# Appeal: Allow Mislabeled Samples the Chance to be Rectified in Partial Label Learning

Chongjie Si, Xuehui Wang, Yan Wang, Xiaokang Yang, *Fellow, IEEE*, Wei Shen

**Abstract**—In partial label learning (PLL), each instance is associated with a set of candidate labels among which only one is ground-truth. The majority of the existing works focuses on constructing robust classifiers to estimate the labeling confidence of candidate labels in order to identify the correct one. However, these methods usually struggle to identify and rectify mislabeled samples. To help these mislabeled samples "appeal" for themselves and help existing PLL methods identify and rectify mislabeled samples, in this paper, we propose the first appeal-based PLL framework. Specifically, we introduce a novel partner classifier and instantiate it predicated on the implicit fact that non-candidate labels of a sample should not be assigned to it, which is inherently accurate and has not been fully investigated in PLL. Furthermore, a novel collaborative term is formulated to link the base classifier and the partner one. During each stage of mutual supervision, both classifiers will blur each other's predictions through a blurring mechanism to prevent overconfidence in a specific label. Extensive experiments demonstrate that the appeal and disambiguation ability of several well-established stand-alone and deep-learning based PLL approaches can be significantly improved by coupling with this learning paradigm.

**Index Terms**—Partial label learning, appeal, disambiguation

✦

## 1 INTRODUCTION

In 2002, Jin and Ghahramani [1] pioneered the concept of Partial Label Learning (PLL), a groundbreaking framework in which each instance is associated with multiple candidate labels, among which only one represents the actual ground-truth. Over the past two decades, the field of PLL has experienced remarkable growth [1]–[10], driven by the increasing necessity to accurately discern the valid label from a set of potential candidates in diverse real-world applications. A notable example, as shown in Fig. 1, is the automatic face naming task [11], [12], where each facial image extracted from various media is linked to a list of names derived from corresponding titles or captions [13]. Another salient application is facial age estimation: for each human face, the ages annotated by crowd-sourcing labelers are considered as candidate labels [14].

Evidently, the principal challenge in PLL lies in the inherent obscurity of the ground-truth label, which conceals within the candidate labels and remains inaccessible during the training phase. To surmount this obstacle, the most extensively researched and pivotal strategy in PLL-*Disambiguation*-emerges as critical. Existing methods predominantly achieve disambiguation by differentiating the labeling confidences of each candidate label to identify the ground truth. This process typically relies on an alternative and iterative optimization algorithm for updating the classifier's parameters. For instance, LALO [15] implements

Fig. 1. Partial label learning in the automatic face naming task. The faces in this image can be automatically detected by a face detector, and then each face is assigned with the candidate name label set extracted from the script: Jennifer Aniston, Courteney Cox, Lisa Kudrow, Matthew Perry, David Schwimmer, and Matt LeBlanc, where only one is ground-truth.

constrained local consistency to differentiate the candidate labels, PL-AGGD [16] employs a similarity graph for effective disambiguation and PL-CL [17] adopts complementary information to help disambiguation. However, a significant yet rarely studied question comes to the front: *can the instances that are incorrectly disambiguated, i.e., mislabeled, have the opportunity to be rectified?* More specifically, *can a classifier correct a false positive candidate label (i.e., invalid candidate label) with a large or upward-trending labeling confidence at a later stage?* To explore this question, we conduct experiments on a real-world data set Lost [18] and record the labeling confidences of several candidate labels generated by PL-AGGD [16] in each iteration, which is shown in Fig. 2. Our findings reveal some intriguing phenomena:

- Each candidate label's labeling confidence is likely to continually increase or decrease until convergence.
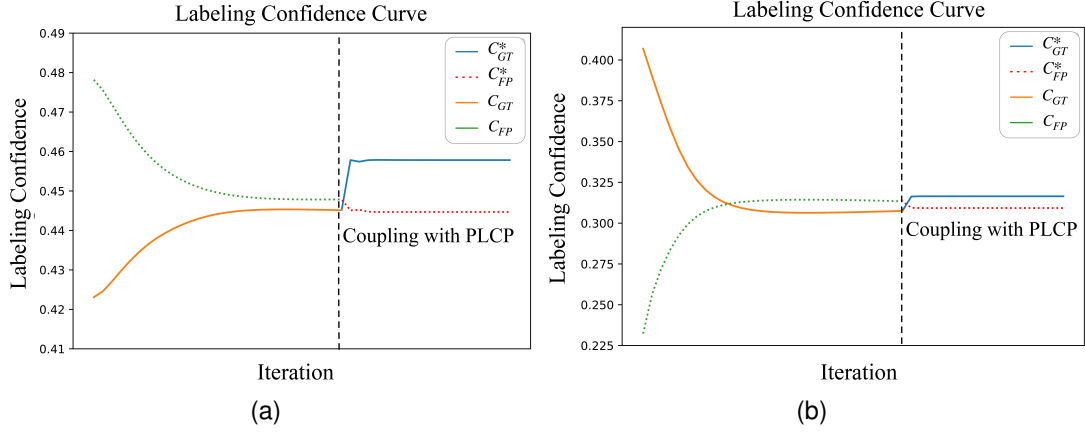- For a false positive candidate label with a large

Fig. 2. Two representative errors a typical PLL classifier, e.g., PL-AGGD [16] may make. $C_{GT}$ and $C^*_{GT}$ stand for the labeling confidence of the ground-truth label generated by PL-AGGD and PL-AGGD coupled with PLCP, and $C_{FP}$ and $C^*_{FP}$ stand for that of a false positive label predicted by PL-AGGD and PL-AGGD coupled with PLCP. (a). For a false positive candidate label with a large labeling confidence, although its confidence may decrease properly, it could still be larger than the ground-truth one's. (b). The labeling confidence of a false positive candidate label keeps increasing and becomes the largest, which misleads the final prediction. When coupled with PLCP, the labeling confidence of each candidate label generated by the partner classifier is adopted as the supervision to help PL-AGGD correct these errors, which results in a mutation in the figures.

labeling confidence, although its confidence may decrease in subsequent iterations, the confidence remains substantial and can easily lead to the incorrect identification of the ground truth label.

The observed phenomena suggest that once the labeling confidence of a false positive candidate label increases, it becomes difficult to decrease in the subsequent iterations. Furthermore, even if the confidence of a false positive candidate label decreases appropriately, it may still be recognized as the ground truth one, as its initial labeling confidence remains large and continues to be greater than the confidence of the ground truth label upon convergence. As a result, correcting mislabeled samples for a PLL classifier itself proves to be quite challenging.

We believe that these mislabeled samples should be afforded the opportunity to be corrected. In other words, these samples should have a chance to "appeal" on their behalf. In this case, the classifier's disambiguation ability can also be further enhanced. Consequently, in this paper, we introduce another major strategy in PLL: *Appeal*, and propose the first appeal-based framework.

Specifically, our framework PLCP, referring to **P**artial **L**abel Learning with a **C**lassifier as **P**artner, provides each instance with the chance to appeal. Given a classifier from any existed PLL approach as the base classifier, as shown in Fig. 3, PLCP integrates an additional partner classifier. This partner classifier assists the base classifier in identifying and rectifying mislabeled samples, offering more precise and complementary information to the base classifier, thereby enhancing disambiguation and fostering mutual supervision between the two classifiers. The design of the partner classifier is crucial in whether mislabeled samples could appeal for themselves, as the partner classifier's feedback significantly influences the base classifier's capability to recognize and amend mislabeled instances. Since the information in non-candidate labels, which indicates that a set of labels DO NOT belong to a sample, is typically more precise yet often overlooked by the majority of existing works, the partner

classifier is devised as a complementary classifier. This classifier specifies the labels that should not be assigned to a sample, thereby complementing the base classifier. Additionally, a collaborative term is also designed to link the base classifier and the partner classifier.

During mutual supervision, the labeling confidence is first updated based on the base classifier's modeling output. A blurring mechanism is then applied to this updated labeling confidence, introducing uncertainty. This could potentially diminish the high confidence of certain false positive candidate labels or elevate the low confidence of the actual ground truth. This updated labeling confidence subsequently serves as the supervision information to interact with the partner classifier, whose final output, in turn, supervises the base classifier. The predictions of the two classifiers, while distinct, are inextricably linked, enhancing the disambiguation ability of this paradigm in two opposing ways. With this mutual supervision paradigm, the instances with disambiguation errors have a higher likelihood to appeal successfully.

The preliminary results of this work are presented in [19]. In this paper, we refine the method in [19] and propose a more robust version of PLCP. Besides, comprehensive theoretical analyses of PLCP are provided in this paper. The rest of the paper is organized as follows. We first introduce the appeal-based framework PLCP in Sec. 2, and then present the experimental results and ablation study in Sec. 3. Further analysis is provided in Sec. 4. We then review some related works in PLL and other related fields in Sec. 5. Finally, conclusion is given in Sec. 6

## 2 APPEAL-BASED FRAMEWORK: PLCP

Let $\mathcal{X} = \mathbb{R}^q$ denote the $q$-dimensional feature space and $\mathcal{Y} = \{0, 1\}^l$ be the label space with $l$ classes. Given a partial label data set $\mathcal{D} = \{\mathbf{x}_i, S_i | 1 \leq i \leq n\}$ where $\mathbf{x}_i \in \mathcal{X}$, $S_i \subseteq \mathcal{Y}$ is the corresponding candidate label set and $n$ is the number of instances. The task of PLL is to induce a multi-class classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$ based on $\mathcal{D}$.

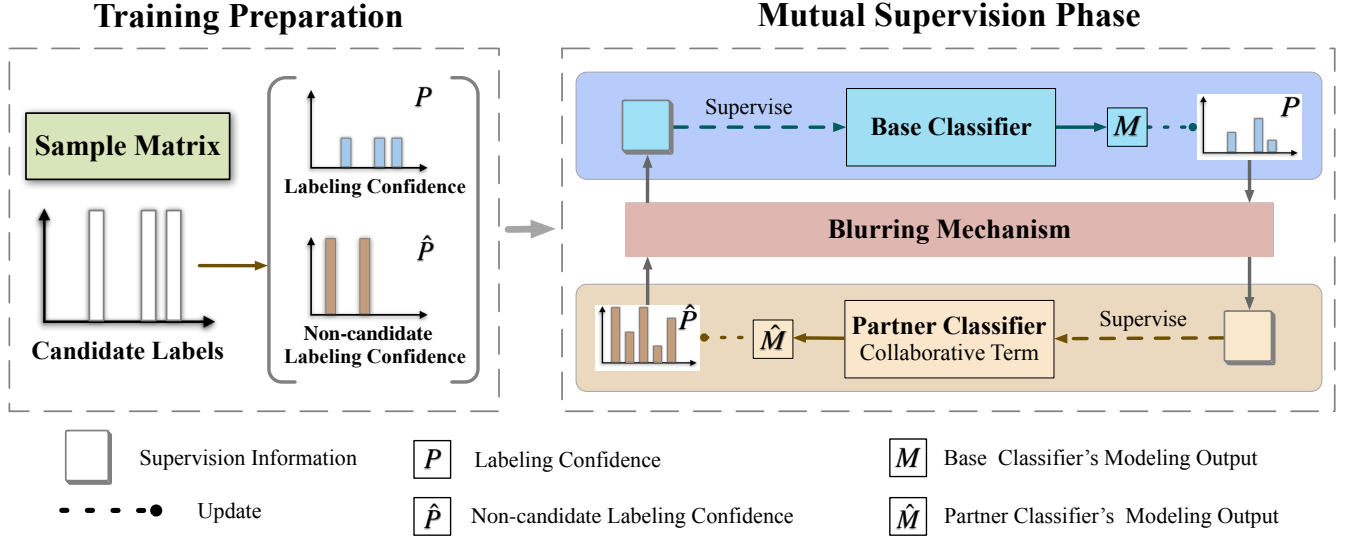## Training Preparation                    Mutual Supervision Phase



Fig. 3. The framework of PLCP. A partner classifier is constructed based on the non-candidate label information to enable mutual supervision between the base classifier and itself. In each stage of mutual supervision, the base classifier updates the labeling confidence $\mathbf{P}$ based on its modeling output $\mathbf{M}$ and blurs it through a blurring mechanism. Afterwards, the output is represented as the supervision information to interact with the partner classifier. The pipeline of the partner classifier is almost the same as the base classifier's.

As shown in Fig. 3, a partner classifier $\mathcal{P}$ is designed to complement a base classifier $\mathcal{B}$ and also can be supervised by $\mathcal{B}$. $\mathcal{B}$ represents any existing PLL classifier with fine generality and flexibility. In each stage of mutual supervision, the base classifier updates the labeling confidence based on its modeling outputs, and then a blurring mechanism is applied to further process it. Subsequently the output of the labeling confidence is taken as the supervision information for the partner classifier. The learning pipeline of the partner classifier closely mirrors that of the base classifier. The following subsections will provide further details on this process.

### 2.1  Base Classifier

Denote $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]^\mathsf{T} \in \mathbb{R}^{n \times q}$ the sample matrix with $n$ instances, and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y_2}, ..., \mathbf{y}_n]^\mathsf{T} \in \{0, 1\}^{n \times l}$ the partial label matrix, where $y_{ij} = 1$ (resp. $y_{ij} = 0$) if the $j$-th label of $\mathbf{x}_i$ resides in (resp. does not reside in) its candidate label set. Suppose $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n]^\mathsf{T} \in \mathbb{R}^{n \times l}$ is the labeling confidence matrix, where $\mathbf{p}_i \in \mathbb{R}^l$ represents the labeling confidence vector of $\mathbf{x}_i$, and $p_{ij}$ denotes the probability of the $j$-th label being the ground-truth label of $\mathbf{x}_i$. For the base classifier utilizing the labeling confidence strategy, $\mathbf{P}$ is initialized according to the base classifier. Generally, it is initialized as follows:

$$p_{ij} = \begin{cases} \frac{1}{\sum_j y_{ij}} & \text{if} \quad y_{ij} = 1 \\ 0 & \text{otherwise .} \end{cases} \quad (1)$$

The labeling confidence vector $\mathbf{p}_i$ typically satisfies the following constraints: $\sum_j p_{ij} = 1$, $0 \leq p_{ij} \leq y_{ij}$ [15], [16]. The first constrains $\mathbf{p}_i$ to be normalized, and the second indicates that only the confidence of a candidate label has a chance to be positive. It should be noted that the ideal state of $\mathbf{p}_i$ is one-hot.

Once the base classifier has been trained, its modeling output will be generated. Denote the modeling output matrix

$\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_n]^\mathsf{T} \in \mathbb{R}^{n \times l}$, where $\mathbf{m}_{ij}$ represents the probability of the $j$-th label being $\mathbf{x}_i$'s ground-truth label as predicted by the base classifier. It should be noted that $\mathbf{m}_i$ may either be a real-valued or one-hot vector depending on the specific base classifier. For instance, for SURE [20] $\mathbf{m}_i$ is real-valued, while for PL-KNN [21] it is one-hot. Afterwards, $\mathbf{P}$ is updated to $\mathbf{P}_1$ through the following equation:

$$\mathbf{P}_1 = \mathcal{T}_0 \left( \mathcal{T}_\mathbf{Y} \left( \alpha \mathbf{P} + (1 - \alpha)\mathbf{M} \right) \right), \quad (2)$$

where $\alpha$ is a hyper-parameter controlling the smoothness of the labeling confidence. $\mathcal{T}_0$, $\mathcal{T}_\mathbf{Y}$ are two thresholding operators in element-wise, i.e., $\mathcal{T}_0(a) := \max\{0, a\}$ with $a$ being a scalar and $\mathcal{T}_\mathbf{Y}(a) := \min\{y_{ij}, a\}$.

### 2.2  Blurring Mechanism

In the next step, a blurring mechanism is designed to further process the labeling confidence matrix $\mathbf{P}_1$ to $\mathbf{O}_1 \in \mathbb{R}^{n \times l}$ :

$$\begin{aligned} \mathbf{Q}_1 &= \mathcal{E}(e^k \mathbf{P}_1) \odot \mathbf{Y}, \\ [\mathbf{O}_1]_i &= \frac{[\mathbf{Q}_1]_i}{\|[\mathbf{Q}_1]_i\|}, \end{aligned} \quad (3)$$

where $\mathcal{E}(\cdot)$ is an element-wise operator, for a matrix $\mathbf{A} = [a_{ij}]_{n \times l} \in \mathbb{R}^{n \times l}$, $\mathcal{E}(\mathbf{A}) = [\exp(a_{ij})]_{n \times l}$. $k$ is a temperature parameter that controls the extent of blurring of labeling confidence. $\odot$ represents the Hadamard product of two matrices. For matrices $\mathbf{A}$ and $\mathbf{B}$ with same size $n \times l$, $\mathbf{A} \odot \mathbf{B} = [a_{ij} b_{ij}]_{n \times l}$. The Hadamard product allows only the candidate label has a positive confidence. We then normalize each row of $\mathbf{Q}_1$ and output the result $\mathbf{O}_1$.

The labeling confidences are expected to be blurred at each stage of mutual supervision to prevent being overconfident in some false positive labels, hence we set $k < 0$ and then normalize the results, which means two labeling confidences that differ significantly can also become close. For more details on why the blurring mechanism can blur the outputs, please refer to Sec. 4.4.

## 2.3 Partner Classifier

Since the partner classifier has significant impacts on the success of PLCP, designing an appropriate partner classifier is quite important. In order to better assist the base classifier, a classifier that specifies the labels that should not be assigned to a sample is instantiated as the partner classifier, since non-candidate label information is exactly accurate and opposite to the candidate label information, making it a valuable complement to the base classifier.

Denote the non-candidate label matrix $\hat{\mathbf{Y}} = [\hat{y}_{ij}]_{n \times l}$ where $\hat{y}_{ij} = 0$ (resp. $\hat{y}_{ij} = 1$) if the $j$-th label is (resp. is not) in the candidate label set of $\mathbf{x}_i$, and $\hat{\mathbf{P}} = [\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, ..., \hat{\mathbf{p}}_n]^{\mathsf{T}} \in \mathbb{R}^{n \times l}$ the non-candidate labeling confidence matrix, where $\hat{p}_{ij}$ represents the probability of the $j$-th label NOT being the ground-truth label of $\mathbf{x}_i$. Similar to the labeling confidence, $\hat{\mathbf{P}}$ is also constrained with two constraints: $\sum_j \hat{p}_{ij} = l - 1, \hat{y}_{ij} \leq \hat{p}_{ij} \leq 1$. The first term constrains the sum of the probability of each label being invalid is strictly $l - 1$, and the second indicates that only candidate label has a chance to update its non-candidate labeling confidence while the others keep 1 (invalid labels). Note that the ideal state of $\hat{\mathbf{p}}_i$ is "zero-hot", i.e., only one element in $\hat{\mathbf{p}}_i$ is 0 with others 1. $\hat{\mathbf{P}}$ is initialized as $\hat{\mathbf{Y}}$. Suppose $\hat{\mathbf{W}} = [\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, ..., \hat{\mathbf{w}}_q]^{\mathsf{T}} \in \mathbb{R}^{q \times l}$ is the weight matrix, the partner classifier is formulated as follows:

$$\min_{\hat{\mathbf{W}}, \hat{\mathbf{b}}, \mathbf{C}} \quad \left\| \mathbf{X}\hat{\mathbf{W}} + \mathbf{1}_n\hat{\mathbf{b}}^{\mathsf{T}} - \mathbf{C} \right\|_F^2 + \lambda \left\| \hat{\mathbf{W}} \right\|_F^2 \tag{4}$$
$$\text{s.t.} \quad \hat{\mathbf{Y}} \leq \mathbf{C} \leq \mathbf{1}_{n \times l}, \mathbf{C}\mathbf{1}_l = (l - 1)\mathbf{1}_n,$$

where $\hat{\mathbf{b}} = [\hat{b}_1, \hat{b}_2, ..., \hat{b}_l]^{\mathsf{T}} \in \mathbb{R}^l$ is the bias term, $\mathbf{1}_n \in \mathbb{R}^n$ is an all one vectors, $\mathbf{1}_{n \times l}$ is an all one matrix with size $n \times l$ and $\lambda$ is a hyper-parameters trading off these terms. $\|\hat{\mathbf{W}}\|_F$ is the Frobenius norm of the weight matrix. $\mathbf{C} \in \mathbb{R}^{n \times l}$ represents non-candidate labeling confidence, which is a temporary variable only used for optimization in the partner classifier. By solving this optimization problem, the partner classifier learns to specify which labels should not be assigned to each sample, improving the overall performance of the mutual supervision process.

## 2.4 The Collaborative Term

Since a label is either ground-truth or not and based on the constraints on $\mathbf{p}_i$ and $\hat{\mathbf{p}}_i$, the smallest value of $\mathbf{p}_i^{\mathsf{T}}\hat{\mathbf{p}}_i$ is obtained when $\hat{\mathbf{p}}_i$ is zero-hot ($\mathbf{p}_i$ is one-hot). Therefore, a collaborative relationship between the outputs of the partner classifier and the prior given by the base classifier can be formulated, with the partner classifier becoming

$$\min_{\hat{\mathbf{W}}, \hat{\mathbf{b}}, \mathbf{C}} \quad \left\| \mathbf{X}\hat{\mathbf{W}} + \mathbf{1}_n\hat{\mathbf{b}}^{\mathsf{T}} - \mathbf{C} \right\|_F^2 + \gamma \mathrm{tr}\left( \mathbf{O}_1\mathbf{C}^{\mathsf{T}} \right) + \lambda \left\| \hat{\mathbf{W}} \right\|_F^2$$
$$\text{s.t.} \quad \hat{\mathbf{Y}} \leq \mathbf{C} \leq \mathbf{1}_{n \times l}, \mathbf{C}\mathbf{1}_l = (l - 1)\mathbf{1}_n, \tag{5}$$

where $\gamma$ is a hyper-parameter. Different from candidate labels, the non-candidate labels are directly accessible and accurate, therefore the information learnt by the partner classifier is precise, which can effectively complement the base classifier.

The problem in Eq. (5) can be solved via an alternative and iterative manner in Sec. 2.6, which results in an optimal

---

**Algorithm 1** The pseudo code of PLCP

1: **Input:** A base classifier $\mathcal{B}$ and the partner classifier $\mathcal{P}$, partial label data $\mathcal{D}$, hyper-parameter $\alpha$, $\gamma$ and $\lambda$, temperature parameter $k$, the maximum number of iteration $iter$ and unseen sample $\mathbf{x}^*$.
2: **Output:** The predicted label $y^*$
3: Initialize $\mathbf{P}_0$, $\hat{\mathbf{P}}_0$ and $i = 1$.
4: **while** i≤iter **do**
5:     Train the base classifier supervised by $\hat{\mathbf{O}}_{i-1}$ and obtain the modeling output $\mathbf{M}_i$.
6:     $\mathbf{P}_i = \mathcal{T}_0\left( \mathcal{T}_{\mathbf{Y}}\left( \alpha\mathbf{P}_{i-1} + (1 - \alpha)\mathbf{M}_i \right) \right)$.
7:     $\mathbf{Q}_i = \mathcal{E}(e^k\mathbf{P}_i) \odot \mathbf{Y}$.
8:     Normalize $\mathbf{Q}_i$ and return $\mathbf{O}_i$.
9:     Train the partner classifier supervised by $\mathbf{O}_i$ and obtain the modeling output $\hat{\mathbf{M}}_i$ according to Eq. (6).
10:     $\hat{\mathbf{P}}_i = \mathcal{T}_1\left( \mathcal{T}_{\hat{\mathbf{Y}}}\left( \alpha\hat{\mathbf{P}}_{i-1} + (1 - \alpha)\hat{\mathbf{M}}_i \right) \right)$.
11:     $\hat{\mathbf{Q}}_i = \mathcal{E}(e^k(1 - \hat{\mathbf{P}}_i)) \odot \mathbf{Y}$.
12:     Normalize $\hat{\mathbf{Q}}_i$ and return $\hat{\mathbf{O}}_i$.
13:     **if** the stopping criterion is meet **then**
14:         **break;**
15:     **end if**
16:     $i = i + 1$.
17: **end while**
18: **Return:** The predicted label $\mathbf{y}^*$ according to Eq. (9).

---

weight matrix $\hat{\mathbf{W}}$ and bias term $\hat{\mathbf{b}}$. The modeling output $\hat{\mathbf{M}}$ for the training data is

$$\hat{\mathbf{M}} = \mathbf{X}\hat{\mathbf{W}} + \mathbf{1}_n\hat{\mathbf{b}}^{\mathsf{T}}. \tag{6}$$

Afterwards, the non-candidate labeling confidence $\hat{\mathbf{P}}$ is updated to $\hat{\mathbf{P}}_1$ following

$$\hat{\mathbf{P}}_1 = \mathcal{T}_1\left( \mathcal{T}_{\hat{\mathbf{Y}}}\left( \alpha\hat{\mathbf{P}} + (1 - \alpha)\hat{\mathbf{M}} \right) \right), \tag{7}$$

where $\mathcal{T}_1$, $\mathcal{T}_{\hat{\mathbf{Y}}}$ are two thresholding operators in element-wise, i.e., $\mathcal{T}_1(m) := \min\{1, m\}$ with $m$ being a scalar and $\mathcal{T}_{\hat{\mathbf{Y}}}(m) := \max\{\hat{y}_{ij}, m\}$. The non-candidate labeling confidence can be further processed as $\hat{\mathbf{O}}_1$ via the blurring mechanism:

$$\hat{\mathbf{Q}}_1 = \mathcal{E}(e^k(1 - \hat{\mathbf{P}}_1)) \odot \mathbf{Y},$$
$$[\hat{\mathbf{O}}_1]_i = \frac{[\hat{\mathbf{Q}}_1]_i}{\|[\hat{\mathbf{Q}}_1]_i\|}. \tag{8}$$

For convenience, we transform the non-candidate labeling confidence into labeling confidence in advance, and finally $\hat{\mathbf{Q}}_1$ is normalized as $\hat{\mathbf{O}}_1$, which is the supervision of the base classifier in the next iteration.

## 2.5 Stopping Criterion

Through continuous mutual supervision, the base classifier could identify and rectify mislabeled samples. This iterative process of mutual supervision persists for multiple iterations. The iteration ceases when there is a consistent correction of over 5% of mislabeled samples across two consecutive iterations.

For an unseen sample $\mathbf{x}^*$, suppose the non-candidate labeling confidence vector predicted by the partner classifier in the last iteration is $\hat{\mathbf{p}}^{pt}$. The prediction $y^*$ of PLCP is

$$y^* = \mathrm{argmax}_i \quad (1 - \hat{p}_i^{pt}), \tag{9}$$

Rewrite $\mathbf{P}_0 = \mathbf{P}$, $\hat{\mathbf{P}}_0 = \hat{\mathbf{P}}$, $\mathbf{H}_1 = \mathbf{H}$, $\hat{\mathbf{H}}_1 = \hat{\mathbf{H}}$ and $\hat{\mathbf{O}}_0 = \mathbf{P}$, the complete procedure of PLCP is shown in Algorithm 1.

### 2.6 Numerical Solution of the Partner Classifier

Th optimization problem in Eq. (5) has three variables with different constraints, therefore we can adopt an alternative and iterative optimization to solve it.

#### 2.6.1 Update $\hat{\mathbf{W}}$ and $\hat{\mathbf{b}}$

With $\mathbf{C}$ fixed, the problem w.r.t. $\hat{\mathbf{W}}$ and $\hat{\mathbf{b}}$ can be written as

$$\min_{\hat{\mathbf{W}}, \hat{\mathbf{b}}} \quad \left\| \mathbf{X}\hat{\mathbf{W}} + \mathbf{1}_n\hat{\mathbf{b}}^\top - \mathbf{C} \right\|_F^2 + \lambda \left\| \hat{\mathbf{W}} \right\|_F^2, \qquad (10)$$

which is a least square problem with the closed-form solution as

$$\hat{\mathbf{W}} = \left( \mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_{n \times n} \right) \mathbf{X}^\top\mathbf{C},$$
$$\hat{\mathbf{b}} = \frac{1}{n} \left( \mathbf{C}^\top\mathbf{1}_n - \hat{\mathbf{W}}^\top\mathbf{X}^\top\mathbf{1}_n \right), \qquad (11)$$

where $\mathbf{I}_{n \times n}$ is the identity matrix with the size $n \times n$.

#### 2.6.2 Update $\mathbf{C}$

With $\hat{\mathbf{W}}$ and $\hat{\mathbf{b}}$ fixed, the $\mathbf{C}$-subproblem can be formulated as

$$\min_{\mathbf{C}} \quad \left\| \mathbf{X}\hat{\mathbf{W}} + \mathbf{1}_n\hat{\mathbf{b}}^\top - \mathbf{C} \right\|_F^2 + \gamma\operatorname{tr}\left( \mathbf{O}_1\mathbf{C}^\top \right)$$
$$\text{s.t.} \quad \hat{\mathbf{Y}} \le \mathbf{C} \le \mathbf{1}_{n \times l}, \mathbf{C}\mathbf{1}_l = (l-1)\mathbf{1}_n. \qquad (12)$$

For simplicity, $\mathbf{O}_1$ is written as $\mathbf{O}$ and $\mathbf{J} = \mathbf{X}\hat{\mathbf{W}} + \mathbf{1}_n\hat{\mathbf{b}}^\top$. Notice that each row of $\mathbf{C}$ is independent to other rows, therefore the problem in Eq. (12) can be solved row by row:

$$\min_{\mathbf{C}_i} \quad \mathbf{C}_i^\top\mathbf{C}_i + (\gamma\mathbf{O}_i - 2\mathbf{J}_i)^\top \mathbf{C}_i$$
$$\text{s.t.} \quad \hat{\mathbf{Y}}_i \le \mathbf{C}_i \le \mathbf{1}_n, \mathbf{C}_i\mathbf{1}_l = l - 1. \qquad (13)$$

The problem in Eq. (13) is a standard Quadratic Programming (QP) problem, which can be solved by off-the-shelf QP tools.

### 2.7 Extensions of PLCP

We can also extend PLCP to a kernel version which extends the feature map to a higher dimensional space or a deep-learning based version which enables deep-learning based methods involved in PLCP.

#### 2.7.1 Kernel Extension

The linear classifier in Eq. (5) may lack some ability to tackle the complex relationships among the data. Therefore, we extend it to a kernel version. Denote $\phi(\cdot) : \mathbb{R}^q \to \mathbb{R}^h$ the feature mapping that maps the feature space to some higher dimension space with $h$ dimensions, then we can rewrite Eq. (10) as

$$\min_{\hat{\mathbf{W}}, \hat{\mathbf{b}}} \quad \|\mathbf{Z}\|_F^2 + \lambda\|\hat{\mathbf{W}}\|_F^2$$
$$\text{s.t.} \quad \mathbf{Z} = \mathbf{\Phi}\hat{\mathbf{W}} + \mathbf{1}_n\hat{\mathbf{b}}^\top - \mathbf{C}, \qquad (14)$$

where $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), ..., \phi(\mathbf{x}_n)]$. The Lagrangian function of this problem is

$$\mathcal{L}(\hat{\mathbf{W}}, \hat{\mathbf{b}}, \mathbf{Z}, \mathbf{A}) = \|\mathbf{Z}\|_F^2 + \lambda\|\hat{\mathbf{W}}\|_F^2$$
$$- \operatorname{tr}(\mathbf{A}^\top(\mathbf{\Phi}\hat{\mathbf{W}} + \mathbf{1}_n\hat{\mathbf{b}}^\top - \mathbf{C} - \mathbf{Z})), \qquad (15)$$

where $\mathbf{A} = [a_{ij}]_{n \times l} \in \mathbb{R}^{n \times l}$ stores the Lagrange multipliers. Following the KKT conditions:

$$\frac{\partial\mathcal{L}}{\partial\hat{\mathbf{W}}} = 2\lambda\hat{\mathbf{W}} - \mathbf{\Phi}^\top\mathbf{A} = 0, \frac{\partial\mathcal{L}}{\partial\hat{\mathbf{b}}} = \mathbf{A}^\top\mathbf{1}_n = 0,$$
$$\frac{\partial\mathcal{L}}{\partial\mathbf{Z}} = 2\mathbf{Z} + \mathbf{A} = 0, \frac{\partial\mathcal{L}}{\partial\mathbf{A}} = \mathbf{\Phi}\hat{\mathbf{W}} + \mathbf{1}_n\hat{\mathbf{b}}^\top - \mathbf{C} - \mathbf{Z} = 0, \qquad (16)$$

we have the corresponding solution:

$$\hat{\mathbf{W}} = \frac{\mathbf{\Phi}^\top\mathbf{A}}{2\lambda}, \mathbf{A} = \left( \frac{1}{2\lambda}\mathbf{K} + \frac{1}{2}\mathbf{I}_{n \times n} \right)^{-1} \left( \mathbf{C} - \mathbf{1}_n\hat{\mathbf{b}}^\top \right),$$
$$\mathbf{s} = \mathbf{1}_n^\top \left( \frac{1}{2\lambda}\mathbf{K} + \frac{1}{2}\mathbf{I}_{n \times n} \right)^{-1}, \hat{\mathbf{b}}^\top = \frac{\mathbf{s}\mathbf{C}}{\mathbf{s}\mathbf{1}_n}, \qquad (17)$$

where $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^\top$ is the kernel matrix with its element $k_{ij} = \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j) = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ based on the kernel function $\mathcal{K}(\cdot, \cdot)$. For PLCP, Gaussian function $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2})$ is adopted as the kernel function with $\sigma$ to the average distance of all pairs of training samples. The modeling output is denoted by

$$\hat{\mathbf{M}} = \frac{1}{2\lambda}\mathbf{K}\mathbf{A} + \mathbf{1}_n\hat{\mathbf{b}}^\top. \qquad (18)$$

#### 2.7.2 Deep Learning Extension

PLCP can be also extended to a deep-learning version to further demonstrate its universality and effectiveness. The model of the majority of the deep-learning based methods usually contains a feature encoder followed by a prediction head, which predicts the labeling confidence of each sample. Denote a model in $\mathcal{B}$ with such architecture $g(\cdot)$, specifically, an additional model $\hat{g}(\cdot)$ with the same architecture as $g(\cdot)$ is introduced as the partner classifier, which predicts the non-candidate labeling confidence of each sample. Given an example $\mathbf{x}$ and its candidate label vector $\mathbf{y}$, the non-candidate loss is defined as

$$\mathcal{L}_{com} = -\sum_{i=1}^{l}(1 - y_i)\log(\hat{g}_i(\mathbf{x})). \qquad (19)$$

Afterwards, a collaborative loss is designed to link the two models:

$$\mathcal{L}_{col} = \sum_{i=1}^{l} v_i\hat{v}_i. \qquad (20)$$

Here, $\mathbf{v} = [v_1, v_2, ..., v_l]$ and $\hat{\mathbf{v}} = [\hat{v}_1, \hat{v}_2, ..., \hat{v}_l]$ are two blurred predictions of $g(\mathbf{x})$ and $\hat{g}(\mathbf{x})$ respectively, where

$$\mathbf{v} = \frac{\mathcal{E}(e^k g(\mathbf{x})) \odot \mathbf{y}}{\|\mathcal{E}(e^k g(\mathbf{x})) \odot \mathbf{y}\|}$$
$$\hat{\mathbf{v}} = \mathbf{1}_l^\top - \frac{\mathcal{E}(e^k(1 - \hat{g}(\mathbf{x}))) \odot \mathbf{y}}{\|\mathcal{E}(e^k(1 - \hat{g}(\mathbf{x}))) \odot \mathbf{y}\|}. \qquad (21)$$

The overall loss function is:

$$\mathcal{L} = \mathcal{L}_{ori} + \mathcal{L}_{com} + \mu\mathcal{L}_{col}, \qquad (22)$$

where $\mathcal{L}_{ori}$ is the original loss function of $\mathcal{B}$ and $\mu$ is a hyper-parameter.

TABLE 1
Characteristics of the real-world data sets.

| Data set | #examples | #features | #labels | average #labels |
|---|---|---|---|---|
| FG-NET [14] | 1002 | 262 | 78 | 7.48 |
| Lost [18] | 1122 | 108 | 16 | 2.23 |
| MSRCv2 [25] | 1758 | 48 | 23 | 3.16 |
| Mirflickr [26] | 2780 | 1536 | 14 | 2.76 |
| Soccer Player [11] | 17472 | 279 | 171 | 2.09 |
| Yahoo!News [12] | 22991 | 163 | 219 | 1.91 |

## 3 EXPERIMENTS

### 3.1 Compared Approaches

To evaluate the effectiveness of PLCP, we couple it with several well-established partial label learning approaches. Suppose $\mathcal{B}$ represents any partial label learning classifier (i.e., the base classifier) and $\mathcal{B}$-PLCP is the $\mathcal{B}$ coupled with PLCP, the performances of $\mathcal{B}$ and $\mathcal{B}$-PLCP are compared to verify the effectiveness of PLCP. In this paper, $\mathcal{B}$ is instantiated by six stand-alone (non-deep) approaches, PL-CL [17], PL-AGGD [16], SURE [20], LALO [15], PL-SVM [22] and PL-KNN [21], and two deep-learning based methods PICO [23] and PRODEN [24]. The hyper-parameters of $\mathcal{B}$ are all set according to the original papers.

### 3.2 Comparison with Stand-alone Methods

#### 3.2.1 Experimental Settings

For PLCP, we set $\lambda = 0.05$, $\alpha = 0.5$, $\gamma = 2$ and $k = -1$, and the maximum iteration of mutual supervision is set to 5. For PL-CL, PL-AGGD, SURE, LALO and PL-SVM, the kernel function is Gaussian function, which is the same as we adopt. Ten runs of 50%/50% random train/test splits are performed, and the average accuracy with standard deviation is represented for all $\mathcal{B}$ and $\mathcal{B}$-PLCP. For PL-SVM and PL-KNN which do not adopt labeling confidence strategy, $\hat{\mathbf{O}}$ is further processed as $\mathcal{G}(\hat{\mathbf{O}} - \mathbf{P})$ where $\mathcal{G}$ is an element-wise operator and $\mathcal{G}(x) = 1$ if $x \geq 0$ otherwise 0 with $x$ being a scalar.

We conduct experiments on six real-world partial label data sets collected from several domains and tasks, including FG-NET [14] for facial age estimation, Lost [18], Soccer Player [11] and Yahoo!News [12] for automatic face naming, MSRCv2 [25] for object classification and Mirflickr [26] for web image classification. For object classification task, image segmentations are taken as instances with objects appearing in the same image as candidate labels. For web image classification task, each example consists of images and a set of candidate tags extracted from the web page. The details of the data sets are summarized in Table 1.

#### 3.2.2 Performance on Real-World Data Set

It is noteworthy that the number of average label of FG-NET is quite large, which could cause quite low classification accuracy for all approaches. The common strategy is to evaluate the mean absolute error (MAE) between the predicted age and the ground-truth one. Specifically, we add another two sets of comparisons on FG-NET w.r.t. MAE3/MAE5, which means that test samples can be considered to be correctly classified if the difference between the predicted age and true age is no more than 3/5 years. Table 2 summaries

the classification accuracy with standard deviation of each approach on real-world data sets, where we can observe that

- $\mathcal{B}$-PLCP significantly outperforms the base classifier $\mathcal{B}$ in all cases according to the pairwise $t$-test with a significance level of 0.05, which validates the effectiveness of PLCP.
- State-of-the-art (SOTA) approaches, such as PL-CL, PL-AGGD, LALO and SURE, can also be significantly improved by PLCP on all data sets. For instance, on FG-NET the performance of SURE can be improved by **45%**, and on FG-NET(MAE3) the performance of LALO can be improved by **5%**. Additionally, on Lost PL-AGGD can be improved by **4%** and PL-CL can be improved by **5.5%**. The partner classifier's non-candidate label information effectively and significantly aids disambiguation, leading to outstanding performance of the PLCP framework.
- The performances of PL-SVM and PL-KNN are improved significantly and impressively when coupled with PLCP. For example, on FG-NET the performance of PL-KNN-PLCP is **more than two times** better than that of PL-KNN, and PL-SVM-PLCP's performance is **more than 1.5 times** better than that of PL-SVM on Lost. Although PL-SVM and PL-KNN are inferior to the SOTA methods, **they can achieve almost the same performance as SOTA's when coupled with PLCP**.

### 3.3 Comparison with Deep-learning Based Methods

#### 3.3.1 Experimental Settings

In PLCP, we set $k = -1$ and $\mu = 0.5$. All the hyper-parameters and experimental setups of $\mathcal{B}$ are set based on their original papers.

We conduct experiments on two benchmarks CIFAR-10 and CIFAR-100 [27], and following the settings in [23], [24], we generate false positive labels by flipping negative labels $\hat{y} \neq y$ for each sample with a probability $q = P(\hat{y} \in \mathcal{Y}|\hat{y} \neq y)$, where $y$ is the ground-truth label. In other words, the probability of a false positive is uniform across all $l - 1$ negative labels. We combine the flipped ones with the ground-truth label to create the set of candidate labels. Specifically, $q$ is set to 0.1, 0.3 and 0.5 for CIFAR-10 and 0.01, 0.05 and 0.1 for CIFAR-100. Five independent runs are performed with the the average accuracy and standard deviation recorded (with different seeds).

#### 3.3.2 Performance on CIFAR-10 and CIFAR-100

Table 3 presents the classification accuracy with standard deviation of each approach on CIFAR-10 and CIFAR-100, where we can find that

- $\mathcal{B}$-PLCP consistently outperforms the base classifier $\mathcal{B}$ in 83.3% cases, which confirms that PLCP effectively improves the performance of deep-learning based models. Notably, $\mathcal{B}$-PLCP is never significantly outperformed by any $\mathcal{B}$.
- Although the performance improvement of $\mathcal{B}$-PLCP on different settings appears to be limited, it is important to note that its performance is very close to that of fully supervised $\mathcal{B}$. In some cases, it even surpasses the performance of fully supervised one.

TABLE 2
Classification accuracy of each compared approach on the real-world data sets. For any compared approach $\mathcal{B}$, ●/○ indicates whether $\mathcal{B}$-PLCP is statistically superior/inferior to $\mathcal{B}$ according to pairwise $t$-test at significance level of 0.05.

| Approaches | Data set | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | FG-NET | Lost | MSRCv2 | Mirflickr | Soccer Player | Yahoo!News | FG-NET(MAE3) | FG-NET(MAE5) |
| PL-CL | 0.072 ± 0.009 | 0.710 ± 0.022 | 0.469 ± 0.016 | 0.647 ± 0.012 | 0.534 ± 0.004 | 0.618 ± 0.003 | 0.433 ± 0.022 | 0.575 ± 0.015 |
| PL-CL-PLCP | 0.081 ± 0.006 ● | 0.762 ± 0.017 ● | 0.496 ± 0.010 ● | 0.667 ± 0.010 ● | 0.545 ± 0.001 ● | 0.628 ± 0.002 ● | 0.450 ± 0.013 ● | 0.595 ± 0.006 ● |
| PL-AGGD | 0.063 ± 0.010 | 0.690 ± 0.020 | 0.451 ± 0.023 | 0.610 ± 0.012 | 0.521 ± 0.004 | 0.605 ± 0.002 | 0.418 ± 0.020 | 0.562 ± 0.020 |
| PL-AGGD-PLCP | 0.076 ± 0.009 ● | 0.717 ± 0.020 ● | 0.474 ± 0.015 ● | 0.668 ± 0.014 ● | 0.534 ± 0.003 ● | 0.611 ± 0.002 ● | 0.442 ± 0.018 ● | 0.582 ± 0.011 ● |
| SURE | 0.052 ± 0.007 | 0.709 ± 0.022 | 0.445 ± 0.022 | 0.630 ± 0.022 | 0.519 ± 0.004 | 0.598 ± 0.002 | 0.356 ± 0.020 | 0.494 ± 0.021 |
| SURE-PLCP | 0.077 ± 0.009 ● | 0.719 ± 0.021 ● | 0.460 ± 0.018 ● | 0.657 ± 0.020 ● | 0.526 ± 0.003 ● | 0.608 ± 0.002 ● | 0.443 ± 0.015 ● | 0.584 ± 0.012 ● |
| LALO | 0.065 ± 0.010 | 0.682 ± 0.019 | 0.449 ± 0.016 | 0.629 ± 0.016 | 0.523 ± 0.003 | 0.601 ± 0.003 | 0.422 ± 0.023 | 0.566 ± 0.015 |
| LALO-PLCP | 0.076 ± 0.010 ● | 0.703 ± 0.017 ● | 0.456 ± 0.018 ● | 0.647 ± 0.018 ● | 0.529 ± 0.004 ● | 0.605 ± 0.002 ● | 0.444 ± 0.015 ● | 0.585 ± 0.015 ● |
| PL-SVM | 0.043 ± 0.008 | 0.406 ± 0.033 | 0.389 ± 0.029 | 0.516 ± 0.022 | 0.412 ± 0.006 | 0.509 ± 0.006 | 0.314 ± 0.019 | 0.445 ± 0.016 |
| PL-SVM-PLCP | 0.081 ± 0.010 ● | 0.692 ± 0.024 ● | 0.468 ± 0.021 ● | 0.611 ± 0.018 ● | 0.526 ± 0.006 ● | 0.609 ± 0.002 ● | 0.439 ± 0.017 ● | 0.584 ± 0.015 ● |
| PL-KNN | 0.036 ± 0.006 | 0.300 ± 0.018 | 0.393 ± 0.014 | 0.454 ± 0.016 | 0.492 ± 0.003 | 0.368 ± 0.004 | 0.288 ± 0.014 | 0.440 ± 0.017 |
| PL-KNN-PLCP | 0.077 ± 0.006 ● | 0.668 ± 0.019 ● | 0.469 ± 0.016 ● | 0.608 ± 0.026 ● | 0.523 ± 0.003 ● | 0.597 ± 0.002 ● | 0.434 ± 0.020 ● | 0.578 ± 0.014 ● |
| Average Improvement Ratio: | PL-CL: 3.61% | PL-AGGD: 5.10 % | SURE: 12.24 % | LALO: 4.01 % | PL-SVM: 39.26 % | PL-KNN: 53.98 % | | |

TABLE 3
Classification accuracy of each compared approach on CIFAR-10 and CIFAR-100. For any compared approach $\mathcal{B}$, ●/○ indicates whether $\mathcal{B}$-PLCP is statistically superior/inferior to $\mathcal{B}$ according to pairwise $t$-test at significance level of 0.05.

| Approaches | CIFAR-10 | | | CIFAR-100 | | |
| --- | --- | --- | --- | --- | --- | --- |
| | $q = 0.1$ | $q = 0.3$ | $q = 0.5$ | $q = 0.01$ | $q = 0.05$ | $q = 0.1$ |
| PICO | 94.39 ± 0.18 % | 94.18 ± 0.12 % | 93.58 ± 0.06 % | 73.09 ± 0.34 % | 72.74 ± 0.30 % | 69.91 ± 0.24 % |
| PICO-PLCP | 94.80 ± 0.07 % ● | 94.53 ± 0.10 % ● | 93.67 ± 0.16 % ● | 73.90 ± 0.20 % ● | 73.51 ± 0.21 % ● | 70.00 ± 0.35 % |
| Fully Supervised | $\mathcal{B}$: 94.91 ± 0.07 % | $\mathcal{B}$-PLCP: | 95.02 ± 0.03 % | $\mathcal{B}$: 73.56 ± 0.10 % | $\mathcal{B}$-PLCP: | 73.69 ± 0.14 % |
| PRODEN | 89.12 ± 0.12 % | 87.56 ± 0.15 % | 84.92 ± 0.31 % | 63.36 ± 0.33 % | 60.88 ± 0.35 % | 50.98 ± 0.74 % |
| PRODEN-PLCP | 89.63 ± 0.15 % ● | 88.19 ± 0.19 % ● | 85.31 ± 0.31 % ● | 64.20 ± 0.25 % ● | 61.78 ± 0.29 ● | 50.76 ± 0.90 % |
| Fully Supervised | $\mathcal{B}$: 90.03 ± 0.13 % | $\mathcal{B}$-PLCP: | 90.30 ± 0.08 % | $\mathcal{B}$: 65.03 ± 0.35 % | $\mathcal{B}$-PLCP: | 65.52 ± 0.32 % |

TABLE 4
Ablation study of PLCP coupled with PL-AGGD on the real-world data sets. ●/○ indicates whether PL-AGGD-PLCP is statistically superior/inferior to its degenerated version according to pairwise $t$-test at significance level of 0.05.

| Kernel | Partner | Blur | Data set | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | | | FG-NET | Lost | MSRCv2 | Mirflickr | Soccer Player | Yahoo!News | FG-NET(MAE3) | FG-NET(MAE5) |
| | PL-AGGD | | 0.063 ± 0.010 | 0.690 ± 0.020 | 0.451 ± 0.023 | 0.610 ± 0.012 | 0.521 ± 0.004 | 0.605 ± 0.002 | 0.418 ± 0.020 | 0.562 ± 0.020 |
| × | P | × | 0.073 ± 0.011 ● | 0.698 ± 0.023 ● | 0.380 ± 0.013 ● | 0.542 ± 0.013 ● | 0.492 ± 0.003 ● | 0.463 ± 0.002 ● | 0.421 ± 0.020 ● | 0.560 ± 0.016 ● |
| ✓ | P | × | 0.073 ± 0.006 ● | 0.721 ± 0.024 ○ | 0.471 ± 0.016 ● | 0.664 ± 0.012 ● | 0.521 ± 0.004 ● | 0.608 ± 0.003 ● | 0.422 ± 0.030 ● | 0.566 ± 0.020 ● |
| ✓ | O | ✓ | 0.071 ± 0.001 ● | 0.721 ± 0.004 ○ | 0.470 ± 0,020 ● | 0.663 ± 0.011 ● | 0.522 ± 0.003 ● | 0.605 ± 0.002 ● | 0.417 ± 0.022 ● | 0.576 ± 0.014 ● |
| ✓ | P | ✓ | 0.076 ± 0.009 | 0.717 ± 0.020 | 0.474 ± 0.015 | 0.668 ± 0.014 | 0.534 ± 0.003 | 0.611 ± 0.002 | 0.442 ± 0.018 | 0.582 ± 0.011 |

TABLE 5
Ablation study of PLCP coupled with PICO on CIFAR-10 and CIFAR-100. ●/○ indicates whether PICO-PLCP is statistically superior/inferior to its degenerated version according to pairwise $t$-test at significance level of 0.05.

| Partner | Blur | CIFAR-10 | | | CIFAR-100 | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| | | $q = 0.1$ | $q = 0.3$ | $q = 0.5$ | $q = 0.01$ | $q = 0.05$ | $q = 0.1$ |
| | PICO | 94.39 ± 0.18 % | 94.18 ± 0.12 % | 93.58 ± 0.06 % | 73.09 ± 0.34 % | 72.74 ± 0.30 % | 69.91 ± 0.24 % |
| P | × | 94.60 ± 0.09 % ● | 94.20 ± 0.11 % ● | 93.63 ± 0.09 % ● | 73.71 ± 0.28 % ● | 73.17 ± 0.22 % ● | 69.91 ± 0.18 % ● |
| O | ✓ | 94.44 ± 0.10 % ● | 94.37 ± 0.11 % ● | 93.59 ± 0.13 % ● | 73.43 ± 0.24 % ● | 72.98 ± 0.24 % ● | 69.77 ± 0.33 % ● |
| | PICO-PLCP | 94.80 ± 0.07 % | 94.53 ± 0.10 % | 93.67 ± 0.16 % | 73.90 ± 0.20 % | 73.51 ± 0.21 % | 70.00 ± 0.35 % |

## 3.4 Ablation Study

### 3.4.1 Effectiveness of the Non-candidate Label Information

In order to validate the effectiveness of the non-candidate label information, we instantiate the partner classifier with the form as

$$\min_{\mathbf{W},\mathbf{b},\mathbf{L}} \quad \left\| \mathbf{XW} + \mathbf{1}_n \mathbf{b}^\mathsf{T} - \mathbf{L} \right\|_F^2 + \lambda \left\| \hat{\mathbf{W}} \right\|_F^2$$
$$+ \gamma \operatorname{tr} \left( \mathbf{O}_1 (\mathbf{1}_{n \times l} - \mathbf{L})^\mathsf{T} \right) \tag{23}$$
$$\text{s.t.} \quad \mathbf{0}_{n \times l} \le \mathbf{L} \le \mathbf{Y}, \mathbf{L}\mathbf{1}_l = \mathbf{1}_n,$$

where $\mathbf{W}$ and $\mathbf{b}$ are two model parameters and $\mathbf{L} \in \mathbb{R}^{n \times l}$ is the labeling confidence matrix. Denote this classifier $O$ and the original partner classifier $P$, the classification accuracy of PL-AGGD mutual-supervised with $O$ on each real-world data set is shown in Table 4, and that of PICO on CIFAR data set is shown in Table 5. It is obvious that the performances of PL-AGGD and PICO coupled with $O$ are significantly inferior to that with $P$ in 87.5% and all cases respectively, which validates the usefulness of the non-candidate label information. The accurate non-candidate label information provided by the partner classifier effectively complements the base classifier, leading to better performance.

### 3.4.2 Usefulness of the Blurring Mechanism

It is also interesting to investigate the usefulness of the blurring mechanism in PLCP. By simply normalizing $\mathbf{P}_1$ to $\mathbf{O}_1$ and $\hat{\mathbf{P}}_1$ to $\hat{\mathbf{O}}_1$ (i.e., skipping Eq. (3) and Eq. (8)), the
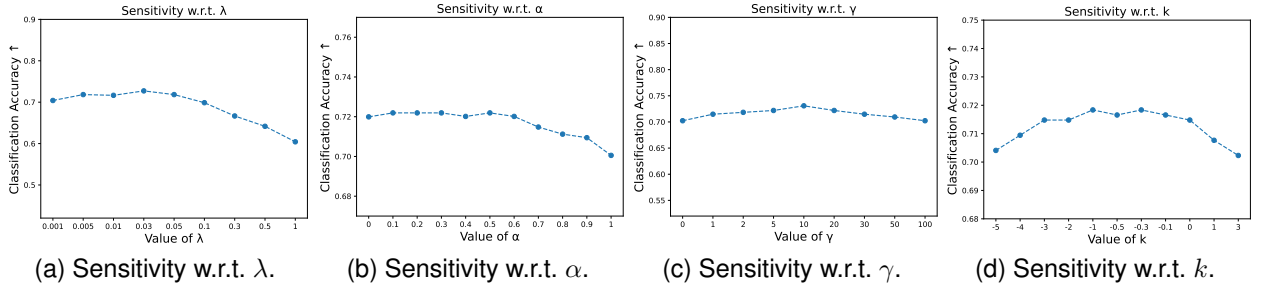
(a) Sensitivity w.r.t. $\lambda$.     (b) Sensitivity w.r.t. $\alpha$.     (c) Sensitivity w.r.t. $\gamma$.     (d) Sensitivity w.r.t. $k$.

Fig. 4. Sensitivity of PLCP.

TABLE 6
Correction ratio of each method on real-world data sets.

| Approaches | Data set | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | FG-NET | Lost | MSRCv2 | Mirflickr | Soccer Player | Yahoo!News | FG-NET(MAE3) | FG-NET(MAE5) |
| PL-CL | $3.65 \pm 0.02$ % | $21.29 \pm 1.04$ % | $6.24 \pm 0.48$ % | $7.96 \pm 1.19$ % | $0.77 \pm 0.08$ % | $4.19 \pm 0.42$ % | $1.82 \pm 0.19$ % | $2.46 \pm 0.28$ % |
| PL-AGGD | $3.64 \pm 0.04$ % | $20.00 \pm 1.55$ % | $5.17 \pm 0.49$ % | $9.31 \pm 1.35$ % | $1.39 \pm 0.21$ % | $1.99 \pm 0.13$ % | $3.92 \pm 0.84$ % | $4.81 \pm 1.02$ % |
| SURE | $6.11 \pm 0.02$ % | $6.67 \pm 0.44$ % | $3.93 \pm 0.73$ % | $11.39 \pm 2.11$ % | $1.83 \pm 0.26$ % | $7.45 \pm 0.39$ % | $26.30 \pm 2.48$ % | $34.90 \pm 1.26$ % |
| LALO | $3.46 \pm 0.01$ % | $13.77 \pm 1.72$ % | $1.94 \pm 0.35$ % | $7.43 \pm 0.91$ % | $1.41 \pm 0.22$ % | $3.48 \pm 0.19$ % | $4.80 \pm 1.26$ % | $5.85 \pm 0.23$ % |
| PL-SVM | $7.49 \pm 0.21$ % | $64.89 \pm 2.51$ % | $31.28 \pm 2.19$ % | $38.86 \pm 2.47$ % | $28.10 \pm 0.73$ % | $34.80 \pm 0.88$ % | $38.98 \pm 1.92$ % | $47.65 \pm 2.59$ % |
| PL-KNN | $8.42 \pm 0.17$ % | $56.93 \pm 1.93$ % | $34.46 \pm 1.83$ % | $49.81 \pm 2.19$ % | $6.81 \pm 0.52$ % | $43.47 \pm 0.81$ % | $41.74 \pm 2.01$ % | $39.22 \pm 1.82$ % |

TABLE 7
Transductive accuracy of each compared approach on the real-world data sets. For any compared approach $\mathcal{B}$, ●/○ indicates whether $\mathcal{B}$-PLCP is statistically superior/inferior to $\mathcal{B}$ according to pairwise $t$-test at significance level of 0.05.

| Approaches | Data set | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | FG-NET | Lost | MSRCv2 | Mirflickr | Soccer Player | Yahoo!News | FG-NET(MAE3) | FG-NET(MAE5) |
| PL-CL | $0.159 \pm 0.016$ | $0.832 \pm 0.019$ | $0.585 \pm 0.012$ | $0.697 \pm 0.019$ | $0.715 \pm 0.001$ | $0.827 \pm 0.003$ | $0.600 \pm 0.029$ | $0.737 \pm 0.018$ |
| PL-CL-PLCP | $0.181 \pm 0.012$ ● | $0.854 \pm 0.010$ ● | $0.638 \pm 0.008$ ● | $0.704 \pm 0.021$ ● | $0.722 \pm 0.004$ ● | $0.831 \pm 0.003$ ● | $0.618 \pm 0.023$ ● | $0.747 \pm 0.018$ ● |
| PL-AGGD | $0.141 \pm 0.012$ | $0.793 \pm 0.020$ | $0.557 \pm 0.015$ | $0.695 \pm 0.015$ | $0.669 \pm 0.003$ | $0.808 \pm 0.005$ | $0.571 \pm 0.027$ | $0.709 \pm 0.025$ |
| PL-AGGD-PLCP | $0.167 \pm 0.015$ ● | $0.829 \pm 0.016$ ● | $0.638 \pm 0.011$ ● | $0.715 \pm 0.015$ ● | $0.714 \pm 0.004$ ● | $0.831 \pm 0.004$ ● | $0.600 \pm 0.022$ ● | $0.738 \pm 0.016$ ● |
| SURE | $0.158 \pm 0.012$ | $0.796 \pm 0.026$ | $0.603 \pm 0.016$ | $0.650 \pm 0.024$ | $0.700 \pm 0.003$ | $0.798 \pm 0.005$ | $0.590 \pm 0.019$ | $0.727 \pm 0.020$ |
| SURE-PLCP | $0.170 \pm 0.011$ ● | $0.836 \pm 0.021$ ● | $0.622 \pm 0.015$ ● | $0.697 \pm 0.017$ ● | $0.706 \pm 0.005$ ● | $0.827 \pm 0.006$ ● | $0.605 \pm 0.026$ ● | $0.739 \pm 0.018$ ● |
| LALO | $0.153 \pm 0.017$ | $0.818 \pm 0.019$ | $0.548 \pm 0.009$ | $0.681 \pm 0.013$ | $0.688 \pm 0.004$ | $0.822 \pm 0.004$ | $0.593 \pm 0.025$ | $0.730 \pm 0.015$ |
| LALO-PLCP | $0.168 \pm 0.018$ ● | $0.831 \pm 0.019$ ● | $0.620 \pm 0.009$ ● | $0.694 \pm 0.019$ ● | $0.706 \pm 0.004$ ● | $0.827 \pm 0.004$ ● | $0.604 \pm 0.025$ ● | $0.741 \pm 0.019$ ● |
| PL-SVM | $0.176 \pm 0.015$ | $0.609 \pm 0.055$ | $0.570 \pm 0.040$ | $0.581 \pm 0.022$ | $0.660 \pm 0.008$ | $0.691 \pm 0.005$ | $0.566 \pm 0.025$ | $0.706 \pm 0.024$ |
| PL-SVM-PLCP | $0.188 \pm 0.008$ ● | $0.792 \pm 0.036$ ● | $0.639 \pm 0.027$ ● | $0.630 \pm 0.028$ ● | $0.705 \pm 0.003$ ● | $0.823 \pm 0.004$ ● | $0.580 \pm 0.027$ ● | $0.721 \pm 0.019$ ● |
| PL-KNN | $0.041 \pm 0.007$ | $0.337 \pm 0.030$ | $0.415 \pm 0.014$ | $0.466 \pm 0.013$ | $0.493 \pm 0.004$ | $0.403 \pm 0.010$ | $0.285 \pm 0.017$ | $0.438 \pm 0.015$ |
| PL-KNN-PLCP | $0.168 \pm 0.012$ ● | $0.781 \pm 0.033$ ● | $0.639 \pm 0.021$ ● | $0.628 \pm 0.015$ ● | $0.699 \pm 0.006$ ● | $0.793 \pm 0.004$ ● | $0.579 \pm 0.025$ ● | $0.722 \pm 0.018$ ● |

classification accuracy of PLCP without this mechanism is recorded in Table 4-5, where we can find that the performance of PL-AGGD-PLCP and PICO-PLCP w/o Blur is significantly inferior to those with Blur in 87.5% and all cases respectively. The blurring mechanism effectively tackles the overconfidence issue in PLL, leading to outstanding performance.

Furthermore, we examine the impact of various values of $k$ on the performance of PLCP, as depicted in Fig. 4(d). It can be clearly observed that when $k$ is too small, the performance of PLCP deteriorates significantly. Additionally, the performance will be also exacerbated when $k \geq 0$ (strictly $k > \ln 2$), for in this case the predictions of the classifiers may be enhanced rather than being blurred, amplifying small differences between two values. As discussed in Sec. 4.3, PLCP contributes little to appeal and disambiguation in this situation, resulting in inferior performance.

### 3.4.3 Influence of Kernel Extension

We also conduct experiments to show the improvement of kernel extension used in partner classifier. Comparing the results in the first and second rows in Table 4, we can observe that the performance of PL-AGGD coupled with the

classifier using kernel extension is superior to that without kernel extension on all the data sets, which validates the effectiveness of the kernel.

### 3.4.4 Sensitive Analysis

Figs. 4(a)-(d) illustrate the sensitivity of PL-AGGD coupled with PLCP w.r.t $\lambda$, $\alpha$, $\gamma$ and $k$ on Lost, where we can find that PL-AGGD-PLCP performs quite stably with the hyperparameters changing within a reasonably wide range, which is also a desirable property in practice.

## 4 FURTHER ANALYSIS

### 4.1 The Appeal Ability

In order to evaluate the appeal ability of PLCP, we propose a metric called "correction ratio". Specifically, we record the indices of mislabeled samples on each data set, and ascertain how many of them can be successfully corrected by PLCP. The results, shown in Table 6, demonstrate that PLCP adeptly addresses mislabeled samples, successfully identifying and correcting them with a high degree of probability. While it is conceivable that PLCP could result in mislabeling of

TABLE 8
Comparison of aggressive term and collaborative term of PLCP coupled with PL-AGGD on real-world data sets. PL-AGGD-PLCP(A) represents the one using the aggressive term. ●/○ indicates whether PL-AGGD-PLCP is statistically superior/inferior to PL-AGGD-PLCP(A) according to pairwise $t$-test at significance level of 0.05.

| Approaches | Data set | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | FG-NET | Lost | MSRCv2 | Mirflickr | Soccer Player | Yahoo!News | FG-NET(MAE3) | FG-NET(MAE5) |
| PL-AGGD | $0.063 \pm 0.010$ | $0.690 \pm 0.020$ | $0.451 \pm 0.023$ | $0.610 \pm 0.012$ | $0.521 \pm 0.004$ | $0.605 \pm 0.002$ | $0.418 \pm 0.020$ | $0.562 \pm 0.020$ |
| PL-AGGD-PLCP(A) | $0.070 \pm 0.010$ | $0.697 \pm 0.019$ | $0.468 \pm 0.022$ | $0.657 \pm 0.009$ | $0.522 \pm 0.004$ | $0.604 \pm 0.002$ | $0.426 \pm 0.019$ | $0.564 \pm 0.016$ |
| PL-AGGD-PLCP | $0.076 \pm 0.009$ ● | $0.717 \pm 0.020$ ● | $0.474 \pm 0.015$ ● | $0.668 \pm 0.014$ ● | $0.534 \pm 0.003$ ● | $0.611 \pm 0.002$ ● | $0.442 \pm 0.018$ ● | $0.582 \pm 0.011$ ● |

originally correct samples, we have observed that such phenomenon occurs with a relatively low probability of 2.56%.

## 4.2 Improvement of the Disambiguation Ability

Transductive accuracy (i.e., classification accuracy on training samples) reflects the disambiguation ability of a PLL approach [2], [16], [28]. In order to validate whether PLCP can correct some mislabeled samples and truly improve the disambiguation ability of $\mathcal{B}$, we summarize the transductive accuracy of $\mathcal{B}$ and $\mathcal{B}$-PLCP on real-world data sets and the results are shown in Table 7. It is obvious that $\mathcal{B}$-PLCP outperforms $\mathcal{B}$ in all cases according to the pairwise $t$-test with a significance level of 0.05, which validates that the disambiguation ability of $\mathcal{B}$ can be truly improved by PLCP. In other words, by integrating PLCP, classifiers can better identify and correct mislabeled samples, leading to outstanding performance.

## 4.3 Analysis of the Collaborative Term

### 4.3.1 How dose the collaborative term work?

We focus our attention on the following convex optimization problem:

$$\min_{\mathbf{C}} \quad \mathrm{tr}(\mathbf{O}\mathbf{C}^{\mathsf{T}})$$
$$\text{s.t.} \quad \hat{\mathbf{Y}} \leq \mathbf{C} \leq \mathbf{1}_{n \times l}, \mathbf{C}\mathbf{1}_l = (l-1)\mathbf{1}_n, \tag{24}$$

where $\mathbf{O}$ is the blurred labeling confidence matrix. Since each row of $\mathbf{C}$ and $\mathbf{O}$ is independent to each other, the problem in Eq. (24) can be further transformed into a series of subproblems, which can be rewritten as

$$\min_{\mathbf{C}_i} \quad L = \sum_j O_{ij} C_{ij}$$
$$\text{s.t.} \quad \hat{y}_{ij} \leq C_{ij} \leq 1, \sum_j C_{ij} = l - 1, \tag{25}$$
$$j = 1, 2, ..., l.$$

For simplicity, suppose $\mathbf{O}_i$ is arranged in ascending order, i.e., $O_{ij} \leq O_{ik}$ if $j \leq k$. Given that the partial derivative of $L$ w.r.t. each $C_{ij}$ is a constant $O_{ij}$, the variation in $L$ remains monotonic when $C_{ij}$ is incremented or decremented. This indicates that $L$ reaches its minimum at the boundaries of its range. Consequently, the optimal solution of $\mathbf{C}_i$ is that only to have precisely one element set to 0 and the remainder set to 1. Notably $C_{il} = 0$ due to $O_{il}$ is the maximal element in $\mathbf{O}_i$. Therefore, the optimal solution for the optimization problem in Eq. (25) adheres to a zero-hot strategy, with the zero precisely being aligned with the peak value in $\mathbf{O}_i$.

### 4.3.2 What properties does the collaborative term possess?

The collaborative term is ingeniously crafted to establish a connection between the base and partner classifiers, manifesting as a trace term in its practical application. Its primary objective is to require $\mathbf{O}_i^{\mathsf{T}}\mathbf{C}_i = 0$ for each sample $\mathbf{x}_i$. Generally, the relative value of $O_{ij}$ and $C_{ij}$ should be opposite, i.e., if $O_{ij}$ is large, $C_{ij}$ should be small. However, the collaborative term permits $C_{ij}$ to be small or even zero if $O_{ij}$ is small, which means that it works in a wild way. This gentle design is intentional: our goal is for the partner classifier to be aware of the base classifier's predictions without being overly swayed by them. Besides, this design also augments the classifier's resilience against faults. To circumvent $\mathbf{C}_i$ to be triviality $\mathbf{0}_l$, we introduce the constraint of $\sum_j C_{ij} = l - 1$.

Furthermore, upon deriving the optimal solution for the problem in Eq. (24) determined, we can further elucidate the collaborative term in PLCP. Specifically, as $\gamma$ in Eq. (5) approaches positive infinity, each row of $\mathbf{C}$ is a strictly zero-hot vector, aligning the predicted ground-truth labels for each training sample with those of the base classifier, for the zero positions of $\mathbf{C}$ are the same as the maximum value positions in $\mathbf{O}$. However, given that $\gamma$ is only a small positive value distinctly short of infinity, the elements of $\mathbf{C}$ are values between 0 and 1 rather than being binary.

Therefore, we can conclude that the collaborative term operates as a wild agent, facilitating a soft and moderated interaction between the classifiers.

### 4.3.3 Should the collaborative term be aggressive?

As analysed above, the collaborative term links the two classifiers in a wild manner. Nonetheless, this raises a pertinent question: would a more aggressive collaborative term enhance performance?

To explore the impact of an aggressive collaborative term on performance, we consider the methodology outlined in [17], which links two classifiers through an aggressive term, denoted as $\|\mathbf{O} + \mathbf{C} - \mathbf{1}_{n \times l}\|_F^2$. This aggressive term significantly diverges from our collaborative term. It compels $\mathbf{O}_i + \mathbf{C}_i = \mathbf{1}_l$ for a sample $\mathbf{x}_i$, indicating that if $\mathbf{O}_i$ is small, $\mathbf{C}_i$ should be large. We compare the performances with these two kinds of linking terms, and the results are shown in Table 8. It becomes apparent that employing the aggressive term within PLCP markedly detracts from performance compared to our collaborative term. Consequently, this analysis suggests that a collaborative term designed to facilitate "wild" rather than "aggressive" interactions yields superior results.

TABLE 9
Two representative errors in SURE and LALO. "→" means continuously increasing or decreasing.

| Confidence | False Positive Label (FP) | Ground-truth Label (GT) | FP coupled with PLCP | GT coupled with PLCP |
|---|---|---|---|---|
| SURE | $0.5000 \rightarrow 0.5334$ | $0.5000 \rightarrow 0.4666$ | $0.5334 \rightarrow 0.4177$ | $0.4666 \rightarrow 0.5823$ |
| SURE | $0.6127 \rightarrow 0.5327$ | $0.3873 \rightarrow 0.4673$ | $0.5327 \rightarrow 0.4881$ | $0.4673 \rightarrow 0.5119$ |
| LALO | $0.2500 \rightarrow 0.3241$ | $0.2500 \rightarrow 0.2289$ | $0.3241 \rightarrow 0.2472$ | $0.2289 \rightarrow 0.3227$ |
| LALO | $0.5625 \rightarrow 0.5081$ | $0.3181 \rightarrow 0.4362$ | $0.5081 \rightarrow 0.4474$ | $0.4362 \rightarrow 0.4781$ |

### 4.4 Analysis of the Blurring Mechanism

#### 4.4.1 Why can blurring mechanism blur?

Suppose $a$ and $b$ are labeling confidences of two candidate labels of a sample. The difference between $a$ and $b$ after the blurring mechanism is

$$\frac{e^{e^k a} - e^{e^k b}}{e^{e^k a} + e^{e^k b}} = 1 - \frac{2}{e^{e^k (a-b)} + 1}, \tag{26}$$

We want to prove the following inequality always holds when $k < 0$:

$$a - b > 1 - \frac{2}{e^{e^k (a-b)} + 1} \tag{27}$$

Define the function $\mathcal{F}(x)$ as $\mathcal{F}(x) = x + \frac{2}{e^{e^k x}+1} - 1$ where $x = a - b > 0$, if $k < 0$, we have

$$\frac{d\mathcal{F}}{dx} = 1 - \frac{e^k}{2\cosh^2(\frac{e^k x}{2})} > 1 - \frac{1}{2} > 0, \tag{28}$$

which means that $\mathcal{F}(x)$ is monotonically increasing when $x > 0$. Therefore, $\mathcal{F}(x) > \mathcal{F}(0) = 0$, i.e., we have shown that the inequality in (27) is true when $k < 0$. Q.E.D.

#### 4.4.2 Why do we need the blurring mechanism?

As analysed in Sec. 4.3, significant discrepancies between elements $a$ and $b$ in $\mathbf{O}$ lead to significant differences between the corresponding elements $\hat{a}$ and $\hat{b}$ in $\mathbf{C}$, often inverting their relative magnitudes (i.e., if $a > b$, then typically $\hat{a} < \hat{b}$). Ideally, the partner classifier should learn non-candidate label information while considering the output of the base classifier, allowing it to supplement and improve its performance. However, if we directly pass the predictions of the base classifier to the collaborative term without blurring, the labeling confidences of different labels predicted by the base classifier may differ greatly, causing the partner classifier to be greatly influenced by the base classifier. In extreme cases, this could result in the partner classifier mirroring the base classifier's label predictions exactly.

To mitigate this, we need to design a blurring mechanism that can blur the outputs, so that elements with significant differences become closer after blurring. This blurring mechanism adjusts the absolute values of elements while preserving their relative order, ensuring the base classifier's influence exists but influence the partner classifier in a gentle way. Therefore, the partner classifier can utilize the non-candidate information to better assist in disambiguation and appeal, rather than dominantly guided by the base classifier's outputs.

### 4.5 More Details of the Representative Errors

Generally, the labeling confidence is initialized following Eq. (1), employing an averaging strategy. Nevertheless, as illustrated in Fig. 2, the initial confidence levels of the two labels are unequal. This difference arises because PL-AGGD bases the initialization of labeling confidences on the local geometric structure among instances, resulting in a non-uniform strategy to set initial confidence levels.

Furthermore, the two representative errors highlighted in the introduction are not exclusive to PL-AGGD. Although PL-AGGD serves as a case study for these errors, they are prevalent across other PLL methods such as SURE and LALO, which also exhibit vulnerability to these issues. To substantiate this point, Table. 9 details how the confidence levels of labels for certain mislabeled samples in the data set Lost change under different confidence initialization strategies (i.e., average or non-uniform). Notably, our observation unveils that both SURE and LALO are susceptible to these two representative errors, indicating that these challenges are not isolated to a single method but are common across various PLL approaches. This observation emphasizes the universal challenge PLL methods face in identifying and correcting mislabeled samples, highlighting a fundamental obstacle in PLL.

## 5 RELATED WORK

### 5.1 Partial Label Learning

Partial label learning (PLL), also known as superset-label learning [25], [29] or ambiguous label learning [11], [21], is a representative weakly supervised learning framework which learns from inaccurate supervision information. In partial label learning, each instance is associated with a set of candidate labels with only one being ground-truth and others being false positive. As the ground-truth label of a sample conceals in the corresponding candidate label set, which can not be directly acquired during the training process, partial label learning task is a quite challenging problem.

To tackle the mentioned challenge, existing works mainly focus on disambiguation [16], [20], [22], [30]–[34], which can be broadly divided into two categories: averaging-based approaches and identification-based approaches. For the averaging-based approaches [2], [21], [30], each candidate label of a training sample is treated equally as the ground-truth one and the final prediction is yielded by averaging the modeling outputs. For instance, PL-KNN [21] averages the candidate labels of neighboring samples to make the prediction. This kind of approach is intuitive, however, it can be easily influenced by false positive candidate labels which results in inferior performance. For identification-based approaches, [15], [20], [22], [35], [36], the ground-truth label

is treated as a latent variable and can be identified through an iterative optimization procedure such as EM. Moreover, labeling confidence based strategy is proposed in many state-of-the-art identification based approaches for better disambiguation. [30] and [16] construct a similarity graph based on the feature space to generate labeling confidence of candidate labels.

Recently, models based on deep learning have been increasingly employed for disambiguation for PLL tasks [24], [37]–[41]. PICO [23] is a contrastive learning-based approach devised to tackle label ambiguity in partial label learning. This method seeks to discern the valid label from the candidate set by utilizing contrastively learned embedding prototypes. Lv et al. proposed PRODEN [24], where the simultaneous updating of the model and identification of true labels are seamlessly integrated. Furthermore, He et al. introduced a partial label learning method based on semantic label representations in [38]. This method employs a novel weighted calibration rank loss to facilitate label disambiguation. By leveraging label confidence, the approach weights the similarity towards all candidate labels and subsequently yields a higher similarity of candidate labels in comparison to each non-candidate label.

### 5.2 Relations to CLL

The concept of leveraging non-candidate labels, also known as complementary labels, is not new in the field of machine learning. Previous research on complementary-label learning (CLL) [42]–[45] primarily focused on scenarios involving a single complementary label. In contrast to CLL, PLL entails generating complementary labels from the set of non-candidate labels, serving the purpose of appeal and disambiguation in PLL.

The research focus on Complementary Labels (CL) in CLL [42]–[45] is primarily motivated by the inherent challenge it presents. Operating within a single CL framework allows for data efficiency but introduces significant complexity. In theory, CLL strategies could be modified for PLL applications. It's crucial, however, to underline the enduring challenges of PLL, even when employing a limited set of labels as candidates. A classifier trained under conditions where only one label is deemed "invalid" while the others are considered "candidates" is prone to sub-optimal performance. Therefore, although CLL methodologies might be theoretically applicable to PLL situations, their practical efficacy in such contexts might be constrained.

### 5.3 Relations to MCLL

Notably, a recent setting known as Multiple Complementary Label Learning (MCLL) [46] has emerged. MCLL introduces the capability to apply the negative learning loss [47] to each CL associated with the same instance. This loss aims to minimize the class-posterior probability of each CL, effectively driving these probabilities towards zero. It's worth noting that in the original formulation of PLL [1], the primary goal was to maximize the cumulative probabilities of all potential candidate labels for each instance towards one, while leaving the precise distribution among these candidates (be it uniform or skewed towards a singular "winner") unspecified. In this sense, MCLL and the original PLL share the exact same problem setting with the same objective.

However, it's essential to recognize that the implemented algorithms may introduce more specific training objectives beyond the default goal, such as identifying the valid labels. This leads to PLL and MCLL being distinct research topics, each with its unique algorithms and nuances.

## 6 CONCLUSION

In this paper, we propose the first appeal-based framework PLCP, which aims to identify and rectify mislabeled samples for the existing PLL classifiers. Specifically, a partner classifier is introduced and a novel collaborative term is designed to link the base classifier and the partner classifier, which enables mutual supervision between the two classifiers. A blurring mechanism is involved in this paradigm for better disambiguation. Comprehensive experiments validate the outstanding performance of PLCP coupling with stand-alone approaches and deep-learning based methods, which further validates that the mislabeled examples can be identified and corrected by PLCP. Further analyses are provided for better understanding of this framework. In the future, it is also interesting to investigate other appeal-based methods to identify and correct mislabeled samples.

## REFERENCES

[1] R. Jin and Z. Ghahramani, "Learning with multiple labels," *Advances in neural information processing systems*, vol. 15, 2002.

[2] T. Cour, B. Sapp, and B. Taskar, "Learning from partial labels," *J. Mach. Learn. Res.*, vol. 12, pp. 1501–1536, 2011. [Online]. Available: https://dl.acm.org/doi/10.5555/1953048.2021049

[3] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. W. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 8536–8546. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/a19744e268754fb0148b017647355b7b-Abstract.html

[4] G. Papandreou, L. Chen, K. P. Murphy, and A. L. Yuille, "Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 2015, pp. 1742–1750. [Online]. Available: https://doi.org/10.1109/ICCV.2015.203

[5] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 4331–4340. [Online]. Available: http://proceedings.mlr.press/v80/ren18a.html

[6] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National science review*, vol. 5, no. 1, pp. 44–53, 2018.

[7] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," 2009. [Online]. Available: https://doi.org/10.2200/S00196ED1V01Y200906AIM006

[8] J. Chai, I. W. Tsang, and W. Chen, "Large margin partial label machine," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 7, pp. 2594–2608, 2020. [Online]. Available: https://doi.org/10.1109/TNNLS.2019.2933530

[9] Y. Li, L. Guo, and Z. Zhou, "Towards safe weakly supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 334–346, 2021. [Online]. Available: https://doi.org/10.1109/TPAMI.2019.2922396

[10] Y. Li and D. Liang, "Safe semi-supervised learning: a brief introduction," *Frontiers Comput. Sci.*, vol. 13, no. 4, pp. 669–676, 2019. [Online]. Available: https://doi.org/10.1007/s11704-019-8452-2

[11] Z. Zeng, S. Xiao, K. Jia, T. Chan, S. Gao, D. Xu, and Y. Ma, "Learning by associating ambiguously labeled images," in *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*. IEEE Computer Society, 2013, pp. 708–715. [Online]. Available: https://doi.org/10.1109/CVPR.2013.97

[12] M. Guillaumin, J. Verbeek, and C. Schmid, "Multiple instance metric learning from automatically labeled bags of faces," in *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds., vol. 6311. Springer, 2010, pp. 634–647. [Online]. Available: https://doi.org/10.1007/978-3-642-15549-9_46

[13] X. Gong, D. Yuan, and W. Bao, "Partial label learning via label influence function," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 2022, pp. 7665–7678. [Online]. Available: https://proceedings.mlr.press/v162/gong22c.html

[14] G. Panis, A. Lanitis, N. Tsapatsoulis, and T. F. Cootes, "Overview of research on facial ageing using the FG-NET ageing database," *IET Biom.*, vol. 5, no. 2, pp. 37–46, 2016. [Online]. Available: https://doi.org/10.1049/iet-bmt.2014.0053

[15] L. Feng and B. An, "Leveraging latent label distributions for partial label learning," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, J. Lang, Ed. ijcai.org, 2018, pp. 2107–2113. [Online]. Available: https://doi.org/10.24963/ijcai.2018/291

[16] D. Wang, M. Zhang, and L. Li, "Adaptive graph guided disambiguation for partial label learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8796–8811, 2022. [Online]. Available: https://doi.org/10.1109/TPAMI.2021.3120012

[17] Y. Jia, C. Si, and M.-l. Zhang, "Complementary classifier induced partial label learning," *arXiv preprint arXiv:2305.09897*, 2023.

[18] T. Cour, B. Sapp, C. Jordan, and B. Taskar, "Learning from ambiguously labeled images," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE Computer Society, 2009, pp. 919–926. [Online]. Available: https://doi.org/10.1109/CVPR.2009.5206667

[19] C. Si, Z. Jiang, X. Wang, Y. Wang, X. Yang, and W. Shen, "Partial label learning with a partner," *arXiv preprint arXiv:2312.11034*, 2023.

[20] L. Feng and B. An, "Partial label learning with self-guided retraining," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2019, pp. 3542–3549. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.33013542

[21] E. Hüllermeier and J. Beringer, "Learning from ambiguously labeled examples," in *Advances in Intelligent Data Analysis VI, 6th International Symposium on Intelligent Data Analysis, IDA 2005, Madrid, Spain, September 8-10, 2005, Proceedings*, ser. Lecture Notes in Computer Science, A. F. Famili, J. N. Kok, J. M. P. Sánchez, A. Siebes, and A. J. Feelders, Eds., vol. 3646. Springer, 2005, pp. 168–179. [Online]. Available: https://doi.org/10.1007/11552253_16

[22] N. Nguyen and R. Caruana, "Classification with partial labels," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, Y. Li, B. Liu, and S. Sarawagi, Eds. ACM, 2008, pp. 551–559. [Online]. Available: https://doi.org/10.1145/1401890.1401958

[23] H. Wang, R. Xiao, Y. Li, L. Feng, G. Niu, G. Chen, and J. Zhao, "Pico: Contrastive label disambiguation for partial label learning," *arXiv preprint arXiv:2201.08984*, 2022.

[24] J. Lv, M. Xu, L. Feng, G. Niu, X. Geng, and M. Sugiyama, "Progressive identification of true labels for partial-label learning," in *international conference on machine learning*. PMLR, 2020, pp. 6500–6510.

[25] L. Liu and T. G. Dietterich, "A conditional multinomial mixture model for superset label learning," pp. 557–565, 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/hash/aaebdb8bb6b0e73f6c3c54a0ab0c6415-Abstract.html

[26] M. J. Huiskes and M. S. Lew, "The MIR flickr retrieval evaluation," in *Proceedings of the 1st ACM SIGMM International Conference on Multimedia Information Retrieval, MIR 2008, Vancouver, British Columbia, Canada, October 30-31, 2008*, M. S. Lew, A. D. Bimbo, and E. M. Bakker, Eds. ACM, 2008, pp. 39–43. [Online]. Available: https://doi.org/10.1145/1460096.1460104

[27] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[28] M. Zhang, B. Zhou, and X. Liu, "Partial label learning via feature-aware disambiguation," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi, Eds. ACM, 2016, pp. 1335–1344. [Online]. Available: https://doi.org/10.1145/2939672.2939788

[29] L. Liu and T. G. Dietterich, "Learnability of the superset label learning problem," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, ser. JMLR Workshop and Conference Proceedings, vol. 32. JMLR.org, 2014, pp. 1629–1637. [Online]. Available: http://proceedings.mlr.press/v32/liug14.html

[30] M. Zhang and F. Yu, "Solving the partial label learning problem: An instance-based approach," pp. 4048–4054, 2015. [Online]. Available: http://ijcai.org/Abstract/15/568

[31] J. Fan, Y. Yu, Z. Wang, and J. Gu, "Partial label learning based on disambiguation correction net with graph representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 8, pp. 4953–4967, 2021.

[32] N. Xu, J. Lv, and X. Geng, "Partial label learning via label enhancement," in *Proceedings of the AAAI Conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 5557–5564.

[33] M.-L. Zhang, J.-H. Wu, and W.-X. Bao, "Disambiguation enabled linear discriminant analysis for partial label dimensionality reduction," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 16, no. 4, pp. 1–18, 2022.

[34] W. Qian, Y. Li, Q. Ye, W. Ding, and W. Shu, "Disambiguation-based partial label feature selection via feature dependency and label consistency," *Information Fusion*, vol. 94, pp. 152–168, 2023.

[35] R. Jin and Z. Ghahramani, "Learning with multiple labels," in *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, 2002, pp. 897–904. [Online]. Available: https://proceedings.neurips.cc/paper/2002/hash/653ac11ca60b3e021a8c609c7198acfc-Abstract.html

[36] F. Yu and M. Zhang, "Maximum margin partial label learning," vol. 106, 2017, pp. 573–593. [Online]. Available: https://doi.org/10.1007/s10994-016-5606-4

[37] N. Xu, C. Qiao, X. Geng, and M.-L. Zhang, "Instance-dependent partial label learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 119–27 130, 2021.

[38] S. He, L. Feng, F. Lv, W. Li, and G. Yang, "Partial label learning with semantic label representations," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 545–553.

[39] D.-D. Wu, D.-B. Wang, and M.-L. Zhang, "Revisiting consistency regularization for deep partial label learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 24 212–24 225.

[40] G. Lyu, Y. Wu, and S. Feng, "Deep graph matching for partial label learning," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2022, pp. 3306–3312.

[41] S. Xia, J. Lv, N. Xu, G. Niu, and X. Geng, "Towards effective visual representations for partial-label learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 589–15 598.

[42] T. Ishida, G. Niu, W. Hu, and M. Sugiyama, "Learning from complementary labels," pp. 5639–5649, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/1dba5eed8838571e1c80af145184e515-Abstract.html

[43] T. Ishida, G. Niu, A. Menon, and M. Sugiyama, "Complementary-label learning for arbitrary losses and models," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2971–2980.

[44] Y. Gao and M.-L. Zhang, "Discriminative complementary-label learning with weighted loss," in *International Conference on Machine Learning*. PMLR, 2021, pp. 3587–3597.

[45] Y. Katsura and M. Uchida, "Bridging ordinary-label learning and complementary-label learning," in *Asian Conference on Machine Learning*. PMLR, 2020, pp. 161–176.

[46] L. Feng, T. Kaneko, B. Han, G. Niu, B. An, and M. Sugiyama, "Learning with multiple complementary labels," in *International Conference on Machine Learning*. PMLR, 2020, pp. 3072–3081.

[47] Y. Kim, J. Yim, J. Yun, and J. Kim, "Nlnl: Negative learning for noisy labels," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 101–110.