46
Kuzushiji-MNIST	54,000	6000	10,000	784	10	3.87
CIFAR-10	45,000	5000	10,000	3,072	10	3.68
CIFAR-100	45,000	5000	10,000	3,072	100	4.64

TABLE 2
Characteristic of the real-world PLL datasets.

Dataset	#Train	#Validation	#Test	#Features	#Class Labels	avg. #CLs	Task Domain
Lost	898	112	112	108	16	2.23	*automatic face naming* [7]
MSRCv2	1,406	176	176	48	23	3.16	*object classification* [29]
BirdSong	3,998	500	500	38	13	2.18	*bird song classification* [3]
Mirflickr	2224	278	278	48	23	3.16	*web image classification* [19]
Malagasy	4250	531	531	38	13	2.18	*POS Tagging* [13]
Soccer Player	13,978	1747	1747	279	171	2.09	*automatic face naming* [51]
Yahoo! News	18,393	2299	2299	163	219	1.91	*automatic face naming* [16]

prove that the loss function ℓ could be bounded by $M = 2A + \log c$ as follows:

$$\begin{aligned} |\ell(f(\boldsymbol{x}_i), \boldsymbol{e}^{y_j})| &= |\log \frac{e^{f_j(\boldsymbol{x}_i)}}{\sum_{k=1}^{c} e^{f_k(\boldsymbol{x}_i)}}| \\ &= |f_j(\boldsymbol{x}_i) + \log \sum_{k=1}^{c} e^{f_k(\boldsymbol{x}_i)}| \\ &\leq A + |\log ce^{A}| \\ &= M \end{aligned} \tag{29}$$

In this way, our theoretical analysis becomes more solid. Fortunately, when we set A to a very large value (such as 10^{12}, which is larger than $\max_{i,j} f_j(\boldsymbol{x}_i)$ in each epoch), the clamping operation does not affect the whole algorithm, and the related experimental results are unchanged with the same random seeds.

According to [32], we let the empirical risk with the Bayes class-probability distribution $\boldsymbol{p} = [P(y_1|\boldsymbol{x}), P(y_2|\boldsymbol{x}), \ldots, P(y_c|\boldsymbol{x})]$ be denoted by

$$\widehat{R}_V^{\star}(f) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{c} P(y_j|\boldsymbol{x}) \ell(f(\boldsymbol{x}, \boldsymbol{e}^{y_j})). \tag{30}$$

Then, we have the following theorem:

Theorem 2. Suppose the loss function ℓ is bounded by M, i.e., $M = \sup_{\boldsymbol{x} \in \mathcal{X}, f \in \mathcal{F}, y_j \in \mathcal{Y}} \ell(f(\boldsymbol{x}), y)$. Fix a hypothesis class $\mathcal{F}$ of predictors $f \colon \mathcal{X} \mapsto \mathbb{R}^c$, with induced class $\mathcal{H} \subset [0,1]^{\mathcal{X}}$ of functions $h(\boldsymbol{x}) = \sum_{y_j \in S_i} \frac{d_i^{y_j}}{\sum_{y_j \in S_i} d_i^{y_j}} \ell(f(\boldsymbol{x}_i), \boldsymbol{e}^{y_j})$. Suppose $\mathcal{H}$ has uniform covering number $\mathcal{N}_{\inf}$. Then for any $\delta \in (0,1)$, with probability at least $1 - \delta$,

$$\begin{aligned} R_V(f) - \widehat{R}_V(f) \leq & M\sqrt{c} \cdot (\mathbb{E}[||\boldsymbol{q} - \boldsymbol{p}||_2]) \\ & + \mathcal{O}\left(\sqrt{\mathbb{V}(f) \cdot \frac{\log \frac{\mathcal{M}_n}{\delta}}{N}} + \frac{\log \frac{\mathcal{M}_n}{\delta}}{N} \right), \end{aligned}$$

where $\mathcal{M}_N = \mathcal{N}_{\inf}(\frac{1}{N}, \mathcal{H}, 2n)$, $\mathbb{V}(f)$ is the empirical variance of the loss values, and $\boldsymbol{q} = \frac{\boldsymbol{d} \cdot \boldsymbol{l}}{\boldsymbol{d}^{\top} \boldsymbol{l}}$ is the estimated label distribution normalized over candidate labels.

The proof of Theorem 2 is provided in Appendix. Theorem 2 shows that the expected risk will decrease as the normalized estimated label distribution $\boldsymbol{q}$ is approximate to the Bayes class-probability distribution $\boldsymbol{p}$.

4 Experiments

4.1 Datasets

In our study, we evaluate the proposed method using five well-established benchmark datasets including `MNIST` [26], `Kuzushiji-MNIST` [6], `Fashion-MNIST` [42], `CIFAR-10`, and `CIFAR-100` [25].

We create partially labeled versions of the benchmark datasets by manually corrupting the original clean labels. Firstly, the ground-truth label of each example is selected into the corresponding candidate label set. Then we introduce a flipping probability $\xi_i^{y_j}$ for each incorrect label $l_i^{y_j}$ corresponding to an example $\boldsymbol{x}_i$. The flipping probability $\xi_i^{y_j}$ represents the probability that the incorrect label $l_i^{y_j}$ will be flipped to the candidate label during the corruption process.

To synthesize the instance-dependent candidate labels, we utilize the confidence predictions of a clean neural network $\hat{\boldsymbol{\theta}}$, which has been trained using the original clean labels [56]. Specifically, we calculate the flipping probability $\xi_i^{y_j}$ for each incorrect label $l_i^{y_j}$ as $\xi_i^{y_j} = \frac{f_j(\boldsymbol{x}_i; \hat{\boldsymbol{\theta}})}{\max_{y_j \in \bar{Y}_i} f_j(\boldsymbol{x}_i; \hat{\boldsymbol{\theta}})}$, where $f_j(\boldsymbol{x}_i; \hat{\boldsymbol{\theta}})$ is the confidence prediction score of the clean neural network $\hat{\boldsymbol{\theta}}$ for the incorrect label $l_i^{y_j}$, and $\bar{Y}_i$ represents the set of all incorrect labels for the example $\boldsymbol{x}_i$. The detailed descriptions of these corrupted benchmark datasets are provided in Table 1.

Additionally, we include seven real-world PLL datasets that have been collected from various application domains. These datasets cover a range of tasks, including `Lost` [7], `Soccer Player` [51], and `Yahoo!News` [16] for automatic face naming from images or videos, `MSRCv2` [29] for object classification, `BirdSong` [3] for bird song classification, `Malagasy` [13] for POS tagging, and `Mirflickr` [19] for web image classification. The detailed descriptions of these datasets are provided in Table 2.

TABLE 3
Classification accuracy (mean±std) of each comparing approach on benchmark datasets.

	MNIST	Fashion-MNIST	Kuzushiji-MNIST	CIFAR-10	CIFAR-100
MILEN	$\mathbf{98.96 \pm 0.07}\%$	$\mathbf{90.96 \pm 0.11}\%$	$\mathbf{96.32 \pm 0.54}\%$	$\mathbf{86.92 \pm 0.34}\%$	$\mathbf{64.97 \pm 0.39}\%$
VALEN	$98.72 \pm 0.05\%$●	$90.63 \pm 0.30\%$	$96.19 \pm 0.75\%$	$85.48 \pm 0.62\%$●	$62.96 \pm 0.96\%$●
PLCR	$98.56 \pm 0.08\%$●	$90.10 \pm 0.21\%$●	$95.29 \pm 0.21\%$●	$86.37 \pm 0.38\%$●	$64.12 \pm 0.23\%$●
PICO	$98.61 \pm 0.12\%$●	$88.41 \pm 0.20\%$●	$94.78 \pm 0.19\%$●	$86.16 \pm 0.21\%$●	$62.98 \pm 0.38\%$●
CAVL	$98.84 \pm 0.05\%$●	$87.94 \pm 0.19\%$●	$93.69 \pm 0.28\%$●	$59.67 \pm 3.30\%$●	$52.59 \pm 1.01\%$●
LWS	$98.56 \pm 0.06\%$●	$88.99 \pm 0.26\%$●	$92.27 \pm 1.03\%$●	$37.49 \pm 2.82\%$●	$53.98 \pm 0.99\%$●
RC	$98.41 \pm 0.09\%$●	$89.60 \pm 0.19\%$●	$93.78 \pm 0.17\%$●	$85.95 \pm 0.40\%$●	$63.41 \pm 0.56\%$●
CC	$98.16 \pm 0.14\%$●	$89.86 \pm 0.11\%$●	$94.08 \pm 0.35\%$●	$79.96 \pm 0.99\%$●	$62.40 \pm 0.84\%$●
PRODEN	$98.39 \pm 0.10\%$●	$89.79 \pm 0.24\%$●	$93.79 \pm 0.24\%$●	$86.04 \pm 0.21\%$●	$62.56 \pm 1.49\%$●
D2CNN	$97.76 \pm 0.14\%$●	$87.81 \pm 0.20\%$●	$91.24 \pm 0.24\%$●	$69.22 \pm 1.56\%$●	$32.42 \pm 0.61\%$●

TABLE 4
Classification accuracy (mean±std) of each comparing approach on the real-world datasets.

	Lost	MSRCv2	BirdSong	Mirflickr	Malagasy	Soccer Player	Yahoo!News
VALEN	$\mathbf{76.87 \pm 0.86}\%$	$49.97 \pm 0.43\%$	$\mathbf{73.75 \pm 0.39}\%$	$\mathbf{60.39 \pm 0.55}\%$	$\mathbf{71.28 \pm 0.45}\%$	$\mathbf{55.81 \pm 0.10}\%$	$\mathbf{66.53 \pm 0.20}\%$
CAVL	$73.96 \pm 0.51\%$●	$46.62 \pm 1.29\%$●	$69.63 \pm 0.93\%$●	$57.13 \pm 0.10\%$●	$65.82 \pm 0.06\%$●	$52.92 \pm 0.40\%$●	$60.97 \pm 0.13\%$●
LWS	$73.13 \pm 0.32\%$●	$49.85 \pm 0.49\%$	$51.45 \pm 0.26\%$●	$54.50 \pm 0.81\%$●	$59.34 \pm 0.25\%$●	$50.24 \pm 0.45\%$●	$48.21 \pm 0.29\%$●
RC	$76.26 \pm 0.46\%$	$49.47 \pm 0.43\%$	$69.33 \pm 0.32\%$●	$58.93 \pm 0.10\%$●	$70.69 \pm 0.14\%$●	$56.02 \pm 0.59\%$●	$63.51 \pm 0.20\%$●
CC	$63.54 \pm 0.25\%$●	$41.50 \pm 0.44\%$●	$69.90 \pm 0.58\%$●	$58.81 \pm 0.54\%$●	$69.53 \pm 0.34\%$●	$49.07 \pm 0.36\%$●	$54.86 \pm 0.48\%$●
PRODEN	$76.47 \pm 0.25\%$	$45.10 \pm 0.16\%$●	$73.44 \pm 0.12\%$●	$59.59 \pm 0.52\%$●	$69.34 \pm 0.09\%$●	$54.05 \pm 0.15\%$●	$66.14 \pm 0.10\%$●
D2CNN	$49.91 \pm 2.33\%$●	$38.58 \pm 2.76\%$●	$63.56 \pm 2.81\%$●	$58.45 \pm 1.65\%$●	$54.55 \pm 4.85\%$●	$49.26 \pm 0.21\%$●	$49.62 \pm 0.64\%$●
CLPL	$63.39 \pm 0.12\%$●	$37.80 \pm 0.71\%$●	$62.90 \pm 3.33\%$●	$58.87 \pm 0.10\%$●	$64.25 \pm 0.29\%$●	$48.23 \pm 0.03\%$●	$49.42 \pm 0.13\%$●
PL-SVM	$61.51 \pm 4.00\%$●	$34.62 \pm 3.77\%$●	$47.51 \pm 3.94\%$●	$43.73 \pm 8.88\%$●	$55.07 \pm 6.75\%$●	$40.37 \pm 2.92\%$●	$50.58 \pm 0.74\%$●
PL-KNN	$38.76 \pm 2.32\%$●	$39.55 \pm 3.50\%$●	$54.01 \pm 1.48\%$●	$50.31 \pm 1.83\%$●	$58.04 \pm 1.69\%$●	$48.61 \pm 0.67\%$●	$46.22 \pm 0.76\%$●
IPAL	$64.53 \pm 4.19\%$●	$\mathbf{50.03 \pm 1.93}\%$	$58.60 \pm 1.26\%$●	$56.27 \pm 3.00\%$●	$62.54 \pm 0.37\%$●	$53.60 \pm 0.91\%$●	$57.64 \pm 0.68\%$●
PLLE	$69.07 \pm 2.08\%$●	$48.73 \pm 2.13\%$	$70.65 \pm 1.09\%$●	$49.91 \pm 2.65\%$●	$60.89 \pm 2.50\%$●	$49.10 \pm 1.20\%$●	$54.28 \pm 0.44\%$●

For benchmark datasets, we split 10% samples from the training datasets for validation. For each real-world dataset, we run the methods with $80\%/10\%/10\%$ train/validation/test split. Then we run five trials on each dataset with different random seeds and report the mean accuracy and standard deviation of all comparing algorithms.

4.2 Baselines

The performance of VALEN and MILEN is compared against eight DNN-based approaches:

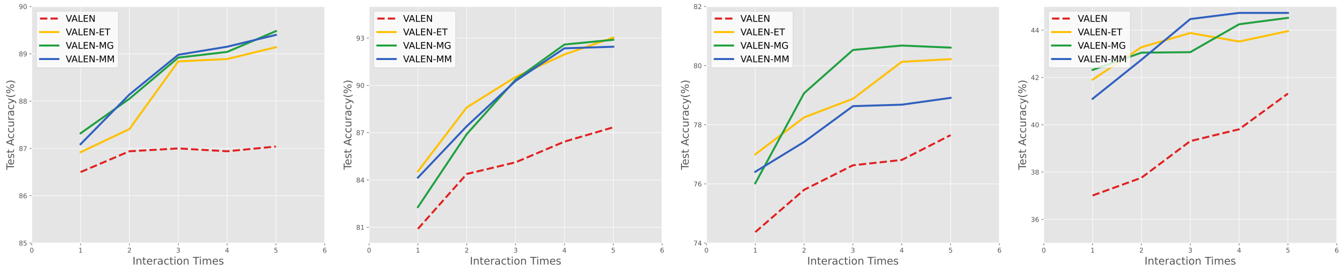
- PLCR [41], a regularized training framework which is based on data augmentation and utilizes the manifold consistency regularization term to preserve the manifold structure both in feature space and label space.
- PICO [39], a contrastive learning framework which is based on data augmentation and performs label disambiguation based on the contrastive prototypes.
- CAVL [52], a discriminative approach which identifies correct labels from candidate labels by class activation value.
- LWS [40], an identification-based method which introduces a leverage parameter to consider the trade-off between losses on candidate and non-candidate labels.
- RC [11]: A risk-consistent partial label learning approach which employs the importance reweighting strategy to converge the true risk minimizer.
- CC [11]: A classifier-consistent partial label learning approach which uses a transition matrix to form an empirical risk estimator.
- PRODEN [31]: A progressive identification partial label learning approach which approximately minimizes a risk estimator and identifies the true labels in a seamless manner.
- D2CNN [49]: A deep partial label learning approach which designs an entropy-based regularizer to maximize the margin between the potentially correct label and the unlikely ones.

For the benchmark datasets, we use the same data augmentation strategy for the data-augmentation-free methods (VALEN, PRODEN, RC, CC, LWS and CAVL) to make fair comparisons with the data-augmentation-based methods (MILEN, PICO and PLCR). We use three augmentations ($K = 3$) including one weak and two strong augmentations. However, data augmentation cannot be employed on the real-world datasets that only contain extracted features from audio and video data. Therefore, we just compare the data-augmentation-free methods on the real-world datasets.

For all the DNN-based approaches, we adopt the same predictive model for fair comparisons. Specifically, a five-layer LeNet is trained on `MNIST`, `Fashion-MNIST`, and `Kuzushiji-MNIST`. The 32-layer ResNet is trained on `CIFAR-10` and `CIFAR-100`. For real-world datasets, we adopt the linear model. The hyperparameters are selected to maximize the accuracy of a validation dataset. With the batch size set to 256, the number of epochs is set to 250, during which the first 10 epochs are dedicated to warm-up training. The model training will be stopped early if its performance on the validation dataset does not improve in 50 epochs. All the DNN-based approaches are implemented with PyTorch.

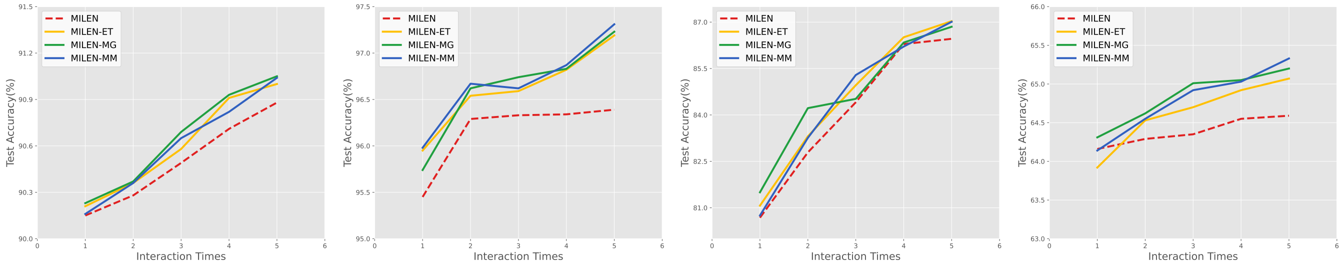
In addition, we also compare five classical partial label learning approaches, each configured with parameters suggested in respective literature:

- CLPL [7]: A convex partial label learning approach which uses averaging-based disambiguation, where the proposed loss function is asymptotically consistent, as well as its generalization and transductive performance.



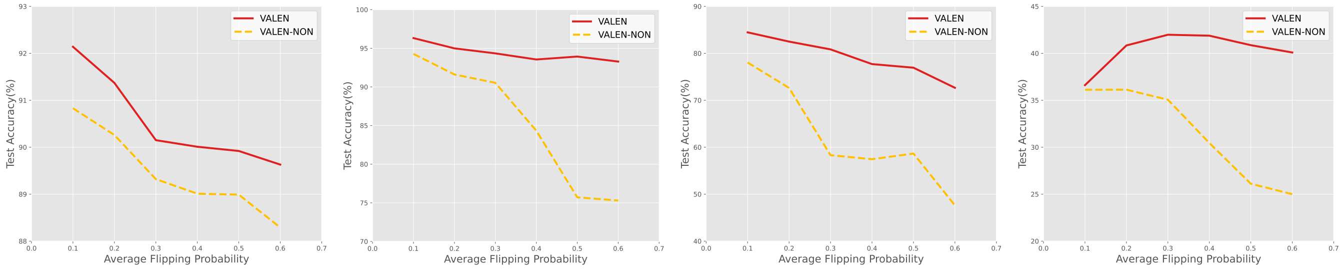
(a) Fashion-MNIST (b) Kuzushiji-MNIST (c) CIFAR-10 (d) CIFAR-100

Fig. 3. Active learning performance of VALEN on Fashion-MNIST, Kuzushiji-MNIST, CIFAR-10 and CIFAR-100.



(a) Fashion-MNIST (b) Kuzushiji-MNIST (c) CIFAR-10 (d) CIFAR-100

Fig. 4. Active learning performance of MILEN on Fashion-MNIST, Kuzushiji-MNIST, CIFAR-10 and CIFAR-100.



(a) Fashion-MNIST (b) Kuzushiji-MNIST (c) CIFAR-10 (d) CIFAR-100

Fig. 5. Ablation studies of VALEN on Fashion-MNIST, Kuzushiji-MNIST, CIFAR-10 and CIFAR-100.

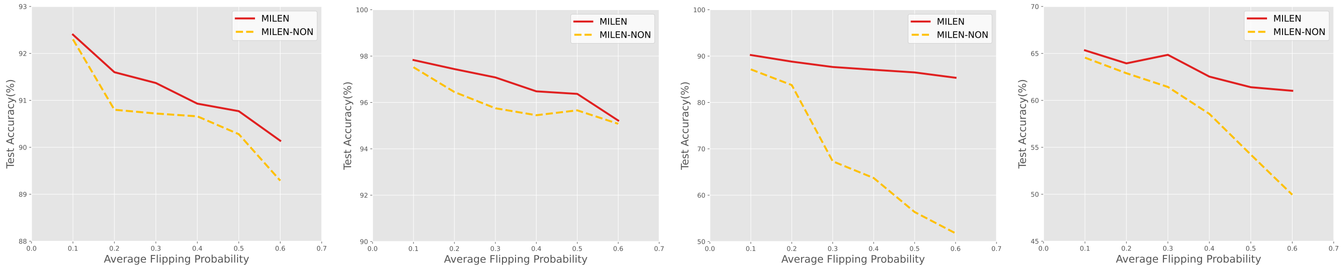
- PL-KNN [20]: An instance-based partial label learning approach which works by k-nearest neighbor weighted voting.
- PL-SVM [33]: A maximum margin partial label learning approach by incorporating partial label information into the conventional margin-based learning framework.
- IPAL [54]: A non-parametric method that applies the label propagation strategy to iteratively update the confidence of each candidate label.
- PLLE [44]: A two-stage partial label learning approach which estimates the generalized description degree of each class label values via graph Laplacian and induces a multi-label predictive model with the generalized description degree in separate stages.

4.3 Experimental Results

Table 3 reports the classification accuracy of each DNN-based method on benchmark datasets corrupted by the instance-dependent generating procedure. The best results are highlighted in bold. In addition, ● / ○ indicates whether MILEN is statistically superior/inferior to the comparing approach on each dataset (pairwise t-test at 0.05 significance level). From the table, we can observe that MILEN always achieves the best performance and significantly outperforms other compared methods in most cases.

Table 4 reports the experimental results on the real-world PLL datasets. Note that data augmentation (the foundation in MILEN, PICO and PLCR) cannot be employed on the real-world datasets that contain extracted features from audio and video data, we just compared VALEN with the data-augmentation-free methods on the real-world datasets. We can find that VALEN achieves the best performance against other DNN-based methods on the real-world PLL datasets. Note that VALEN achieves the best performance against classical methods on all datasets except MSRCv2 as these datasets are small-scale and the average number of candidate labels in each dataset is low (can be seen in Table 2), which leads to the result that DNN-based methods cannot take full advantage.

The performance of MILEN surpasses that of VALEN, primarily because MILEN capitalizes on the mutual information inherent in augmented data. However, the efficacy of MILEN is contingent upon the applicability of data augmentation techniques, which are not universally applicable across all data types. For instance,



(a) `Fashion-MNIST` (b) `Kuzushiji-MNIST` (c) `CIFAR-10` (d) `CIFAR-100`

Fig. 6. Ablation studies of Milen on `Fashion-MNIST`, `Kuzushiji-MNIST`, `CIFAR-10` and `CIFAR-100`.

tabular data often cannot be enhanced through data augmentation techniques, limiting the versatility of Milen. In contrast, Valen exhibits greater adaptability and can be applied to a broader spectrum of data types. Consequently, when dealing with datasets amenable to augmentation, such as images, Milen emerges as the preferred choice for training predictive models. In scenarios where data augmentation is not feasible, Valen stands out as the more appropriate option for model training.

4.4 Active Learning Performance

Compared to the typical PLL problem, instance-dependent PLL is more challenging to disambiguate the candidate labels. Therefore, motivated by Wang [38], we consider the particular case in which the classifier is allowed to selectively query examples in the training dataset for manual disambiguation. In practice, the performance of the classifier could be improved through active interaction between humans and the classifier via adopting the paradigm of active learning [18], [47]. Specifically, the classifier is first trained with the original instance-dependent partial labels. Then a selector chooses the examples with high uncertainty in the training datasets and asks an oracle (human expert) to annotate them. Finally, the examples with high uncertainty from the selector with the annotation from experts are re-applied to the training to improve the performance of the classifier. In this way, the classifier finishes one active interaction with the oracle.

Furthermore, effective criteria should be set on the selector to actively mine the examples with high uncertainty for more precise annotation. Given the training sample $(\boldsymbol{x}_i, S_i)$, we employ three frequently-used ways [38] to achieve its uncertainty score $U(\boldsymbol{x}_i, S_i)$ using the recovered distributions:

- *Entropy-based*:
  $U_{ET}(\boldsymbol{x}_i, S_i) = -\sum_{j=1}^{c} d_i^j \log d_i^j$;
- *Margin-based*:
  $U_{MG}(\boldsymbol{x}_i, S_i) = d_i^o - d_i^m$, where $m = \arg\max_{j \in S_i} d_i^j$ and $o = \arg\max_{j \in S_i, j \neq m} d_i^j$;
- *Maximum-based*:
  $U_{MM}(\boldsymbol{x}_i, S_i) = 1 - \max_{j \in S_i} d_i^j$.

Valen and Milen are evaluated in the following setting of active learning mentioned above. We set the number of training epochs $T = 500$, and fix the times of active interaction $s = 10$, which means the interaction is performed every 50 epochs. The selector mines a the example set with high uncertainty $\{\boldsymbol{x}_{a_1}, \boldsymbol{x}_{a_1}, \ldots, \boldsymbol{x}_{a_v}\}$ $(v = 3n/100)$ each interaction, which means at most 30% of training examples are queried during the training process. The candidate label set S_{a_i} of each selected example is replaced by the oracle with the correct label $y_{\boldsymbol{x}_{a_i}}$, i.e., the logical label vector $\boldsymbol{l}_{a_i}$ will be one-hot with $l_{a_i}^{y_{\boldsymbol{x}_{a_i}}} = 1$ after interaction. Valen and Milen are equipped with entropy-based, margin-based, maximum-based and random-based selectors to form Valen-Et, Valen-Mg, Valen-Mm, Valen-Rd and Milen-Et, Milen-Mg, Milen-Mm, Milen-Rd, respectively.

Figure 3 and Figure 4 illustrate the respective performance of Valen on and Milen on `Fashion-MNIST`, `Kuzushiji-MNIST`, `CIFAR-10` and `CIFAR-100` by plotting the accuracy curves of Valen and Milen at each interaction time equipped with different selectors. We can observe that on all these datasets Valen and Milen with active learning perform better than Valen and Milen without active learning, which is a nice property for those that require human interaction to improve the designed algorithm.

4.5 Further Analysis

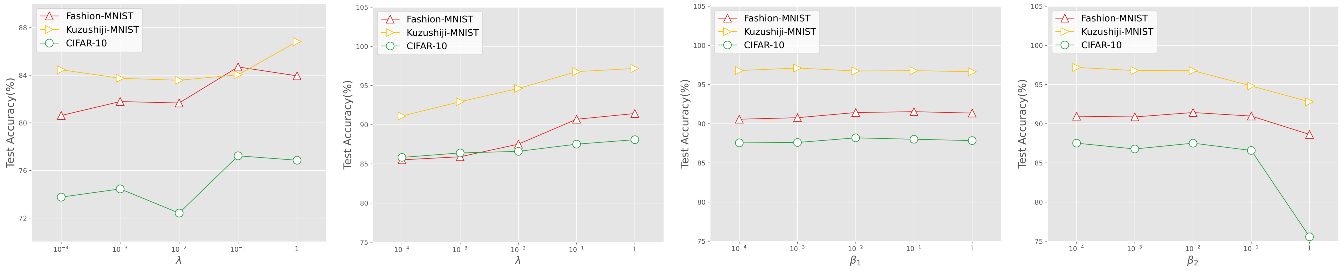
4.5.1 Ablation Studies

Figure 5(a) - 5(d) illustrate the performance of Valen on `Fashion-MNIST`, `CIFAR-10` and `CIFAR-100` corrupted by the instance-dependent generating procedure under different flipping probability, while Figure 6(a) - 6(d) illustrate the performance of Milen on `Fashion-MNIST`, `CIFAR-10` and `CIFAR-100` corrupted by the instance-dependent generating procedure under different flipping probability. Besides, the performance of the ablation version that removes the label enhancement and trains the predictive model with PLL minimal loss (denoted by Valen-non and Milen-non) is recorded. These results clearly validate the usefulness of recovered label distributions for improving predictive performance.

4.5.2 The Impact of Warm-Up

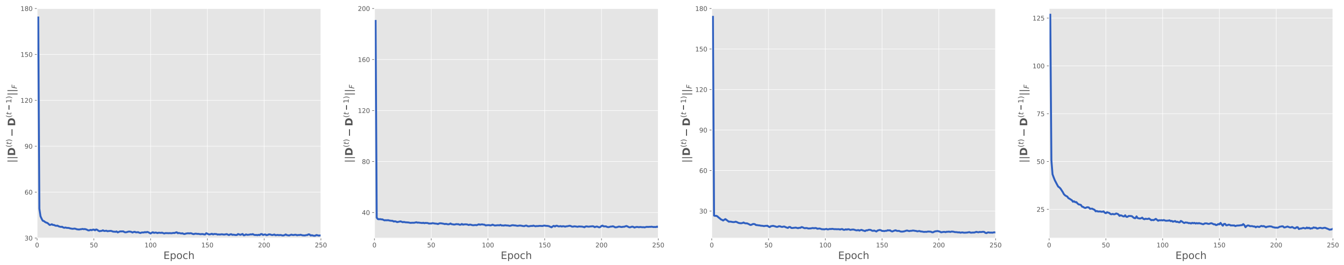
Table 5 illustrates the performance of Valen and Milen on `CIFAR-10` corrupted by the instance-dependent PLL generating procedure under different epochs of warm-up. Besides, the performance of Valen and Milen with the warm-up process adopted by multi-label loss (denoted by Valen-ml and Milen-ml) and consistent loss PLL (denoted by Valen-cl and Milen-cl) is recorded.

When using multi-label loss or consistency loss to replace Proden loss [31] in the warm-up training, Valen and Milen seem to have little improvement. This is because Proden loss could enable the model to have a good discrimination ability to progressively identify correct labels thanks to the memorization effects of neural networks. However, multi-label loss treats all candidate labels equally, leading to interference from incorrect candidate labels and weakening the model's discrimination ability



(a) varying λ in VALEN (b) varying λ in MILEN (c) varying β_1 in MILEN (d) varying β_2 in MILEN

Fig. 7. Performance of VALEN and MILEN changes as their hyper-parameters vary on `Fashion-MNIST`, `Kuzushiji-MNIST`, `CIFAR-10`.



(a) VALEN, `Fashion-MNIST` (b) VALEN, `Kuzushiji-MNIST` (c) MILEN, `Fashion-MNIST` (d) MILEN, `Kuzushiji-MNIST`

Fig. 8. Convergence curves of the recovered label distribution $\mathbf{D}$ in VALEN and MILEN on `Fashion-MNIST` and `Kuzushiji-MNIST`.

TABLE 5
Classification accuracy (mean ± std) of VALEN and MILEN on different warm-up loss functions and epochs, where 'w/o warm-up' denotes the performance of VALEN and MILEN trained without warm-up.

Warm-up Epochs	1	5	10	20	30	w/o warm-up
VALEN	$85.50 \pm 0.58\%$	$85.85 \pm 0.49\%$	$85.48 \pm 0.48\%$	$85.79 \pm 0.30\%$	$85.88 \pm 0.20\%$	$85.45 \pm 0.48\%$
VALEN-ML	$85.74 \pm 0.47\%$	$85.83 \pm 0.39\%$	$85.43 \pm 0.51\%$	$85.51 \pm 0.60\%$	$85.12 \pm 0.44\%$	
VALEN-CL	$85.47 \pm 0.51\%$	$85.26 \pm 0.32\%$	$85.39 \pm 0.61\%$	$85.58 \pm 0.50\%$	$85.67 \pm 0.35\%$	
MILEN	$86.89 \pm 0.49\%$	$86.58 \pm 0.34\%$	$86.92 \pm 0.34\%$	$87.33 \pm 0.34\%$	$87.19 \pm 0.16\%$	$83.22 \pm 0.76\%$
MILEN-ML	$86.52 \pm 0.63\%$	$86.76 \pm 0.48\%$	$87.27 \pm 0.23\%$	$87.04 \pm 0.35\%$	$87.12 \pm 0.45\%$	
MILEN-CL	$85.40 \pm 1.56\%$	$85.93 \pm 0.31\%$	$86.81 \pm 0.46\%$	$86.62 \pm 0.65\%$	$86.54 \pm 0.49\%$	

TABLE 6
Quality of label distributions (mean ± std) estimated by our proposed algorithms and compared baselines on the label distribution datasets, measured by $\mathbb{E}[||\boldsymbol{q} - \boldsymbol{p}||_2]$. The best and second best performance among all the approaches are denoted by boldface and underline.

Datasets	RAF-ML	Twitter_LDL	Flickr_LDL
MILEN	$\mathbf{0.0226 \pm 0.0003}$	$\mathbf{0.0175 \pm 0.0002}$	$\mathbf{0.0212 \pm 0.0003}$
VALEN	$\underline{0.0270 \pm 0.0003}$	$\underline{0.0191 \pm 0.0002}$	$\underline{0.0229 \pm 0.0003}$
PLCR	0.0434 ± 0.0068	0.0553 ± 0.0003	0.0437 ± 0.0004
LALO	0.0451 ± 0.0061	0.0238 ± 0.0001	0.0313 ± 0.0003

[31]. Additionally, consistent loss aligns the model outputs of k augmented images with the inferred conformal label distribution, which may degrade the performance due to the low quality of inferred conformal label distribution in the beginning stage [41].

Note that MILEN is more reliant on the warm-up process than VALEN. MILEN considers estimating label distributions from k augmented instance matrices and further capture the relationship between them via mutual information according to the optimization objective in Eq. (17), which is more complicated and demands a better model to launch the optimization process. Therefore, through appropriate warm-up training, MILEN can fully leverage its advantages of utilizing multiple augmented views and capturing mutual information, reflecting that it more effectively utilizes data augmentation and mutual information estimation to improve the model's performance.

4.5.3 Sensitivity Analysis

The sensitivity of VALEN with respect to its parameter λ and MILEN with respect to its parameters λ, β_1 and β_2 is studied. Fig. 7 illustrates the performance of VALEN and MILEN under different parameter configurations on `Fashion-MNIST`, `Kuzushiji-MNIST` and `CIFAR-10`. As is shown in Fig.7, the performance of VALEN and MILEN is relatively stable as the value of the parameter λ changes within a reasonable even broad range.

TABLE 7
Classification accuracy (mean±std) with different choices of K strong augmentations on `Fashion-MNIST`, `Kuzushiji-MNIST` and `CIFAR-10`.

	Fashion-MNIST	Kuzushiji-MNIST	CIFAR-10
$K=1$	89.94 ± 0.08	93.71 ± 0.32	84.67 ± 0.08
$K=2$	90.66 ± 0.11	94.85 ± 0.10	86.32 ± 0.49
$K=3$	90.83 ± 0.22	95.59 ± 0.15	86.87 ± 0.53
$K=4$	90.87 ± 0.33	96.33 ± 0.23	87.07 ± 1.56
$K=5$	90.96 ± 0.17	96.42 ± 0.14	87.13 ± 0.52
$K=6$	$\mathbf{91.17 \pm 0.04}$	$\mathbf{96.50 \pm 0.15}$	$\mathbf{87.19 \pm 0.53}$

TABLE 8
Classification accuracy (mean±std) with different augmentations when $K=3$ on `Fashion-MNIST`, `Kuzushiji-MNIST` and `CIFAR-10`.

	Fashion-MNIST	Kuzushiji-MNIST	CIFAR-10
3W	90.48 ± 0.22	94.47 ± 0.73	82.55 ± 0.41
2W+1S	90.75 ± 0.17	95.66 ± 0.12	86.32 ± 0.49
1W + 2S	$\mathbf{90.96 \pm 0.11}$	$\mathbf{96.32 \pm 0.54}$	$\mathbf{86.92 \pm 0.34}$
3S	90.83 ± 0.22	95.59 ± 0.15	86.87 ± 0.53

For example, VALEN is quite insensitive when λ varying between $[1e-4, 1e-1]$ in `Fashion-MNIST`. So does β_1 and β_2 of MILEN varying between $[1e-4, 1e-1]$. This stability indicates VALEN and MILEN could be robustness, which is desirable for algorithm design.

4.5.4 Quality of the Recovered Label Distribution

Three label distribution datasets `REF-ML` [28], `Twitter_LDL` [48] and `Flickr_LDL` [48] are adopted in the experiment, where the ground-truth label distribution could be regarded as the Bayes class-probability distribution $\boldsymbol{p}$. The instance-dependent candidate labels are generated as in Section 4.1, where we replace $f_j(\boldsymbol{x}_i; \hat{\boldsymbol{\theta}})$ with p_i^j to calculate the flipping probability. A DNN-based PLL baseline PLCR [41] and a classical PLL baseline LALO [10] are adopted as comparing methods, both of which could estimate label distributions on PLL data during training. For the classic PLL baseline LALO, we extract features by employing pre-trained `Resnet-18` provided by `Pytorch`. The normalized label distribution $\boldsymbol{q}$ is recovered from the datasets with candidate labels by VALEN, MILEN and other comparing algorithms, and then the term $\mathbb{E}[||\boldsymbol{q}-\boldsymbol{p}||_2]$ could be calculated with the ground-truth label distributions. Table 6 illustrates that $\mathbb{E}[||\boldsymbol{q}-\boldsymbol{p}||_2]$ of our proposed algorithms MILEN and VALEN in Theorem 2 is relatively small.

4.5.5 Convergence of the Recovered Label Distribution

Figure 8 demonstrates the convergence of the recovered label distribution matrix over all training examples as the number of epochs increases (after warm-up training) on the `Fashion-MNIST` and `Kuzushiji-MNIST` datasets. It is evident from the plots that the recovered label distributions converge rapidly with the increasing number of epochs.

4.5.6 The Impact of Data Augmentation

In this study, we investigate the impact of data augmentation on MILEN, considering both the number of augmented images and different augmentation strategies. The results presented in Table 7 demonstrate that MILEN performs consistently well when the number of strongly augmented augmentations K varies. Based on accuracy and efficiency considerations, we choose a configuration with $K=3$ for our experiments.

Furthermore, Table 8 provides insights into the performance of MILEN under different augmentation strategies. Here, "W" represents weak augmentation, and "S" represents strong augmentation. The results indicate that MILEN achieves the best performance when utilizing one weak augmentation and two strong augmentations in the data augmentation process.

5 CONCLUSION

This paper focuses on addressing the problem of partial label learning and introduces two novel approaches, namely VALEN and MILEN. In this work, we specifically consider the instance-dependent PLL scenario, where each partially labeled example is associated with a latent label distribution. Recovering this latent label distribution is crucial for effective predictive model training. The proposed methods VALEN and MILEN iteratively recover the latent label distributions and train the predictive model in every epoch. The effectiveness of the proposed approaches is validated via comprehensive experiments on both synthesis datasets and real-world PLL datasets.

Looking ahead, it would be interesting to explore alternative methods for recovering latent label distributions in the context of instance-dependent PLL. Additionally, the incorporation of pseudo-labels with meta-information could be further investigated as a means to handle unreliable PLL scenarios. These avenues of research hold promise for advancing the field of partial label learning.

REFERENCES

[1] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France, 2017.

[2] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[3] Forrest Briggs, Xiaoli Z Fern, and Raviv Raich. Rank-loss support instance machines for MIML instance annotation. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 534–542, Beijing, China, 2012.

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

[5] Yi-Chen Chen, Vishal M Patel, Rama Chellappa, and P Jonathon Phillips. Ambiguously labeled learning using dictionaries. *IEEE Transactions on Information Forensics and Security*, 9(12):2076–2088, 2014.

[6] 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.

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

[8] Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the 32nd IEEE Conference on Computer Vision and Pattern Recognition*, pages 113–123, Long Beach, CA, 2019.

[9] Xinyue Dong, Tingjin Luo, Ruidong Fan, Wenzhang Zhuge, and Chenping Hou. Active label distribution learning via kernel maximum mean discrepancy. *Frontiers Comput. Sci.*, 17(4):174327, 2023.

[10] Lei Feng and Bo An. Leveraging latent label distributions for partial label learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2107–2113, Stockholm, Sweden, 2018.

[11] 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 33*, pages 10948–10960, Virtual Event, 2020.

[12] Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems 31*, pages 439–450, Montréal, Canada, 2018.

[13] Dan Garrette and Jason Baldridge. Learning a part-of-speech tagger from two hours of annotation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 138–147, Atlanta, Georgia, 2013.

[14] Xin Geng. Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1734–1748, 2016.

[15] Xin Geng, Chao Yin, and Zhi-Hua Zhou. Facial age estimation by learning from label distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(10):2401–2412, 2013.

[16] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *Proceedings of the 11th European Conference on Computer Vision*, pages 634–647, Heraklion, Crete, Greece, 2010.

[17] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, LA, 2019.

[18] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(10):1936–1949, 2014.

[19] Mark J Huiskes and Michael S Lew. The mir flickr retrieval evaluation. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 39–43, Vancouver, Canada, 2008.

[20] Eyke Hüllermeier and Jürgen Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439, 2006.

[21] Rong Jin and Zoubin Ghahramani. Learning with multiple labels. In *Advances in Neural Information Processing Systems 15*, pages 897–904, Vancouver, British Columbia, Canada, 2002.

[22] Julian Katz-Samuels, Gilles Blanchard, and Clayton Scott. Decontamination of mutual contamination models. *Journal of Machine Learning Research*, 20(41):1–57, 2019.

[23] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations*, Banff, AB, Canada, 2014.

[24] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

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

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

[27] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *the 30th International Conference on Machine Learning Workshop on challenges in representation learning*, page 896, Atlanta, Georgia, 2013.

[28] Shan Li and Weihong Deng. Blended emotion in-the-wild: Multi-label facial expression recognition using crowdsourced annotations and deep locality feature learning. *International Journal of Computer Vision*, 127(6):884–906, 2019.

[29] Liping Liu and Thomas G Dietterich. A conditional multinomial mixture model for superset label learning. In *Advances in Neural Information Processing Systems 25*, pages 557–565, Lake Tahoe, Nevada, 2012.

[30] Jie Luo and Francesco Orabona. Learning from candidate labeling sets. In *Advances in Neural Information Processing Systems 23*, pages 1504–1512. Vancouver, British Columbia,Canada, 2010.

[31] 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 37th International Conference on Machine Learning*, pages 6500–6510, Virtual Event, 2020.

[32] Aditya Krishna Menon, Ankit Singh Rawat, Sashank J. Reddi, Seungyeon Kim, and Sanjiv Kumar. A statistical perspective on distillation. In *Proceedings of the 38th International Conference on Machine Learning*, pages 2128–2136, Virtual Event, 2021.

[33] Nam Nguyen and Rich Caruana. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 381–389, Las Vegas, NV, 2008.

[34] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[35] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the 34th IEEE Conference on Computer Vision and Pattern Recognition*, pages 11557–11568, Virtual Event, 2021.

[36] Cai-Zhi Tang and Min-Ling Zhang. Confidence-rated discriminative partial label learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 2611–2617, San Francisco, CA, 2017.

[37] Wei Tang, Weijia Zhang, and Min-Ling Zhang. Multi-instance partial-label learning: Towards exploiting dual inexact supervision. *Science China Information Sciences*, page in press, 2023.

[38] Deng-Bao Wang, Min-Ling Zhang, and Li Li. Adaptive graph guided disambiguation for partial label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):8796–8811, 2021.

[39] Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. Pico: Contrastive label disambiguation for partial label learning. In *in Proceedings of the 10th International Conference on Learning Representations*, Virtual Event, 2022.

[40] Hongwei Wen, Jingyi Cui, Hanyuan Hang, Jiabin Liu, Yisen Wang, and Zhouchen Lin. Leveraged weighted loss for partial label learning. In *Proceedings of the 38th International Conference on Machine Learning*, pages 11091–11100, Virtual Event, 2021.

[41] Dong-Dong Wu, Deng-Bao Wang, and Min-Ling Zhang. Revisiting consistency regularization for deep partial label learning. In *Proceedings of the 39th International Conference on Machine Learning*, pages 24212–24225, Baltimore, Maryland, 2022.

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

[43] Ning Xu, Yun-Peng Liu, and Xin Geng. Label enhancement for label distribution learning. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1632 – 1643, 2021.

[44] Ning Xu, Jiaqi Lv, and Xin Geng. Partial label learning via label enhancement. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 5557–5564, Honolulu, Hawaii, 2019.

[45] Ning Xu, Congyu Qiao, Xin Geng, and Min-Ling Zhang. Instance-dependent partial label learning. In *Advances in Neural Information Processing Systems 34*, pages 27119–27130, Virtual Event, 2021.

[46] Ning Xu, Jun Shu, RenYi Zheng, Xin Geng, Deyu Meng, and Min-Ling Zhang. Variational label enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6537–6551, 2023.

[47] Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. Active learning from imperfect labelers. In *Advances in Neural Information Processing Systems 29*, pages 2128–2136, Barcelona, Spain, 2016.

[48] Jufeng Yang, Ming Sun, and Xiaoxiao Sun. Learning visual sentiment distributions via augmented conditional probability neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 224–230, San Francisco, California, 2017.

[49] Yao Yao, Jiehui Deng, Xiuhua Chen, Chen Gong, Jianxin Wu, and Jian Yang. Deep discriminative cnn with temporal ensembling for ambiguously-labeled image classification. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 12669–12676, New York, NY, 2020.

[50] Fei Yu and Min-Ling Zhang. Maximum margin partial label learning. *Machine Learning*, 106(4):573–593, 2017.

[51] Zinan Zeng, Shijie Xiao, Kui Jia, Tsung-Han Chan, Shenghua Gao, Dong Xu, and Yi Ma. Learning by associating ambiguously labeled images. In *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition*, pages 708–715, Portland, OR, 2013.

[52] 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 10th International Conference on Learning Representations*, Virtual Event, 2022.

[53] Min-Ling Zhang and Fei Yu. Solving the partial label learning problem: An instance-based approach. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 4048–4054, Buenos Aires, Argentina, 2015.

[54] Min-Ling Zhang, Fei Yu, and Cai-Zhi Tang. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2155–2167, 2017.

[55] Min-Ling Zhang, Bin-Bin Zhou, and Xu-Ying Liu. Partial label learning via feature-aware disambiguation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1335–1344, San Francisco, CA, 2016.

[56] Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. Learning with feature dependent label noise: a progressive approach. In *Proceedings of the 9th International Conference on Learning Representations*, Virtual Event, 2021.

[57] Qinghai Zheng, Jihua Zhu, and Haoyu Tang. Label information bottleneck for label enhancement. In *Proceedings of the 36th IEEE Conference on Computer Vision and Pattern Recognition*, pages 7497–7506, Vancouver, Canada, 2023.