Making Partial-Label Datasets Easier: A Simple Yet Highly Effective Data Augmentation for Deep Partial-Label Learning

Dong-Dong Wu[1], Zhaoyi Li[1], Xiang Li[2], and Zhiqiang Shen[1]

[1] Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
[2] George Mason University, Fairfax, USA

Abstract. Partial-label learning (PLL), which refers to the classification task where each training instance is associated with a set of candidate-labels, among which only one is correct. While recent advances in deep PLL primarily focus on designing sophisticated disambiguation strategies, the role of PLL-specific data augmentation has been less explored. Most methods often treat data augmentation as a generic plug-in, overlooking its potential interaction with the label disambiguation process unique to PLL. To address this gap, we propose a local-ambiguity-aware augmentation framework that consists of label identification and model regularization modules and accommodates a wide range of PLL algorithms. Within this framework, a PLL-specific augmentation strategy based on a mixing-renormalization process is applied to partial-label data. Theoretically, we show that our augmentation can make the PLL dataset easier by reducing the local ambiguity degree, and it induces multiple implicit regularizations during training thereby facilitating highly effective disambiguation. Extensive experiments on both benchmark and real-world datasets demonstrate that our approach consistently improves the generalization performance of state-of-the-art PLL algorithms.

1 Introduction

Partial-label learning (PLL) is a weakly supervised learning paradigm that aims to reduce the cost of manual annotation, and has recently attracted significant attention [35, 37]. In PLL, each training instance is associated with a set of candidate-labels, with only one being the true label [5]. Due to its cost-effectiveness, PLL has been successfully applied in various fields, including visual recognition [4,48,55], natural language processing [61], and audio processing [17].

The core challenge of PLL lies in label ambiguity, where the true label is unknown but included in the candidate set. This ambiguity can be categorized into class-dependent [13, 46] and instance-dependent [54] types. To tackle this challenge, numerous approaches have been proposed from the perspectives of statistical consistency [13,32,46,53] and training techniques [9,40,41,49,51]. While these sophisticated approaches have shown great promise in disambiguation, the role of data augmentation in deep PLL has remained relatively underexplored

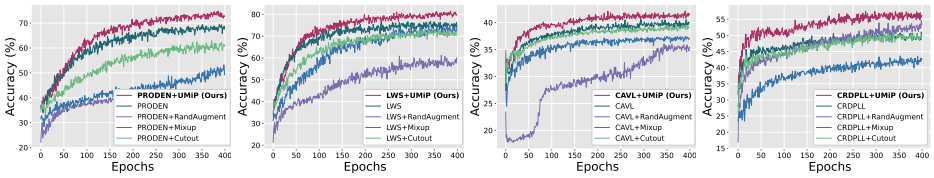


Fig. 1: Test accuracy on the instance-dependent CIFAR-10 dataset: four PLL baselines (PRODEN [25], LWS [46], CAVL [58], CRDPLL [49]), baselines with advanced data augmentations applied directly to partial-label data, and baselines with UMiP.

and insufficiently understood. Most existing PLL methods treat data augmentation as a simple plug-in module, without systematically considering its interaction with the inherent ambiguity of PLL datasets. Although advanced data augmentations [6–8, 59] have been empirically effective in fully supervised learning, directly applying them to partial-label data, i.e., input-candidate-label set pair $(\boldsymbol{x}_i, \mathcal{S}_i)$, is nontrivial and can even be detrimental. As shown in Fig. 1, when RandAugment [6] or Cutout [8] are applied to produce $(\mathrm{Aug}(\boldsymbol{x}_i), \mathcal{S}_i)$, methods relying on a single-view augmentation, such as PRODEN [25], LWS [46], and CAVL [58], exhibit diminished effectiveness. Similarly, when Mixup [59] is applied to generate $(\lambda\boldsymbol{x}_i + (1-\lambda)\boldsymbol{x}_j, \lambda\mathcal{S}_i + (1-\lambda)\mathcal{S}_j)$, approaches without pseudo-labeling mechanisms, including PRODEN, LWS, CAVL, and CRDPLL [49], also suffer performance degradation. These observations raise a critical question:

What data augmentations are broadly effective for PLL?

To address this question, we argue that *an effective PLL-specific data augmentation should reduce the difficulty of label disambiguation of PLL datasets.* To this end, we first analyze the determinants of disambiguation difficulty through the lens of local neighborhood structure in PLL data. Notably, even when PLL datasets contain the same number of candidate-labels, their disambiguation difficulty can still vary significantly depending on how the candidate-labels are assigned, such as through instance-based generation, class-wise assignment or random sampling. The fundamental difference among these cases lies in the local neighborhood structure of the data distribution, particularly in the extent of manifold coupling between neighboring sample features and their candidate-labels. For example, the instance-dependent setting is often the most challenging, as neighboring samples tend to share highly similar candidate-label sets. To quantify this coupling difference, we introduce a new dataset-level metric, the *local ambiguity degree* (LAD), which captures subtle local structural patterns. Building on this analysis, an effective perturbation-based strategy, which perturbs the local neighborhood structure in the input space while keeping the candidate-label set unchanged, is designed to reduce the coupling between feature and label manifolds. Accordingly, we propose a general training framework named *UMiP*, which integrates a label identification module with a model regularization module, making it compatible with a wide range of PLL algorithms. Specifically, PLL-augmentation is realized in the label identification module by

applying dominant mixing, where the input and label confidence of each sample are skewedly mixed with those of a randomly selected counterpart, while preserving the original candidate-label set and subsequently renormalizing the label confidence. As for the model regularization component, we retain the regularization module used by the original methods.

To validate the effectiveness of our proposed PLL augmentation, we conduct both empirical evaluations and theoretical analysis. Empirically, UMiP proves to be simple yet highly effective, especially compared to approaches tailored to specific candidate-label types or involving complex architectures. For example, as shown in Tab. 1 (SVHN, $q_1 = 1.0$), integrating UMiP into PRODEN improves accuracy from 83.74% to a state-of-the-art 95.64%, outperforming PiCO [40]'s 93.82%, with minimal computational overhead. On the theoretical side, the performance gains can be attributed to the reduction of LAD in PLL datasets and multiple implicit regularization effects induced during the disambiguation process. Our main contributions can be summarized as follows:

- We identify a previously overlooked challenge of data augmentation in deep PLL and propose a general augmentation framework that can be seamlessly integrated into a wide range of identification-based PLL algorithms with minimal computational overhead.
- We introduce the local ambiguity degree, a general metric that quantifies the disambiguation difficulty jointly induced by the local neighborhood structure in the feature space and ambiguous label information in the label space.
- We provide a rigorous theoretical analysis showing that our method effectively reduces local ambiguity degree and induces implicit regularizations, such as Jacobian regularization, to facilitate disambiguation.
- Extensive evaluations on benchmark and real-world PLL datasets demonstrate the effectiveness of our augmentation method in enhancing the generalization performance of state-of-the-art PLL algorithms.

2 Related Work

Partial-label learning (PLL) has been widely applied in various fields [2, 4, 10, 11, 23, 26, 42, 57, 61]. Recent deep PLL methods broadly fall into two categories: theory-oriented and empirically-oriented. Theory-oriented approaches focus on modeling the data generation process and ensuring statistical consistency [13, 46, 53]. Empirically-oriented methods leverage advanced training strategies to identify true labels, such as contrastive learning [33,40], consistency regularization [49], label smoothing [15], sample selection [36] and diffusion disambiguation [12, 47]. Despite these advances, the role of data augmentation in PLL remains underexplored. Most methods treat it as a standard plug-in without addressing the specific challenges of label ambiguity, motivating our proposed PLL-specific augmentation framework.

Data Augmentation in PLL. While advanced augmentations like Mixup and RandAugment are effective in semi-supervised [1] and noisy-label learning [27, 29], their direct application in PLL remains challenging. As demonstrated in [24], directly applying those advanced augmentations to partial-label data risks discarding task-relevant information, yet it remains effective in auxiliary regularization terms (e.g., contrastive learning or consistency penalties) to aid representation learning, as exemplified by PiCO and CRDPLL. In contrast, our UMiP solves this dilemma by operating directly within the label-identification loss to reduce the local ambiguity, thereby restoring identification information while preserving representation ability. Although PLDA [43] attempts PLL-specific augmentation, it relies on hand-crafted features and linear models, limiting its scalability.

3 Methodology

Notations. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the d-dimensional feature space and $\mathcal{Y} = \{1, \ldots, C\}$ be the label space with C classes. We consider a PLL dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \mathcal{S}_i)\}_{i=1}^n$, where each instance $\boldsymbol{x}_i \in \mathcal{X}$ is associated with a candidate-label set $\mathcal{S}_i \subseteq \mathcal{Y}$ that contains the latent true label y_i. The objective is to learn a classifier $f : \mathcal{X} \to \mathbb{R}^C$ that generalizes well to unseen data. We denote by $f_c(\boldsymbol{x})$ the model output (e.g., predicted probability) for the c-th class given input $\boldsymbol{x}$.

3.1 Analysis of Local Ambiguity Degree

To characterize the ambiguity in PLL datasets beyond global statistics [5], we introduce a *local* measure that explicitly accounts for local neighbourhood structure. Crucially, even when candidate-label sets exhibit identical global statistics, instance-dependent PLL is often more challenging than class-dependent PLL due to local neighbourhood effects—a distinction our measure is designed to capture.

Assumption 1 (k-NN label clusterability [62]) *A dataset satisfies the k-NN label clusterability condition if each sample and its k-nearest neighbours belong to the same ground-truth label.*

Assuming the underlying oracle dataset $\hat{\mathcal{D}}_n = \{(\boldsymbol{x}_i, y_i, \mathcal{S}_i)\}_{i=1}^n$ satisfies this condition, we can estimate the local ambiguity degree of each sample by aggregating candidate-label information from its k-nearest neighbors. Based on this, we define the label that contributes most to a sample's local ambiguity as the *k-consensus ambiguous label*.

Definition 1 (k-consensus ambiguous label). *Given a PLL instance $(\boldsymbol{x}_i, \mathcal{S}_i)$, its k-consensus ambiguous label, denoted by $\widetilde{y}_i \in \mathcal{S}_i \setminus \{y_i\}$, is defined as the label that appears most frequently in the candidate-label sets of its k-nearest neighbors.*

This Definition 1 provides a way to identify the primary source of local ambiguity for each sample. Then we define the local ambiguity degree (LAD) to quantify the PLL dataset's class-wise worst-case ambiguity as follows.

Definition 2 (Local ambiguity degree). *Given an oracle dataset* $\bar{\mathcal{D}}_n = \{(\boldsymbol{x}_i, y_i, \tilde{y}_i, \mathcal{S}_i)\}_{i=1}^n$*, let* $\Psi(\cdot) \in \{0,1\}^C$ *denote the function that transforms the candidate set* $\mathcal{S}_i$ *into an indicator vector, such that its c-th entry* $[\Psi(\mathcal{S}_i)]_c = 1$ *if* $c \in \mathcal{S}_i$ *and 0 otherwise. The local ambiguity degree* ε *is defined as*

$$\varepsilon = \sup_{Y \in \mathcal{Y}} \mathbb{E}_{(x,y,\tilde{y},\mathcal{S}) \sim \bar{\mathcal{D}}_n} [Z_{\widetilde{y}}(\boldsymbol{x}) | y = Y], \tag{1}$$

where $Z_{\widetilde{y}}(\boldsymbol{x}) = \frac{1}{k} \sum_{j \in \mathcal{N}_{\boldsymbol{x}}} [\Psi(\mathcal{S}_j)]_{\widetilde{y}}$ *represents the frequency of the consensus ambiguous label* $\widetilde{y}$ *among the k-nearest neighbors* $\mathcal{N}_{\boldsymbol{x}}$ *of* $\boldsymbol{x}$*.*

Remark 1 (Interpretation and Extreme Cases). (i) The value $\varepsilon \in [0,1]$ quantifies the expected frequency of the most confusing incorrect label within the local neighborhood of the worst-case class. (ii) If $\varepsilon = 0$, it implies that each instance's candidate set contains only the true label, reducing to the fully supervised setting. (iii) If $\varepsilon = 1$, there exists a class where every sample's candidate set consistently includes a specific indistinguishable incorrect label, rendering the class unlearnable.

Remark 2 (On Oracle Label Access). Although PLL assumes ground-truth labels are unavailable during training, we adopt an oracle (*proof-of-concept*) setup in this analysis solely to quantify the underlying LAD changes and establish theoretical results. They are never used for training. Moreover, the underlying mechanism of our proposed augmentation persists under the standard PLL protocol without any oracle access.

Remark 3 (Difference from Traditional Ambiguity Measures [5]). Classical ambiguity measures often rely on knowledge of the underlying data distribution or strong modeling assumptions, which are difficult to satisfy in PLL, particularly in instance-dependent settings. In contrast, the proposed LAD is data-driven and computable: it leverages k-NN aggregation over observed candidate sets to estimate the practical ambiguity, without requiring full distribution modeling.

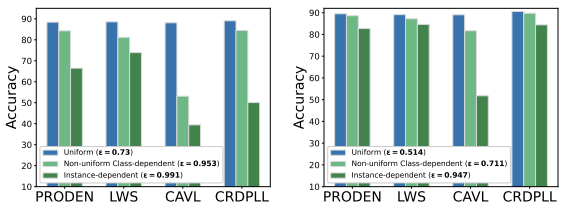


Fig. 2: Baseline performance on CIFAR-10 dataset under two average candidate-label (avgCL) settings: (Left) avgCL = 5.4 and (Right) avgCL = 4.0.

As noted above, different candidate-labels generation mechanisms induce varying coupling behaviors, and thus different LAD. Specifically, in the instance-dependent setting, feature-similar samples tend to share similar candidate-labels. In the class-dependent setting, samples from the same class are more likely to share similar candidate-labels. Conversely, in the uniform setting [13], candidate-labels are sampled independently of both features and classes. To isolate the impact of LAD on performance, we compare models across these

settings while fixing the average candidate set size (Fig. 2). The results reveal a clear inverse relationship: lower LAD corresponds to higher accuracy.

Theoretically, this trend admits a precise characterization: With a fixed average candidate-set size, the uniform setting minimizes LAD, as formalized below.

Proposition 1. *Consider a PLL dataset where the expected candidate set size is fixed at r, i.e., $\mathbb{E}[|\mathcal{S}|] = r$. Then, the LAD ε satisfies the lower bound:*

$$\varepsilon \;\geq\; \varepsilon_{\text{uni}} \;:=\; \frac{r-1}{C-1},$$

where ε_{uni} corresponds to the LAD under the uniform generation setting with expected size r. Furthermore, the equality holds if and only if, for every class Y and every incorrect label $c \neq Y$,

$$\mathbb{P}(c \in \mathcal{S} \mid y = Y) = \frac{r-1}{C-1},$$

which implies that the generation process is uniform. The proof is in Appendix.

Taken together, these empirical and theoretical observations suggest a design principle for LAD-aware augmentation: *Weaken the local coupling structure to reduce the dependency between a sample's candidate-label set and those of its neighbors. This strategy aims to push ε toward its lower bound ε_{uni}, thereby improving performance.*

3.2 LAD-aware Augmentation for PLL

We implement this principle by perturbing only the input space to weaken local neighborhood coupling, while strictly preserving the candidate-label sets. We instantiate it via an algorithm-agnostic mixing–renormalization process.

General PLL training framework. To ensure broad compatibility with deep PLL algorithms, we adopt a unified identification-based formulation, which typically consists of a label identification loss and a regularization term, balanced by a weighting function $\Phi(\gamma)$ with hyperparameter γ. For an instance $\boldsymbol{x}_i$, the loss is formulated as

$$\mathcal{L}(\boldsymbol{x}_i, \boldsymbol{p}_i, \mathcal{S}_i) = \ell(\boldsymbol{p}_i, g(f(\boldsymbol{x}_i))) + \Phi(\gamma) \cdot \mathcal{L}_{\text{reg}}(\boldsymbol{x}_i, \mathcal{S}_i), \tag{2}$$

where $\boldsymbol{p}_i$ denotes the estimated label confidence, $g(\cdot)$ is a post-processing function (e.g., an identity mapping), and $\ell(\cdot)$ represents a type of alignment loss function that inherently drives the label disambiguation. Eq. (2) provides a unified perspective on the identification-based PLL training. In Appendix B.3, we further show how various PLL algorithms [25, 40, 46, 49–52, 58] can be cast into this framework. Within this structure, we incorporate our LAD-aware augmentation into the label-identification loss $\ell(\boldsymbol{p}_i, g(f(\boldsymbol{x}_i)))$, as detailed below.

Dominant Mixing. To better disrupt local neighborhood structure while preserving semantic integrity, we adopt a mixing-based strategy over generic augmentations like Cutout [8] or RandAugment [6]. Specifically, we apply dominant

mixing to both the input and the estimated label confidence while restricting the confidence mixing to the original candidate-label set. Concretely, each instance is mixed with a randomly selected counterpart as follows

$$\widetilde{\boldsymbol{x}}_i = \lambda \boldsymbol{x}_i + (1-\lambda)\boldsymbol{x}_j, \quad \widetilde{\boldsymbol{p}}_i = \lambda \boldsymbol{p}_i + (1-\lambda)\boldsymbol{p}_j, \quad \widetilde{\mathcal{S}}_i = \mathcal{S}_i, \tag{3}$$

where λ is drawn from truncated $\text{Beta}_{[0.5,1]}(\alpha, \alpha)$ to ensure that $\boldsymbol{x}_i$ remains dominant in the mixture. In our implementation, α is set to 1.0 so that λ is uniformly drawn from the interval $[0.5, 1]$. This operation stochastically shifted samples and distorts relative positions while keeping the candidate set $\mathcal{S}_i$ unchanged.

Label Renormalization. As shown in Proposition 2 (proved in Appendix A.2), dominant mixing acts as a stochastic perturbation applied to input-confidence pairs. However, this operation alone does not enforce the PLL constraint on original candidate-label sets, so the perturbed confidence may place probability mass on non-candidate labels. To prevent such leakage, we propose to renormalize the confidence vector over the original candidate set using a post-processing operator $\hbar(\cdot)$.

Proposition 2. *Let $\lambda \sim Beta_{[0.5,1]}(\alpha, \alpha)$ and $j \sim Uniform([n])$ be two random variables with $\alpha > 0$, $n > 0$, and let $\bar{\lambda} = \mathbb{E}[\lambda]$. For any $i \in [n]$, the dominant mixing sample $(\widetilde{\boldsymbol{x}}_i, \widetilde{\boldsymbol{p}}_i)$ defined in Eq. (3) can be reformulated as:*

$$\begin{aligned} \widetilde{\boldsymbol{x}}_i &= \bar{\boldsymbol{x}} + \bar{\lambda}\,(\boldsymbol{x}_i - \bar{\boldsymbol{x}}) + (\lambda - \bar{\lambda})\boldsymbol{x}_i + (1-\lambda)\boldsymbol{x}_j - (1-\bar{\lambda})\bar{\boldsymbol{x}}, \\ \widetilde{\boldsymbol{p}}_i &= \underbrace{\bar{\boldsymbol{p}} + \bar{\lambda}\,(\boldsymbol{p}_i - \bar{\boldsymbol{p}})}_{\text{Data Transformation } \boldsymbol{x}_i', \boldsymbol{p}_i'} + \underbrace{(\lambda - \bar{\lambda})\boldsymbol{p}_i + (1-\lambda)\boldsymbol{p}_j - (1-\bar{\lambda})\bar{\boldsymbol{p}}}_{\text{Random Perturbation } \boldsymbol{\epsilon}_i^x, \boldsymbol{\epsilon}_i^p}, \end{aligned}$$

where $\bar{\boldsymbol{x}}$, $\bar{\boldsymbol{p}}$ are the means of data inputs and label confidences of all training samples. The perturbation terms satisfy $\mathbb{E}_{[\lambda,j]}[\boldsymbol{\epsilon}_i^x] = \mathbb{E}_{[\lambda,j]}[\boldsymbol{\epsilon}_i^p] = \mathbf{0}$, with $\boldsymbol{\epsilon}_i^x \in \mathbb{R}^d, \boldsymbol{\epsilon}_i^p \in \mathbb{R}^C$. For any classification function $f(\cdot)$, the label identification loss in Eq. (2) can be viewed as learning with stochastic perturbations:

$$\ell(\widetilde{\boldsymbol{p}}_i, g(f(\widetilde{\boldsymbol{x}}_i))) = \mathbb{E}_{\theta,j}\ell(\boldsymbol{p}_i' + \boldsymbol{\epsilon}_i^p, g(f(\boldsymbol{x}_i' + \boldsymbol{\epsilon}_i^x))).$$

Notably, the concrete implementation of $\hbar(\cdot)$ varies by PLL algorithms. To illustrate how $\hbar(\cdot)$ is instantiated in practice, we categorize common renormalization strategies used in existing deep PLL methods as follows:

– **In-Set Renormalization** [19, 25, 40, 49, 51, 52]. Only candidate labels are retained, and the confidence distribution is renormalized over the original candidate set, while non-candidate labels are zeroed out. This preserves relative confidence within the candidate set and enforces the PLL constraint:

$$\hbar_c(\widetilde{\boldsymbol{p}}_i) = \frac{\widetilde{p}_{ic}\,\mathbf{1}_{\{c \in \mathcal{S}_i\}}}{\sum_{k \in \mathcal{S}_i} \widetilde{p}_{ik}}\,. \tag{4}$$

– **Bi-Partition Renormalization** [14, 39, 46]. Candidate and non-candidate blocks are renormalized independently within their respective subsets and then

combined. This retains information carried by non-candidate labels while preserving intra-subset rankings on both sides:

$$\hbar_c(\widetilde{\boldsymbol{p}}_i) = \frac{\widetilde{p}_{ic}\,\mathbf{1}_{\{c\in\mathcal{S}_i\}}}{\sum_{k\in\mathcal{S}_i}\widetilde{p}_{ik}} + \frac{\widetilde{p}_{ic}\,\mathbf{1}_{\{c\notin\mathcal{S}_i\}}}{\sum_{k\notin\mathcal{S}_i}\widetilde{p}_{ik}}\,. \tag{5}$$

– **Margin-Controlled Bi-Partition** [50]. An adaptive factor $\eta \in [0,1]$, derived from the confidence margin, balances the two subsets to ensure a separation constraint: the lowest confidence among candidate labels should exceed the highest confidence among non-candidate labels. This data-dependent margin helps in distinguishing likely true labels from distractors:

$$\hbar_c(\widetilde{\boldsymbol{p}}_i) = \eta\,\frac{\widetilde{p}_{ic}\,\mathbf{1}_{\{c\in\mathcal{S}_i\}}}{\sum_{k\in\mathcal{S}_i}\widetilde{p}_{ik}} + (1-\eta)\,\frac{\widetilde{p}_{ic}\,\mathbf{1}_{\{c\notin\mathcal{S}_i\}}}{\sum_{k\notin\mathcal{S}_i}\widetilde{p}_{ik}}\,. \tag{6}$$

– **In-Set Argmax** [36, 58]. Within the candidate set, the maximal-confidence class is selected to form a one-hot target; other classes are zeroed. This converts soft confidences to a hard pseudo-label constrained by the candidate set:

$$\hbar_c(\widetilde{\boldsymbol{p}}_i) = \mathbf{1}\left\{\, c = \arg\max_{k\in\mathcal{S}_i}\widetilde{p}_{ik}\right\}. \tag{7}$$

After renormalization, the final mixed label confidence $\widehat{\boldsymbol{p}}_i = \hbar(\widetilde{\boldsymbol{p}}_i)$ is obtained for the perturbed input $\widetilde{\boldsymbol{x}}_i$. These augmented input-confidence pairs $(\widetilde{\boldsymbol{x}}_i, \widehat{\boldsymbol{p}}_i)$ are then substituted into Eq. (2) to compute the overall training loss:

$$\mathcal{L}(\widetilde{\boldsymbol{x}}_i, \widehat{\boldsymbol{p}}_i, \mathcal{S}_i) = \ell(\widehat{\boldsymbol{p}}_i, g(f(\widetilde{\boldsymbol{x}}_i))) + \Phi(\gamma)\cdot\mathcal{L}_{\mathrm{reg}}(\boldsymbol{x}_i, \mathcal{S}_i), \tag{8}$$

As demonstrated, our method introduces minimal modifications to the training pipeline and can be seamlessly integrated into a wide range of existing PLL algorithms without requiring additional architectural changes.

Remark 4 (Distinction from Commonly-used Augmentations in PLL). Prior PLL methods often integrate generic augmentations (e.g., Mixup, RandAugment, Cutout) into auxiliary objectives within the *regularization term* $\mathcal{L}_{\mathrm{reg}}$ of Eq. (2) (e.g., contrastive learning [40, 51], knowledge distillation [50], and consistency penalties [49]). In contrast, UMiP directly operates within the *label-identification loss* on input-candidate pairs. This design is explicitly guided by our LAD analysis to weaken local neighborhood coupling. Consequently, UMiP is PLL-specific and orthogonal to regularizer-side generic augmentations; in practice, it can be effectively combined with them.

4 Theoretical Analysis

Beyond empirical effectiveness, we provide a theoretical analysis to elucidate the mechanism behind UMiP. In this section, we demonstrate how it facilitates disambiguation through ambiguity reduction and implicit regularization.

Ambiguity reduction. When the proposed augmentation UMiP is applied to partial-label data, the augmented PLL dataset's LAD decreases. We formalize this effect in the following theorem (proved in Appendix A.3).

Theorem 1 (LAD Reduction). *Suppose the dataset satisfies the k-NN label clusterability. Let ε be the LAD of the original PLL dataset. Then, there exists a coefficient $\rho \in (0,1)$, determined by the mixing distribution and feature geometry, such that the post-augmentation LAD, denoted $\varepsilon^{\mathrm{UMiP}}$, satisfies:*
(a) ***Class-dependent case.*** *Let $T \in [0,1]^{C\times C}$ be the class-dependent transition matrix with entries $T_{uv} = \Pr(v \in \mathcal{S} \,|\, y = u)$, and let $\varepsilon = \max_{u\in\mathcal{Y}} \max_{v\neq u} T_{uv}$, $\tau_v = \frac{1}{C}\sum_{u\in\mathcal{Y}} T_{uv}$, $\tau_{\max} = \max_{v\in\mathcal{Y}} \tau_v$. Then,*

$$\varepsilon_{\mathrm{uni}} \;\leq\; \varepsilon^{\mathrm{UMiP}} \;\leq\; \rho\,\varepsilon + (1-\rho)\,\tau_{\max} \;<\; \varepsilon \tag{9}$$

where $\varepsilon_{\mathrm{uni}} = \frac{r-1}{C-1}$ is the uniform lower bound (Proposition 1). The strict inequality $\varepsilon^{\mathrm{UMiP}} < \varepsilon$ holds whenever T is non-uniform and $\rho < 1$.
(b) ***Instance-dependent case.*** *With the same $\rho \in (0,1)$ as above,*

$$\varepsilon_{\mathrm{uni}} \;\leq\; \varepsilon^{\mathrm{UMiP}} \;\leq\; \rho\,\varepsilon + (1-\rho)\,\varepsilon_{\mathrm{uni}} \;<\; \varepsilon. \tag{10}$$

Hence, whenever $\varepsilon > \varepsilon_{\mathrm{uni}}$, UMiP strictly reduces the LAD.

Implicit regularization. Building on the formulation in Proposition 2, the expected label identification loss under UMiP can be expressed as

$$\ell(\widehat{\boldsymbol{p}}_i, g(f(\widetilde{\boldsymbol{x}}_i))) = \mathbb{E}_{\theta,j}\,\ell(\hbar(\boldsymbol{p}'_i + \boldsymbol{\epsilon}^p_i), g(f(\boldsymbol{x}'_i + \boldsymbol{\epsilon}^x_i))). \tag{11}$$

As established in prior theoretical works [3,34,38,44], learning with such stochastic perturbations naturally induces implicit regularization effects (such as *Jacobian regularization*, *weight normalization*, and *label smoothing*) that help prevent overfitting, enhance generalization, and improve calibration. Consequently, even in uniform PLL scenarios where LAD reduction is minimal, UMiP can still yield performance gains through this implicit regularization mechanism.

5 Experiment

5.1 Evaluation Dataset

Dataset. We employ four widely used benchmark image datasets: CIFAR-10 [21], CIFAR-100 [21], SVHN [28], and CUB-200 [45], as well as six real-world PLL datasets: Lost [5], BirdSong [2], MSRCv2 [22], Soccer Player [57], Yahoo! News [16], and Mirflicker [20].
Candidate-label generation. For the benchmark image datasets, we simulate the real-world process of generating candidate-label sets using instance-dependent generation [54] and class-dependent generation [46] models. Specifically, in the instance-dependent generation process, we use the prediction of a neural network trained on the original clean labels to determine the instance-wise flipping probabilities, with $q_1 \in (0,\infty)$ controlling the level of ambiguity.

For the class-dependent generation process, we define a transition matrix that associates each true label with a set of similar labels, where each similar label is chosen as a candidate-label with a probability $q_2 \in (0, 1)$. In addition, we consider a special case of the class-dependent setting, known as the uniform setting [13], where each incorrect label is selected into the candidate-label set with the same probability $q_3 \in (0, 1)$. Further details on the generation process and characteristic information of PLL datasets can be found in Appendix B.2.

Table 1: Test accuracy (%) under instance-dependent PLL settings. Results with our UMiP augmentation are shaded in gray, where ↑ indicates performance improvement over the original baseline.

Dataset	q_1	PRODEN	LWS	CRDPLL	PiCO	ABLE	DIRK	PaPi
CIFAR-10	0.6	77.41±2.0	81.65±0.2	73.40±1.3	77.33±1.5	80.13±0.9	76.98±1.2	81.23±0.8
		80.07±0.7↑	83.32±0.4↑	74.51±0.3↑	79.37±0.9↑	82.08±0.4↑	79.68±0.8↑	82.81±0.3↑
	0.8	66.48±1.2	73.88±1.1	50.08±1.6	66.57±0.3	67.21±0.9	68.96±1.1	62.86±1.2
		70.11±1.3↑	77.87±0.3↑	50.14±0.9↑	68.79±1.4↑	72.62±0.9↑	74.50±0.4↑	69.04±1.0↑
	1.0	55.41±0.5	66.08±0.9	43.86±0.5	53.42±1.9	58.34±0.5	62.48±0.4	52.25±0.3
		60.31±1.6↑	70.38±1.6↑	43.98±0.3↑	57.76±1.0↑	62.98±0.1↑	68.23±0.3↑	60.02±0.3↑
SVHN	0.6	91.79±0.2	90.42±1.1	79.45±0.1	93.71±0.2	95.02±0.1	92.64±0.3	89.25±0.5
		95.82±0.2↑	95.76±0.2↑	79.70±0.3↑	95.34±0.3↑	95.96±0.1↑	94.58±0.2↑	93.89±0.1↑
	0.8	88.88±1.6	84.18±1.7	78.50±0.5	93.82±0.1	94.35±0.1	92.11±0.4	86.47±0.4
		95.82±0.2↑	95.62±0.1↑	78.81±0.1↑	94.75±0.0↑	94.36±0.1↑	93.83±0.2↑	89.14±0.1↑
	1.0	83.74±0.3	81.67±1.2	68.39±0.8	93.59±0.3	92.92±0.1	91.72±0.8	82.36±0.4
		95.64±0.1↑	93.50±1.9↑	71.78±1.5↑	95.07±0.2↑	93.97±0.1↑	92.72±0.7↑	88.34±0.4↑
CIFAR-100	0.01	75.05±0.3	74.61±0.3	77.51±0.8	72.82±1.4	74.78±0.4	74.01±0.3	76.10±0.6
		76.08±0.6↑	76.16±0.5↑	77.69±0.6↑	77.84±0.6↑	75.02±0.3↑	75.38±0.3↑	76.13±0.2↑
	0.05	71.57±0.2	71.93±0.6	72.74±0.1	67.50±0.1	71.78±0.3	70.52±0.1	72.03±0.3
		72.31±1.2↑	73.07±0.5↑	72.86±0.3↑	68.39±0.8↑	71.86±0.3↑	72.43±0.1↑	73.15±0.2↑
	0.1	66.14±0.4	67.55±0.3	64.53±0.3	58.42±1.4	65.94±0.4	62.17±0.5	64.65±0.2
		66.65±0.6↑	67.59±0.5↑	64.64±0.1↑	58.65±0.6↑	66.36±0.3↑	63.42±0.3↑	64.81±0.2↑
CUB-200	0.01	51.21±3.0	51.38±3.4	62.95±3.5	37.83±1.9	58.12±0.9	65.59±0.3	61.23±1.1
		56.35±1.3↑	57.42±3.2↑	67.72±1.3↑	45.08±1.1↑	60.21±0.3↑	66.45±0.2↑	64.36±0.5↑
	0.02	46.69±2.8	49.66±0.9	55.63±2.1	27.98±0.8	49.41±1.2	57.35±0.8	56.63±1.2
		51.90±1.8↑	52.47±1.2↑	60.29±0.3↑	31.74±1.0↑	52.89±0.7↑	59.09±0.7↑	59.44±0.4↑
	0.03	41.04±2.4	42.87±6.3	46.85±1.8	23.23±0.7	48.25±1.1	53.27±1.2	47.85±1.1
		44.70±2.1↑	47.98±2.3↑	51.57±1.1↑	26.25±0.3↑	52.43±0.7↑	54.24±0.7↑	52.32±0.4↑

5.2 Experimental Setup

We integrate UMiP into representative PLL baselines and evaluate their performance on both synthetic and real-world PLL tasks. We then compare the augmented models with their original counterparts to assess whether the proposed augmentation method consistently improves the generalization performance of PLL algorithms. To perform a comprehensive evaluation, we adopt seven well-performing identification-based PLL algorithms as baselines: PRODEN [25], LWS [46], CRDPLL [49], PiCO [40], ABLE [51], DIRK [50], PaPi [52]. Although PLDA [43] is a PLL-aware augmentation, its reliance on linear models restricts it to small-scale settings and precludes end-to-end deep learning. To provide a comparative analysis on tabular datasets, we adapt PLDA into a two-stage pipeline: first generating augmented instances via the original PLDA optimization, then training deep PLL baselines on these augmented instances.

Table 2: Test accuracy (%) under class-dependent PLL settings. Results enhanced by UMiP are shaded in gray, with ↑ denoting improvement.

Dataset	q_2	PRODEN	LWS	CRDPLL	PiCO	ABLE	DIRK	PaPi
CIFAR-10	0.7	88.18±0.5	88.27±0.6	88.01±0.6	88.26±0.6	88.04±0.3	83.24±0.5	89.02±0.4
		89.19±0.3↑	89.44±0.5↑	88.97±0.5↑	88.82±0.4↑	88.92±0.3↑	86.17±0.2↑	89.52±0.2↑
	0.8	86.73±0.6	87.68±0.3	85.72±0.4	87.21±0.5	86.10±0.2	82.48±0.3	86.23±0.4
		87.90±0.5↑	88.39±0.5↑	86.33±0.4↑	88.50±0.5↑	86.42±0.2↑	85.12±0.2↑	86.83±0.3↑
	0.9	75.95±1.0	83.88±0.3	57.03±0.6	82.45±1.0	82.32±0.2	80.89±0.3	78.98±0.6
		80.19±0.4↑	85.07±0.4↑	58.07±1.1↑	83.41±2.1↑	83.22±0.2↑	81.89±0.2↑	80.03±0.2↑
SVHN	0.7	94.52±0.2	94.76±0.3	96.78±0.2	94.54±0.1	94.68±0.3	93.23±0.4	96.71±0.2
		95.58±0.1↑	95.63±0.1↑	97.01±0.2↑	95.42±0.1↑	95.44±0.3↑	94.32±0.3↑	96.84±0.1↑
	0.8	94.48±0.6	94.74±0.2	96.78±0.1	94.55±0.3	94.65±0.1	93.15±0.5	95.87±0.3
		95.74±0.2↑	95.77±0.2↑	97.02±0.1↑	95.30±0.1↑	95.15±0.1↑	94.08±0.3↑	96.15±0.1↑
	0.9	94.47±0.4	94.76±0.3	96.41±0.2	84.02±1.1	94.45±0.2	93.45±0.2	95.81±0.1
		95.65±0.1↑	95.58±0.1↑	96.80±0.2↑	85.58±3.0↑	95.05±0.1↑	93.84±0.1↑	96.05±0.1↑

Table 3: Test accuracy (%) under uniform PLL settings. Results enhanced by UMiP are shaded in gray, with ↑ denoting improvement.

Dataset	q_3	PRODEN	LWS	CRDPLL	PiCO	ABLE	DIRK	PaPi
CIFAR-10	0.3	89.34±0.3	89.63±0.2	90.26±0.2	89.13±0.8	90.07±0.1	89.41±0.2	90.11±0.1
		89.89±0.4↑	90.10±0.5↑	90.61±0.4↑	89.59±0.3↑	90.35±0.1↑	89.86±0.1↑	90.32±0.1↑
	0.5	87.97±0.6	88.30±0.5	89.06±0.1	87.77±0.7	88.15±0.3	87.99±0.1	89.93±0.1
		89.62±0.4↑	89.49±0.1↑	89.44±0.3↑	89.21±0.2↑	88.72±0.1↑	89.32±0.1↑	90.16±0.1↑
	0.7	85.79±0.1	86.42±0.2	86.24±0.4	82.24±0.6	86.38±0.3	84.57±0.3	85.77±0.2
		87.15±0.8↑	87.64±0.3↑	87.15±0.3↑	85.66±1.1↑	86.94±0.1↑	86.91±0.5↑	87.58±0.2↑
SVHN	0.3	94.91±0.3	94.99±0.1	96.93±0.1	94.28±0.2	95.31±0.1	96.99±0.1	96.95±0.0
		95.65±0.3↑	95.44±0.2↑	97.10±0.1↑	95.40±0.1↑	95.45±0.4↑	97.02±0.0↑	97.12±0.1↑
	0.5	94.20±0.6	94.50±0.3	96.91±0.2	94.07±0.1	94.88±0.1	96.52±0.1	96.89±0.0
		95.61±0.2↑	95.62±0.1↑	97.03±0.3↑	95.26±0.1↑	95.31±0.1↑	96.61±0.1↑	97.13±0.0↑
	0.7	92.71±1.1	93.25±0.4	96.88±0.1	93.27±1.4	93.97±0.3	95.51±0.2	96.66±0.1
		95.19±0.3↑	95.32±0.3↑	97.12±0.1↑	94.84±1.1↑	94.72±0.1↑	96.82±0.6↑	96.68±0.1↑
CIFAR-100	0.01	75.35±0.8	74.58±0.4	78.05±0.7	74.62±0.4	75.81±0.2	77.41±0.6	78.10±0.3
		77.28±0.5↑	76.54±0.8↑	78.47±0.5↑	76.49±1.1↑	77.20±0.2↑	78.36±0.3↑	79.12±0.2↑
	0.05	74.88±1.1	74.02±0.6	77.16±0.5	74.21±0.3	75.98±0.4	76.98±0.1	77.01±0.2
		76.35±0.2↑	76.49±0.7↑	77.89±0.3↑	75.03±0.6↑	76.19±0.1↑	77.30±0.1↑	77.21±0.2↑
	0.1	70.21±0.2	69.13±0.8	72.85±0.1	62.88±1.9	68.96±1.1	68.31±0.2	71.05±0.1
		71.78±0.7↑	71.21±0.3↑	73.20±0.7↑	66.95±1.3↑	71.08±0.5↑	70.74±0.2↑	72.01±0.1↑
CUB-200	0.01	53.44±1.2	57.06±5.6	66.17±2.4	63.14±1.8	63.95±1.7	64.12±0.2	66.07±1.0
		60.80±1.0↑	61.63±3.1↑	71.00±1.3↑	67.19±1.1↑	67.45±1.2↑	69.62±0.2↑	71.31±1.0↑
	0.02	50.86±2.1	51.48±2.0	55.45±3.2	52.37±2.0	53.01±1.5	53.48±1.1	55.69±1.6
		55.90±1.0↑	58.01±1.3↑	66.69±2.9↑	59.68±1.5↑	57.71±1.1↑	58.42±0.3↑	65.24±1.2↑
	0.03	31.68±3.3	37.88±1.3	46.93±3.8	31.53±2.0	31.35±2.5	39.14±2.1	47.14±1.1
		44.62±1.2↑	48.08±1.1↑	48.91±1.2↑	42.82±1.9↑	45.49±1.1↑	45.18±2.0↑	47.28±1.1↑

Table 4: Test accuracy (%) on real-world datasets comparing original baselines against PLDA-augmented, and UMiP-augmented counterparts. Best results are in **bold**.

Method		Lost	BirdSong	MSRCv2	Soccer	Yahoo!News	Mirflicker
PRODEN	Original	71.15±2.2	69.88±1.2	51.71±1.6	56.59±0.4	67.36±1.0	67.39±0.5
	+ PLDA	71.19±1.7	70.34±0.3	52.03±0.3	56.98±0.2	67.48±0.3	68.24±0.9
	+ UMiP	**73.81±1.5**	**70.72±0.4**	**52.84±0.9**	**57.03±0.2**	**67.96±0.3**	**68.79±0.9**
LWS	Original	72.21±1.9	70.00±0.6	52.16±1.8	56.52±0.2	67.66±0.8	67.27±1.5
	+ PLDA	74.12±2.5	70.97±0.9	53.61±1.9	56.95±0.3	68.05±0.4	67.99±1.2
	+ UMiP	**76.28±2.1**	**71.60±0.8**	**55.34±1.5**	**57.87±0.3**	**68.10±0.3**	**68.85±1.1**

Implementation Details. Our implementation is built upon PyTorch [31]. For all methods, we used the Adam optimizer with a momentum of 0.9, a weight de-

cay of 1e-5, and a learning rate of 1e-3. The number of training epochs was set to 400. For the backbone model, we used ResNet-32 for benchmark datasets, and a multilayer perceptron with a hidden layer width of 500 and ReLU activation for real-world PLL datasets. For each case, we report the mean and standard deviation over three independent runs with different random seeds. Full hyperparameter configurations for all baselines are provided in Appendix B.4.

5.3 Experimental Results

Tabs. 1–4 report the performance of original baselines and UMiP-augmented counterparts across different datasets, yielding the following observations:

Consistent performance gains. As shown in Tab. 3, UMiP consistently improves all baselines, with especially notable gains under high ambiguity. For example, on the uniform CUB-200 dataset, LWS gains **4.57**%, **6.53**%, and **10.2**% as ambiguity rises. These gains are most pronounced in the instance-dependent settings where candidate labels are more closely coupled with instance features. For instance, on the instance-dependent SVHN dataset at $q_1 = 1.0$, UMiP improves PRODEN's accuracy from 83.74% to 95.64%.

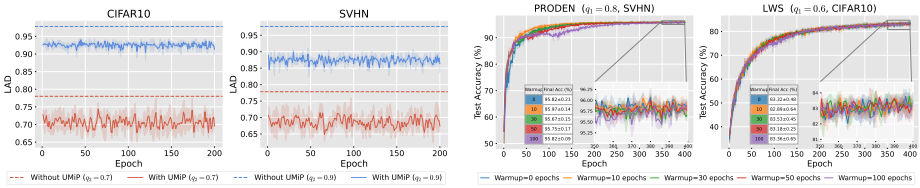
Comparison with PLDA. Tab. 4 compares UMiP with the adapted PLDA on real-world datasets. While PLDA yields gains, it consistently falls short of UMiP. This gap stems from PLDA's inherent limitations: poor scalability on high-dimensional data and the lack of joint optimization in its decoupled pipeline. In contrast, UMiP seamlessly integrates with deep models to boost performance, underscoring the necessity of augmentation tailored to deep PLL.

Simple integration, strong gains. UMiP delivers substantial performance improvements via simple integration with existing PLL methods, without the need to redesign complex architectures or incur substantial computational overhead. For instance, on the instance-dependent SVHN dataset at $q_1 = 0.6$ in Tab. 1 and Fig. 4, UMiP boosts PRODEN from 91.79% to 95.82%, surpassing the state-of-the-art ABLE (95.02%) with negligible extra cost.

5.4 Quantitative Analysis

Computational overhead analysis. As shown in Fig. 4, under identical settings on a V100 GPU, PRODEN+UMiP achieves state-of-the-art accuracy while being faster and lighter than all redesigned PLL methods. Specifically, it adds only two lightweight addition and division steps, i.e., Eqs. (3-7), without introducing learnable parameters, thus incurring minimal additional training time.

Dynamic evolution of LAD throughout training. Beyond the static reduction reported above, we further track how LAD evolves during training on class-dependent CIFAR-10 (q_2=0.7) and SVHN (q_2=0.9), as shown in Fig. 3(a,b). Without UMiP augmentation, LAD stays essentially flat across epochs (dashed lines). In contrast, with UMiP augmentation, LAD consistently fluctuates around a markedly lower level throughout the entire training process (solid lines). Notably, the epoch-to-epoch variation stems from the stochastic mixing operations



(a) CIFAR-10 (q_2=0.7) (b) SVHN (q_2=0.9) (c) PRODEN (d) LWS

Fig. 3: Training dynamics of UMiP. (a)–(b) Dynamic evolution of LAD throughout training on class-dependent CIFAR-10 (q_2=0.7) and SVHN (q_2=0.9), with and without UMiP augmentation. (c)–(d) Warm-up ablation: convergence and final test accuracy when UMiP is enabled after 0/10/30/50/100 warm-up epochs.

rather than from the model training itself, indicating that UMiP persistently eases the disambiguation difficulty of the dataset.

The effect of regularization exhibits a distinct impact. As illustrated in Fig. 5, some baseline methods tend to overfit at high ambiguity levels. UMiP augmentation mitigates this issue through its regularizing effect, helping these methods maintain robust generalization. On the other hand, reliability diagrams reveal that the original models tend to be overconfident (ECE = 11.01) when predicting samples, while the UMiP-augmented model reduces the error to 5.54, demonstrating its regularization in uncertainty calibration.

Significant reduction in LAD. Tab. 5 quantifies the LAD on CIFAR-10 *before* and *after* applying UMiP augmentation. As shown, UMiP consistently reduces LAD across all levels and settings, decreasing even from the high ambiguity level (0.993) to a value (0.974) close to the original low ambiguity level (0.973). Crucially, this reduction is achieved without shrinking candidate sets, thus avoiding the risk of discarding the true label.

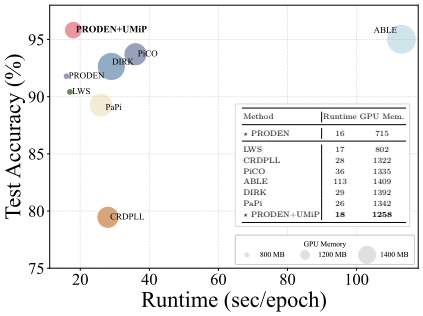


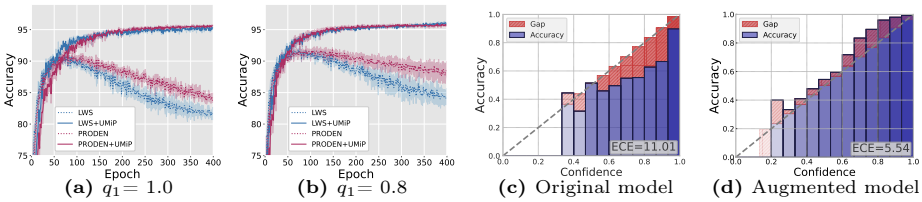
Fig. 4: Computational overheads.

5.5 Ablation Study

The importance of the renormalization step. To assess the effectiveness of the renormalization step in our method, we compare model performance on

Table 5: Magnitude change of local ambiguity degree.

Ambiguity Level	Metric	Low	Medium	High
Class-dependent 0.7/0.8/0.9	LAD (Before)	0.780	0.869	0.978
	LAD (After)	**0.706**	**0.851**	**0.925**
Instance-dependent 0.6/0.8/1.0	LAD (Before)	0.973	0.987	0.993
	LAD (After)	**0.958**	**0.969**	**0.974**



(a) $q_1 = 1.0$ **(b)** $q_1 = 0.8$ **(c)** Original model **(d)** Augmented model

Fig. 5: Generalization analysis of original and *UMiP*-augmented models on the instance-dependent SVHN dataset. (a–b) Test accuracy under different ambiguity levels. (c–d) Reliability diagrams and expected calibration error (ECE) for original and *UMiP*-augmented models.

the uniform PLL dataset at $q_3 = 0.9$. Fig. 6(a) shows that incorporating renormalization consistently improves baselines, achieving accuracy gains of up to **8.85**% (e.g., LWS on the SVHN dataset). Furthermore, removing the renormalization step degrades performance and may even lead to training failures (e.g., PRODEN on the CIFAR-10 dataset), confirming its necessity in our method.

Robustness to early-stage label confidences. A natural concern is that mixing unreliable label confidences in the early training stage might induce confirmation bias. To examine this, we run the original PLL method for 0/10/30/50/100 warm-up epochs before enabling UMiP, and report the resulting convergence and final accuracy in Fig. 3(c,d). The curves are near-identical regardless of the warm-up length, confirming that UMiP is robust to early-stage training. We attribute this stability to three factors: **(i)** dominant mixing keeps the paired sample a mild perturbation; **(ii)** renormalization confines the probability mass within the original candidate set; and **(iii)** the near-uniform early-stage confidences make mixing resemble label smoothing rather than aggressive self-training.

The effect of Beta parameter α. We evaluate UMiP under varying Beta distribution parameters α, including $\alpha = 0.5$ and $\alpha = 2.0$ in Fig. 6(b). In our experimental setup, we defaultly use $\alpha = 1$ so that the mixing coefficient λ is uniform on $[0.5, 1]$. With $\alpha = 0.5$, λ skews toward larger values, retaining more of the original sample; with $\alpha = 2.0$, toward smaller values, retaining less original information. As shown, both excessive and insufficient mixing limit UMiP's effectiveness, whereas moderate mixing ($\alpha = 1$) yields the best results.

The sensitivity to k and clusterability. Although the k-NN clusterability assumption expectedly holds for well-separated data, the LAD reduction is still achieved even under boundary overlaps. Tab. 6 reports neighborhood label purity $\mathcal{P}_k$ (the fraction of k-nearest neighbors sharing the same true class) and LAD reduction for varying k, under both well-separated and boundary-overlapping conditions. UMiP consistently reduces LAD across all k and conditions.

The effect of semantic information in the mixing process. To assess the role of semantic content in mixing, we interpolate between UMiP–Cutout and UMiP–CutMix by filling the Cutout hole with pixels from a counterpart image scaled by $\omega \in [0, 1]$. When $\omega = 0$, the operation reduces to vanilla Cutout; when $\omega = 1$, it becomes a CutMix-style UMiP. A large value of ω indicates that more

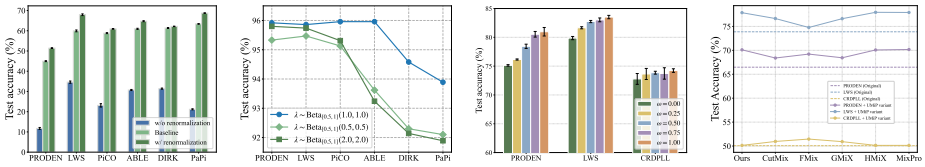


Fig. 6: (a)Renormalization ablation: test accuracy on the CIFAR-10 uniform PLL datasets at $q_3 = 0.9$; (b) Beta distribution ablation: test accuracy on the instance-dependent SVHN dataset at $q_1 = 0.6$. (c) Effect of semantic information weighted by ω. (d) Effect of different mixing strategies.

semantic information is included in the perturbation. As shown in Fig. 6(c), performance improves as the perturbation carries more semantic content, indicating that semantically meaningful perturbations benefit PLL augmentation.

Table 6: LAD reduction on the instance-dependent CIFAR-10 dataset (q_1=1.0).

Condition	Metric	k=5	k=10	k=20	k=50	k=100	k=200
Well-separated	$\mathcal{P}_k$	0.9948	0.9948	0.9948	0.9948	0.9948	0.9947
	LAD (Before)	0.9987	0.9973	0.9954	0.9927	0.9906	0.9883
	LAD (After)	**0.9722**	**0.9620**	**0.9547**	**0.9491**	**0.9469**	**0.9455**
Boundary overlaps	$\mathcal{P}_k$	0.7791	0.7753	0.7719	0.7669	0.7622	0.7565
	LAD (Before)	0.9761	0.9648	0.9563	0.9488	0.9451	0.9422
	LAD (After)	**0.9525**	**0.9358**	**0.9236**	**0.9139**	**0.9096**	**0.9065**

Influence of mixing strategies. In our method, we introduce perturbation through full pixel mixing, as detailed in Eq. (3). The same effect can also be achieved with other mixing-based methods such as CutMix [56], FMix [18], GMix/HMix [30] and MixPro [60]. To explore this further, we implement our method with different mixing strategies in Fig. 6(d). All consistently improve performance, with full pixel mixing yielding slightly better results.

6 Conclusion

In this work, we investigated an underexplored data augmentation problem in deep PLL and proposed a PLL-specific, LAD-aware augmentation method called UMiP, which perturbs only the input space to weaken neighborhood coupling while keeping candidate-label sets unchanged. This effectively makes PLL dataset easier to learn through lowering the local ambiguity degree. Theoretically, UMiP yields both ambiguity reduction and implicit regularization effects, which together facilitate effective disambiguation. Empirically, extensive experiments on both benchmark and real-world PLL datasets show consistent improvements over state-of-the-art PLL methods.

Acknowledgements

We sincerely thank the anonymous reviewers and the area chairs for their constructive comments and suggestions, which have greatly helped improve this work. The first author would like to thank Wei Wang (RIKEN AIP) for the valuable insights and discussions on this work.

References

1. Berthelot, D., Carlini, N., Goodfellow, I.J., Papernot, N., Oliver, A., Raffel, C.: Mixmatch: A holistic approach to semi-supervised learning. In: Advances in Neural Information Processing Systems. pp. 5050–5060 (2019)
2. Briggs, F., Fern, X.Z., Raich, R.: Rank-loss support instance machines for MIML instance annotation. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 534–542 (2012)
3. Carratino, L., Cissé, M., Jenatton, R., Vert, J.P.: On mixup regularization. Journal of Machine Learning Research **23**(325), 1–31 (2022)
4. Chen, C., Patel, V.M., Chellappa, R.: Learning from ambiguously labeled face images. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(7), 1653–1667 (2018)
5. Cour, T., Sapp, B., Taskar, B.: Learning from partial labels. The Journal of Machine Learning Research **12**, 1501–1536 (2011)
6. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 702–703 (2020)
7. Cubuk, E.D., Zoph, B., Mané, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation policies from data. CoRR **abs/1805.09501** (2018)
8. Devries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. CoRR **abs/1708.04552** (2017)
9. Fan, J., Huang, L., Gong, C., You, Y., Gan, M., Wang, Z.: Kmt-pll: K-means cross-attention transformer for partial label learning. IEEE Transactions on Neural Networks and Learning Systems **36**(2), 2789–2800 (2025)
10. Fan, J., Li, J., Huang, L., Gan, M., Chen, C.P.: Partial label zero-shot learning with semantic mining and instance–label alignment. Engineering Applications of Artificial Intelligence **177**, 114854 (2026)
11. Fan, J., Li, J., Zhong, X., Ren, K., Jiang, Z., Gan, M., Gu, T., Huang, L.: Evidential deep partial label learning to quantify disambiguation uncertainty. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24770–24779 (2026)
12. Fan, J., Zhong, X., Ren, K., Li, J., Huang, L., Gan, M., Chen, C.P.: Diffusion disambiguation models for partial label learning. IEEE Transactions on Circuits and Systems for Video Technology (2026)
13. Feng, L., Lv, J., Han, B., Xu, M., Niu, G., Geng, X., An, B., Sugiyama, M.: Provably consistent partial-label learning. In: Advances in Neural Information Processing Systems (2020)
14. Gao, Y., Zhang, M.: Discriminative complementary-label learning with weighted loss. In: International Conference on Machine Learning. vol. 139, pp. 3587–3597 (2021)

15. Gong, X., Bisht, N., Xu, G.: Does label smoothing help deep partial label learning? In: International Conference on Machine Learning (2024)
16. Guillaumin, M., Verbeek, J., Schmid, C.: Multiple instance metric learning from automatically labeled bags of faces. In: European Conference on Computer Vision. vol. 6311, pp. 634–647 (2010)
17. Gururani, S., Lerch, A.: Semi-supervised audio classification with partially labeled data. In: IEEE International Symposium on Multimedia. pp. 111–114 (2021)
18. Harris, E., Marcu, A., Painter, M., Niranjan, M., Prügel-Bennett, A., Hare, J.S.: Understanding and enhancing mixed sample data augmentation. CoRR **abs/2002.12047** (2020)
19. He, S., Yang, G., Feng, L.: Candidate-aware selective disambiguation based on normalized entropy for instance-dependent partial-label learning. In: IEEE/CVF International Conference on Computer Vision. pp. 1792–1801 (2023)
20. Huiskes, M.J., Lew, M.S.: The mirflickr retrieval evaluation. In: ACM international conference on Multimedia information retrieval. pp. 39–43 (2008)
21. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
22. Liu, L., Dietterich, T.G.: A conditional multinomial mixture model for superset label learning. In: Advances in Neural Information Processing Systems. pp. 557–565 (2012)
23. Luo, J., Orabona, F.: Learning from candidate labeling sets. In: Advances in Neural Information Processing Systems. pp. 1504–1512 (2010)
24. Lv, J., Liu, Y., Xia, S., Xu, N., Xu, M., Niu, G., Zhang, M.L., Sugiyama, M., Geng, X.: What makes partial-label learning algorithms effective? In: Advances in Neural Information Processing Systems. vol. 37, pp. 89513–89534 (2024)
25. Lv, J., Xu, M., Feng, L., Niu, G., Geng, X., Sugiyama, M.: Progressive identification of true labels for partial-label learning. In: International Conference on Machine Learning. vol. 119, pp. 6500–6510 (2020)
26. Mai, T.H., Chiang, C.K., Shih, H.H., Niu, G., Sugiyama, M., Lin, H.T.: Embracing biased transition matrices for complementary-label learning with many classes. arXiv preprint arXiv:2605.15586 (2026)
27. Mai, T.H., Lin, H.T.: Intra-cluster mixup: An effective data augmentation technique for complementary-label learning. arXiv preprint arXiv:2509.17971 (2025)
28. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS workshop on deep learning and unsupervised feature learning. vol. 2011, p. 4 (2011)
29. Nishi, K., Ding, Y., Rich, A., Höllerer, T.: Augmentation strategies for learning with noisy labels. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 8022–8031 (2021)
30. Park, C., Yun, S., Chun, S.: A unified analysis of mixed sample data augmentation: A loss function perspective. In: Advances in neural information processing systems. vol. 35, pp. 35504–35518 (2022)
31. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E.Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. pp. 8024–8035 (2019)
32. Qiao, C., Xu, N., Geng, X.: Decompositional generation process for instance-dependent partial label learning. In: The Eleventh International Conference on Learning Representations (2023)

33. Shi, Y., Wu, D., Geng, X., Zhang, M.: Robust representation learning for unreliable partial label learning. CoRR **abs/2308.16718** (2023)
34. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research **15**(1), 1929–1958 (2014)
35. Sugiyama, M., Bao, H., Ishida, T., Lu, N., Sakai, T., Niu, G.: Machine learning from weak supervision: An empirical risk minimization approach. MIT Press (2022)
36. Tian, S., Wei, H., Wang, Y., Feng, L.: Crosel: Cross selection of confident pseudo labels for partial-label learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19479–19488 (2024)
37. Tian, Y., Yu, X., Fu, S.: Partial label learning: Taxonomy, analysis and outlook. Neural Networks **161**, 708–734 (2023)
38. Wager, S., Wang, S., Liang, P.: Dropout training as adaptive regularization. In: Advances in Neural Information Processing Systems. pp. 351–359 (2013)
39. Wang, D., Feng, L., Zhang, M.: Learning from complementary labels via partial-output consistency regularization. In: International Joint Conference on Artificial Intelligence. pp. 3075–3081 (2021)
40. Wang, H., Xiao, R., Li, Y., Feng, L., Niu, G., Chen, G., Zhao, J.: Pico: Contrastive label disambiguation for partial label learning. In: International Conference on Learning Representations (2022)
41. Wang, H., Yang, S., Lyu, G., Liu, W., Hu, T., Chen, K., Feng, S., Chen, G.: Deep partial multi-label learning with graph disambiguation. In: International Joint Conference on Artificial Intelligence. pp. 4308–4316 (2023)
42. Wang, W., Wu, D.D., Wang, J., Niu, G., masashi, S.: PLENCH: Realistic evaluation of deep partial-label learning algorithms. In: The Thirteenth International Conference on Learning Representations (2025)
43. Wang, W., Zhang, M.: Partial label learning with discrimination augmentation. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 1920–1928 (2022)
44. Wei, C., Kakade, S., Ma, T.: The implicit and explicit regularization effects of dropout. In: International conference on machine learning. pp. 10181–10192 (2020)
45. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-ucsd birds 200 (2010)
46. Wen, H., Cui, J., Hang, H., Liu, J., Wang, Y., Lin, Z.: Leveraged weighted loss for partial label learning. In: International Conference on Machine Learning. vol. 139, pp. 11091–11100 (2021)
47. Wu, D.D., Cui, J., Wang, W., Shen, Z., Sugiyama, M.: Delta: Robustly training diffusion models with weak annotations. In: ICLR 2026 2nd Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy
48. Wu, D.D., Fu, C., Wu, W., Xia, W., Zhang, X., Zhou, J., Zhang, M.L.: Efficient model stealing defense with noise transition matrix. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 24305–24315 (2024)
49. Wu, D., Wang, D., Zhang, M.: Revisiting consistency regularization for deep partial label learning. In: International Conference on Machine Learning. vol. 162, pp. 24212–24225 (2022)
50. Wu, D., Wang, D., Zhang, M.: Distilling reliable knowledge for instance-dependent partial label learning. In: AAAI Conference on Artificial Intelligence. pp. 15888–15896 (2024)

51. Xia, S., Lv, J., Xu, N., Geng, X.: Ambiguity-induced contrastive learning for instance-dependent partial label learning. In: International Joint Conference on Artificial Intelligence. pp. 3615–3621 (2022)
52. Xia, S., Lv, J., Xu, N., Niu, G., Geng, X.: Towards effective visual representations for partial-label learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15589–15598 (2023)
53. Xu, N., Liu, B., Lv, J., Qiao, C., Geng, X.: Progressive purification for instance-dependent partial label learning. In: International Conference on Machine Learning. vol. 202, pp. 38551–38565 (2023)
54. Xu, N., Qiao, C., Geng, X., Zhang, M.: Instance-dependent partial label learning. In: Advances in Neural Information Processing Systems. pp. 27119–27130 (2021)
55. Xu, Y., Shen, Y., Wei, X., Yang, J.: Webly-supervised fine-grained recognition with partial label learning. In: International Joint Conference on Artificial Intelligence. pp. 1502–1508 (2022)
56. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: IEEE/CVF international conference on computer vision. pp. 6023–6032 (2019)
57. Zeng, Z., Xiao, S., Jia, K., Chan, T., Gao, S., Xu, D., Ma, Y.: Learning by associating ambiguously labeled images. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 708–715 (2013)
58. Zhang, F., Feng, L., Han, B., Liu, T., Niu, G., Qin, T., Sugiyama, M.: Exploiting class activation value for partial-label learning. In: International Conference on Learning Representations (2022)
59. Zhang, H., Cissé, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018)
60. Zhao, Q., Huang, Y., Hu, W., Zhang, F., Liu, J.: Mixpro: Data augmentation with maskmix and progressive attention labeling for vision transformer. In: The Thirteenth International Conference on Learning Representations (2023)
61. Zhou, D., Zhang, Z., Zhang, M., He, Y.: Weakly supervised POS tagging without disambiguation. ACM Transactions on Asian and Low-Resource Language Information Processing **17**(4), 35:1–35:19 (2018)
62. Zhu, Z., Song, Y., Liu, Y., Liu, Y.: Clusterability as an alternative to anchor points when learning with noisy labels. In: International Conference on Machine Learning. vol. 139, pp. 12912–12923 (2021)