PROGRESSIVE PURIFICATION FOR INSTANCE-DEPENDENT PARTIAL LABEL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Partial label learning (PLL) aims to train multi-class classifiers from instances with partial labels (PLs)—a PL for an instance is a set of candidate labels where a fixed but unknown candidate is the true label. In the last few years, the instance-independent generation process of PLs has been extensively studied, on the basis of which many practical and theoretical advances have been made in PLL, whereas relatively less attention has been paid to the practical setting of instance-dependent PLs, namely, the PL depends not only on the true label but the instance itself. In this paper, we propose a theoretically grounded and practically effective approach called PrOgressive Purification (POP) for instance-dependent PLL: in each epoch, POP updates the learning model while purifying each PL for the next epoch of the model training by progressively moving out false candidate labels. Theoretically, we prove that POP enlarges the region appropriately fast where the model is reliable, and eventually approximates the Bayes optimal classifier with mild assumptions; technically, POP is flexible with arbitrary losses and compatible with deep networks, so that the previous advanced PLL losses can be embedded in it and the performance is often significantly improved.

1 INTRODUCTION

Over-parameterized deep neural networks owe their popularity much to their ability to (nearly) perfectly memorize large numbers of training examples, and the memorization is known to decrease the generalization error [Feldman (2020)]. On the other hand, scaling the acquisition of examples for training neural networks inevitably introduces non-fully supervised data annotation, a typical example among which is partial label [Nguyen & Caruana (2008); Cour et al. (2011); Zhang et al. (2016); 2017b; Feng & An (2018); Xu et al. (2019); Yao et al. (2020b); Lv et al. (2020); Feng et al. (2020b); Wen et al. (2021)]—a partial label for an instance is a set of candidate labels where a fixed but unknown candidate is the true label.

Partial label learning (PLL) trains multi-class classifiers from instances that are associated with partial labels. It is therefore apparent that some techniques should be applied to prevent memorizing the false candidate labels when PLL resorts to deep learning, and unfortunately, empirical evidence has shown general-purpose regularization cannot achieve that goal [Lv et al. (2021)].

A large number of deep PLL algorithms have recently emerged that aimed to design regularizers [Yao et al. (2020); Lyu et al. (2022)] or network architectures [Wang et al. (2022a)] for PLL data. Further, there are some PLL works that provided theoretical guarantees while making their methods compatible with deep networks [Lv et al. (2020); Feng et al. (2020b); Wen et al. (2021); Wu & Sugiyama (2021)]. We observe that these existing theoretical works have focused on the instance-independent setting where the generation process of partial labels is homogeneous across training examples. With an explicit formulation of the generation process, the asymptotical consistency [Mohri et al. (2018)] of the methods, namely, whether the classifier learned from partial labels approximates the Bayes optimal classifier, can be analyzed.

However, the instance-independent process cannot model the real world well since data labeling is prone to different levels of error in tasks of varying difficulty. Intuitively, instance-dependent (ID) partial labels should be quite realistic as some poor-quality or ambiguous instances are more difficult to be labeled with an exact true label. Although the instance-independent setting has been extensively studied, on the basis of which many practical and theoretical advances have been made
In PLL, relatively less attention has been paid to the practically relevant setting of ID partial labels. Very recently, one solution has been proposed [Xu et al. (2021)] which learned directly from ID partial labels, nevertheless, it is still unclear in theory whether the learned classifier is good. Motivated by the above observations, we set out to investigate ID PLL with the aim of proposing a learning approach that is model-independent and theoretically explain when and why the proposed method works.

In this paper, we propose PrOgressive Purification (POP), a theoretically grounded PLL framework for ID partial labels. Specifically, we use the observed partial labels to pretrain a randomly initialized classifier (deep network) for several epochs, and then we update both partial labels and the classifier for the remaining epochs. In each epoch, we purify each partial label by moving out the candidate labels for which the current classifier has high confidence of being incorrect, and subsequently we train the classifier with the purified partial labels in the next epoch. As a consequence, the false candidate labels are gradually sifted out and the classification performance of the classifier is improved. We justify POP and outline the main contributions below:

- We propose a novel approach named POP for the ID PLL problem, which purifies the partial labels and refines the classifier iteratively. Extensive experiments validate the effectiveness of POP.
- We prove that POP can be guaranteed to enlarge the region where the model is reliable by a promising rate, and eventually approximates the Bayes optimal classifier with mild assumptions. This proof process does not rely on the assumption of the instance-independent setting. To the best of our knowledge, this is the first theoretically guaranteed approach for the general ID PLL problem.
- POP is flexible with respect to losses, so that the losses designed for the instance-independent PLL problems can be embedded directly. We empirically show that such embedding allows advanced PLL losses can be applied to the ID problem and achieve state-of-the-art learning performance.

2 Related Work

In this section, we briefly go through the seminal works in PLL, focusing on the theoretical works and discussing the underlying assumptions behind them.

Non-deep PLL There have been substantial non-deep PLL algorithms from the pioneering work [Jin & Ghahramani (2003)]. From a practical standpoint, they have been studied along two different research routes: the identification-based strategy and the average-based strategy. The identification-based strategy purifies each partial label and extracts the true label heuristically in the training phase, so as to identify the true labels [Chen et al. (2014); Zhang et al. (2016); Tang & Zhang (2017); Feng & An (2019); Xu et al. (2019)]. On the contrary, the average-based strategy treats all candidates equally [Hüllermeier & Beringer (2006); Cour et al. (2011); Zhang & Yu (2015)]. On the theoretical side, Liu and Dietterich Liu & Dietterich (2012) analyzed the learnability of PLL by making a small ambiguity degree condition assumption, which ensures classification errors on any instance have a probability of being detected. And Cour et al. [Cour et al. (2011)] proposed a consistent approach under the small ambiguity degree condition and a dominance assumption on data distribution (Proposition 5 in Cour et al. (2011)). Liu and Dietterich Liu & Dietterich (2012) proposed a Logistic Stick-Breaking Conditional Multinomial Model to portray the mapping between instances and true labels while assuming the generation of the partial label is independent of the instance itself. It should be noted that the vast majority of non-deep PLL works have only empirically verified the performance of algorithms on small data sets, without formalizing the statistical model for the PLL problem, and therefore even less so for theoretical analysis of when and why the algorithms work.

Deep PLL In recent years, deep learning has been applied to PLL and has greatly advanced the practical application of PLL. Yao et al. [Yao et al. (2020a,b)] and Lv et al. [Lv et al. (2020)] proposed learning objectives that are compatible with stochastic optimization and thus can be implemented by deep networks. Soon Feng et al. [Feng et al. (2020b)] formalized the first generation process for PLL. They assumed that given the latent true label, the probability of all incorrect labels being added into the candidate label set is uniform and independent of the instance. Thanks to the uniform generation process, they proposed two provably consistent algorithms. Wen et al. [Wen et al. (2021)] extended the uniform one to the class-dependent case, but still keep the instance-independent as-
assumption unchanged. In addition, a new paradigm called complementary label learning [Ishida et al. (2017), Yu et al. (2018), Ishida et al. (2019); Feng et al. (2020a) has been proposed that learns from instances equipped with a complementary label. A complementary label specifies the classes to which the instance does not belong, so it can be considered to be an inverted PLL problem. However, all of them made the instance-independent assumption for analyzing the statistic consistency. Wu and Sugiyma [Wu & Sugiyma (2021)] proposed a framework that unifies the formalization of multiple generation processes under the instance-independent assumption. Wang et al. [Wang et al. (2022a)] proposed a data-augmentation-based framework to disambiguate partial labels with contrastive learning. Zhang et al. [Zhang et al. (2021a)] exploited the class activation value to identify the true label in candidate label sets.

Very recently, some researchers are beginning to notice a more general setting—ID PLL. Learning with the ID partial labels is challenging, and all instance-independent approaches cannot handle the ID PLL problem directly. Specifically, the theoretical approaches mentioned above utilize mainly the loss correction technique, which corrects the prediction or the loss of the classifier using a prior or estimated knowledge of data generation processes, i.e., a set of parameters controlling the probability of generating incorrect candidate labels, or it is often called transition matrix [Patrini et al. (2017)]. The transition matrix can be characterized fixedly in the instance-independent setting since it does not need to include instance-level information, a condition that does not hold in ID PLL. Furthermore, it is ill-posed to estimate the transition matrix by only exploiting partially labeled data, i.e., the transition matrix is unidentifiable [Xia et al. (2020)]. Therefore, some new methods should be proposed to tackle this issue. Xu et al. [Xu et al. (2021)] introduced a solution that infers the latent label posterior via variational inference methods [Blei et al. (2017)], nevertheless, its effectiveness would be hardly guaranteed. In this paper, we propose PoP for the ID PLL problem and theoretically prove that the learned classifier approximates well to the Bayes optimal.

3 Proposed Method

3.1 Preliminaries

First of all, we briefly introduce some necessary notations. Consider a multi-class classification problem of $c$ classes. Let $X = \mathbb{R}^q$ be the $q$-dimensional instance space and $Y = \{1, 2, \ldots, c\}$ be the label space with $c$ class labels. In supervised learning, let $p(x, y)$ be the underlying “clean” distribution generating $(x, y^*) \in X \times Y$ from which i.i.d. samples $\{(x_i, y_i^*)\}_{i=1}^n$ are drawn.

In PLL, there is a partial label space $S := \{S|S \subseteq Y, S \neq \emptyset\}$ and the PLL training set $D = \{(x_i, S_i) | 1 \leq i \leq n\}$ is sampled independently and identically from a “corrupted” density $\tilde{p}(x, S)$ over $X \times S$. It is generally assumed that $p(x, y)$ and $p(x, S)$ have the same marginal distribution of instances $p(x)$. Then the generation process of partial labels can thus be formalized as $p(S|x) = \sum_y p(S|x, y)p(y|x)$. We define the probability that, given the instance $x$ and its class label $y^*$, $j$-label being included in its partial label as the flipping probability:

$$\xi^j(x) = p(j \in S|x, y^*), \forall j \in Y,$$

The key definition in PLL is that the latent true label of an instance is always one of its candidate label, i.e., $\xi^{y^*}(x) = 1$.

We consider use deep models by the aid of an inverse link function [Reidand & Williamson (2010) $\phi: \mathbb{R}^c \rightarrow \Delta^{c-1}$ where $\Delta^{c-1}$ denotes the $c$-dimensional simplex, for example, the softmax, as learning model in this paper. Then the goal of supervised multi-class classification and PLL is the same: a scoring function $f: X \rightarrow \Delta^{c-1}$ that can make correct predictions on unseen inputs. Typically, the classifier takes the form:

$$h(x) = \arg \max_{j \in Y} f_j(x).$$

The Bayes optimal classifier $h^*$ (learned using supervised data) is the one that minimizes the risk w.r.t the 0-1 loss (or some classification-calibrated loss [Bartlett et al. (2006)], i.e.,

$$h^* = \arg \min_{h} \mathcal{R}_{01} = \arg \min_{h} \mathbb{E}_{(x, y) \sim p(x, y)} \mathbb{I}_{\{h(x) \neq y\}}.$$

For strictly proper losses [Gneiting & Raftery (2007)], the scoring function $f^*$ recovers the class-posterior probabilities, i.e., $f^*(x) = p(y|x), \forall x \in X$. When the supervision information available
is partial label, the PLL risk under $\tilde{p}(x, S)$ w.r.t. a suitable PLL loss $L : \mathbb{R}^k \times S \rightarrow \mathbb{R}^+$ is defined as
\[ \tilde{R} = \mathbb{E}_{(X, S) \sim \tilde{p}(x, S)}[L(h(X), S)]. \]

Minimizing $\tilde{R}$ induces the classifier and it is desirable that the minimizer approach $h^\star$. In addition, let $o = \arg \max_{j \neq y=\star} p(y = j|x)$ be the class label with the second highest posterior possibility among all labels.

### 3.2 Overview

In the latter part of this section, we will introduce a concept pure level set as the region where the model is reliable. We prove that given a tiny reliable region, one could progressively enlarge this region and improves the model with a sufficient rate by disambiguating the partial labels. Motivated by the theoretical results, we propose an approach POP that works by progressively purifying the partial labels to move out the false candidate labels, and eventually the learned classifier could approximate the Bayes optimal classifier.

POP employs the observed partial labels to pre-train a randomly initialized classifier for several epochs, and then updates both partial labels and the classifier for the remaining epochs. We start with a warm-up period, in which we train the predictive model with a well-defined PLL loss $L_{\text{PLL}}$. This allows us to attain a reasonable predictive model before it starts fitting incorrect labels. After the warm-up period, we iteratively purify each partial label by moving out the candidate labels for which the current classifier has high confidence of being incorrect, and subsequently we train the classifier with the purified partial labels in the next epoch. After the model has been fully trained, the predictive model can perform prediction for unseen instances.

### 3.3 The POP Method

We assume that the hypothesis class $\mathcal{H}$ is sufficiently complex (and deep networks could meet this condition), such that the approximation error equals zero, i.e., $\arg \min_h R = \arg \min_{h \in \mathcal{H}} R$ and we have enough training data i.e., $n \rightarrow \infty$. The classifier is able to at least approximate the Bayes optimal classifier $h^\star$ and the gap between the learned $f(x)$ and the true scoring function $f^\star(x)$ corresponding to $h^\star$ is determined by the inconsistency between incorrect candidate labels and output of the Bayes optimal classifier.

For two instance $x$ and $z$ that satisfy $p(y^\star | z) - p(o | z) > p(y^\star | x) - p(o | x)$, i.e., the margin between the posterior of ground-truth label $p(y^\star | z)$ and the second highest posterior possibility $p(o | z)$ is larger than that in point $x$, the indicator function $\left[ \mathbf{1}_{j \neq h^\star(z)} \right] p(y^\star | z) - p(o | z) \geq p(y^\star | x) - p(o | x)$, $j \in S_z$ equals 1 if the candidate label $j$ of $z$ is inconsistent with the output of the optimal Bayes classifier $h^\star(z)$. Then, the gap between $f_j(x)$ and $f_j^\star(x)$, i.e., the approximation error of the classifier, could be controlled by the inconsistency between the incorrect candidate labels and the output of the Bayes optimal classifier $h^\star$ for all the instances $z$. Therefore, we assume that there exist constants $\alpha$, $\epsilon < 1$, such that for $f(x)$,
\[
|f_j(x) - f_j^\star(x)| \leq \alpha \mathbb{E}_{(X, S) \sim \tilde{p}(x, S)} \left[ \mathbf{1}_{j \neq h^\star(z)} \right] \left| p(y^\star | z) - p(o | z) \right| \geq p(y^\star | x) - p(o | x), j \in S_z \right] + \frac{\epsilon}{6} \tag{1}
\]
where the scoring function $f^\star$ corresponding to $h^\star$ on strictly proper losses [Gneiting & Raftery (2007)] recovers the class-posterior probabilities, i.e., $f_j^\star(x) = p(y = j|x)$. In addition, for the probability density function $d(u)$ of cumulative distribution function $D(u) = P_{x \sim p(x,y)}(u(x) \leq u)$ where $0 \leq u \leq 1$ and the margin $u(x) = p(y^\star | x) - p(o | x)$, we assume that there exist constants $c_\star$, $c^\star > 0$ such that $c_\star < d(u) < c^\star$. Then, the worst-case density-imbalance ratio is denoted by $l = \frac{c^\star}{c_\star}$. As the flipping probability of the incorrect label in the instance-dependent generation process is related to its posterior probability, we assume that there exists a constant $t > 0$ such that:
\[
\xi_j^\star(x) \leq p(y = j|x)t. \tag{2}
\]

Motivated by the pure level set in binary classification [Zhang et al. (2021b)], we define the pure level set in instance-dependent PLL, i.e., the region where the model is reliable:
Definition 1 (Pure \((e, f)\)-level set). A set \(L(e) := \{x \mid p(y^p|x) - p(o|x)| \geq e\}\) is pure for \(f\) if \(y^p = \arg \max_j f_j(x)\) for all \(x \in L(e)\).

Assume that there exists a set \(L(e)\) for all \(x \in L(e)\) which satisfies \(y^p = \arg \max_j f_j(x)\), we have
\[
\mathbb{E}_{(z,S) \sim \tilde{p}(z,S)} \left[ 1_{\{j \neq h^*(z)\}} \right] p(y^z|x) - p(o|x) \geq p(y^f|x) - p(o|x), \ j \in S_z = \emptyset
\]
which means that there is a tiny region \(L(e)\) where the model \(f\) is reliable.

Let \(e_{\text{new}}\) be the new boundary and \(\frac{\partial}{\partial e} (p(y^f|x) - e) \leq e - e_{\text{new}} \leq \frac{\partial}{\partial e} (p(y^p|x) - e)\). As the probability density function \(d(u)\) of the margin \(u(x) = p(y^p|x) - p(o|x)\) is bounded by \(c_s < d(u) < c^*\), we have the following result for \(x\) that satisfies \(e > p(y^p|x) - p(o|x) \geq e_{\text{new}}\)
\[
\mathbb{E}_{(z,S) \sim \tilde{p}(z,S)} \left[ 1_{\{j \neq h^*(z)\}} \right] p(y^z|x) - p(o|x) \geq p(y^f|x) - p(o|x), \ j \in S_z \leq \frac{e}{3\alpha}.
\]
Combining Eq. (1) and Eq. (4), there is
\[
|f_j(x) - f_j^*(x)| \leq \frac{\epsilon}{2}.
\]
Denote by \(m = \arg \max_j f_j(x)\) the label with the highest posterior probability for the current prediction. If \(f_m(x) - f_{j \neq m}(x) \geq e + \epsilon\), we have
\[
p(y^p|x) \geq p(y = j|x) + \epsilon
\]
which means that the label \(j\) is incorrect label. Therefore, we could move the label \(j\) out from the candidate label set to disambiguate the partial label, and then refine the learning model with the partial label with less ambiguity. In this way, we would move one step forward by trusting the model with the tiny reliable region with following theorem.

We start with a warm-up period, as the classifier is able to attain reasonable outputs before fitting label noise [Zhang et al. (2017)]. Note that the warm-up training is employed to find a tiny reliable region and the ablation experiments show that the performance of POP does not rely on the warm-up strategy. The predictive model \(\theta\) could be trained on partially labeled examples by minimizing any PLL loss function. Here we adopt PRODEN loss [Lv et al. (2020)] to find a tiny reliable region:
\[
L_{PLL} = \sum_{i=1}^{n} \sum_{j=1}^{c} w_{ij} \ell(f_j(x_i), S_i).
\]
Here, \(\ell\) is the cross-entropy loss and the weight \(w_{ij}\) is initialized with with uniform weights and then could be tackled simply using the current predictions for slightly putting more weights on more possible labels [Lv et al. (2020)]:
\[
w_{ij} = \left\{ \begin{array}{ll} f_j (x_i) / \sum_{j \in S_i} f_j (x_i) & \text{if } j \in S_i \text{ otherwise} \end{array} \right.
\]

Theorem 1 Assume that we have enough training data\(n \rightarrow \infty\) and there is a pure \((e, f)\)-level set where \(x \in L(e)\) can be correctly classified by \(f\). For each \(x\) and \(j \in S\), if \(f_m(x) - f_j(x) \geq e + \epsilon\), if we move out label \(j\) from the candidate label set and then update the candidate label set as \(S_{\text{new}}\). Then the new classifier \(f_{\text{new}}(x)\) is trained on the updated data with the new distribution \(\tilde{p}(x, S_{\text{new}})\). Let \(e_{\text{new}}\) be the minimum boundary that \(L(e_{\text{new}})\) is pure for \(f_{\text{new}}\). Then, we have
\[
p(y^p|x) - e_{\text{new}} \geq (1 + e/6\alpha) p(y^p|x) - e).
\]

The detailed proof can be found in Appendix A.1. Theorem 1 shows that the purified region \(\gamma = p(y^p|x) - e\) would be enlarged by at least a constant factor with the given purification strategy.

---

1 More details could be found in Appendix A.1.
2 More details could be found in Appendix A.2.
Algorithm 1 POP Algorithm

Input: The PLL training set \( \mathcal{D} = \{(x_1, S_1), \ldots, (x_n, S_n)\} \), initial threshold \( e_0 \), end threshold \( e_{\text{end}} \), total round \( R \), step-size \( e_s \);

1: Initialize the predictive model \( \theta \) by warm-up training with the PLL loss Eq. 7, and threshold \( e = e_0 \);
2: for \( r = 1, \ldots, R \) do
3:   Train the predictive model \( f \) on \( \mathcal{D} \);
4: for \( i = 1, \ldots, n \) do
5:   for \( j \in S_i \) do
6:     if \( f_{m_i}(x_i) - f_j(x_i) \geq e + \epsilon \) then
7:       Purify the incorrect label \( j \) by removing it from the candidate label set \( S_i \);
8:     end if
9:   end for
10: end for
11: if \( e \leq e_{\text{end}} \), and there is no purification for any candidate label set then
12:   Decrease \( e \) with step-size \( e_s \);
13: end if
14: end for

Output: The final predictive model \( f \)

After the warm-up period, the classifier could be employed for purification. According to Theorem 1, we could progressively move out the incorrect candidate label with the continuously strict bound, and subsequently train an effective classifier with the purified labels with the PLL loss [Lv et al. (2020)] since the PLL loss [Lv et al. (2020)] is model-independent and could operate in a mini-batched training manner to update the model with the labeling-confidence weight. Specifically, we set a high threshold \( e_0 \) and calculate the difference \( f_{m_i}(x_i) - f_j(x_i) \) for each candidate label. If there is a label \( j \) for \( x_i \) satisfies \( f_{m_i}(x_i) - f_j(x_i) \geq e_0 \), we move out it from the candidate label set and update the candidate label set. We depart from the theory by reusing the same fixed dataset over and over, but the empirical results are reasonable.

If there is no purification for all partial labels, we begin to decrease the threshold \( e \) and continue the purification for improving the training of the model. In this way, the incorrect candidate labels are progressively removed from the partial label round by round, and the performance of the classifier is continuously improved. The algorithmic description of POP is shown in Algorithm 1.

Then we prove that if there exists a pure level set for an initialized model, our proposed approach can purify incorrect labels and the classifier \( f \) will finally match the Bayes optimal classifier \( h \) after sufficient rounds \( R \) under the instance-dependence PLL setting.

**Theorem 2** For any flipping probability of each incorrect label \( \xi_i(x) \), define \( e_0 = \frac{(1+t)\alpha + \epsilon}{1+\alpha} \). And for a given function \( f_0 \) there exists a level set \( L(e_0) \) which is pure for \( f_0 \). If one runs purification in Theorem 2 with enough training data \( (n \to \infty) \) starting with \( f_0 \) and the initialization: (1) \( e_0 \geq \frac{(1+t)\alpha + \epsilon}{1+\alpha} \), (2) \( R \geq \frac{6l}{\epsilon} \log\left(\frac{1}{\epsilon e_0}\right) \), (3) \( e_{\text{end}} \geq \epsilon \), then we have:

\[
\mathbb{P}_{x \sim \mathcal{D}}[y_{f_{\text{final}}}(x) = h^*(x)] \geq 1 - c^* \epsilon
\]

The proof of Theorem 2 is provided in Appendix A.3. According to Theorem 2, the learned classifier under the instance-dependent PLL setting will be consistent with the Bayes optimal classifier eventually. Theorem 2 shows that the classifier can be guaranteed to eventually approximate the Bayes optimal classifier.

4 EXPERIMENTS

4.1 DATASETS

We adopt five widely used benchmark datasets including MNIST [LeCun et al. (1998)], Kuzushiji-MNIST [Clanuwat et al. (2018)], Fashion-MNIST [Xiao et al. (2017)], CIFAR-10 [Krizhevsky & Hinton].
Table 1: Classification accuracy (mean±std) of each comparing approach on benchmark datasets corrupted by the ID generation process.

<table>
<thead>
<tr>
<th></th>
<th>MNIST</th>
<th>Kuzushiji-MNIST</th>
<th>Fashion-MNIST</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>POP</td>
<td>99.28±0.02%</td>
<td>91.09±0.14%</td>
<td>96.93±0.07%</td>
<td>93.00±0.26%</td>
<td>71.82±0.08%</td>
</tr>
<tr>
<td>VALEN</td>
<td>99.03±0.02%</td>
<td>90.15±0.02%</td>
<td>96.31±0.12%</td>
<td>92.01±0.09%</td>
<td>71.48±0.12%</td>
</tr>
<tr>
<td>RCC</td>
<td>98.81±0.07%</td>
<td>90.62±0.22%</td>
<td>96.64±0.10%</td>
<td>86.11±0.43%</td>
<td>71.07±0.25%</td>
</tr>
<tr>
<td>PICO</td>
<td>98.76±0.04%</td>
<td>88.87±0.06%</td>
<td>94.83±0.17%</td>
<td>89.35±0.17%</td>
<td>66.30±0.24%</td>
</tr>
<tr>
<td>PRODEN</td>
<td>99.01±0.02%</td>
<td>90.48±0.14%</td>
<td>96.14±0.07%</td>
<td>78.87±0.26%</td>
<td>55.59±0.08%</td>
</tr>
<tr>
<td>RC</td>
<td>99.09±0.09%</td>
<td>90.56±0.14%</td>
<td>96.17±0.08%</td>
<td>80.13±0.14%</td>
<td>56.41±0.17%</td>
</tr>
<tr>
<td>CC</td>
<td>99.08±0.10%</td>
<td>90.40±0.20%</td>
<td>96.12±0.10%</td>
<td>76.17±0.11%</td>
<td>56.48±0.06%</td>
</tr>
<tr>
<td>LW</td>
<td>98.98±0.05%</td>
<td>89.82±0.2%</td>
<td>93.23±0.08%</td>
<td>43.16±0.63%</td>
<td>49.63±0.12%</td>
</tr>
<tr>
<td>CAVL</td>
<td>98.95±0.05%</td>
<td>87.85±0.06%</td>
<td>95.84±0.06%</td>
<td>75.41±0.77%</td>
<td>58.17±0.11%</td>
</tr>
<tr>
<td>CLPL</td>
<td>98.83±0.05%</td>
<td>90.21±0.08%</td>
<td>93.18±0.08%</td>
<td>51.61±0.39%</td>
<td>30.84±0.40%</td>
</tr>
</tbody>
</table>

Table 2: Classification accuracy (mean±std) of each comparing approach on the real-world datasets.

<table>
<thead>
<tr>
<th></th>
<th>Lost</th>
<th>BirdSong</th>
<th>MSRCv2</th>
<th>Mirflickr</th>
<th>Malagasy</th>
<th>Soccer Player</th>
<th>Yahoo!News</th>
</tr>
</thead>
<tbody>
<tr>
<td>POP</td>
<td>78.57±0.45%</td>
<td>74.47±0.36%</td>
<td>45.86±0.28%</td>
<td>61.09±0.10%</td>
<td>72.29±0.33%</td>
<td>54.48±0.10%</td>
<td>66.38±0.07%</td>
</tr>
<tr>
<td>VALEN</td>
<td>76.87±0.86%</td>
<td>73.39±0.26%</td>
<td>49.97±0.43%</td>
<td>59.13±0.12%</td>
<td>69.44±0.06%</td>
<td>55.81±0.10%</td>
<td>66.26±0.13%</td>
</tr>
<tr>
<td>PRODEN</td>
<td>76.47±0.25%</td>
<td>73.44±0.12%</td>
<td>45.10±0.16%</td>
<td>59.59±0.52%</td>
<td>69.34±0.09%</td>
<td>54.05±0.15%</td>
<td>66.14±0.10%</td>
</tr>
<tr>
<td>RC</td>
<td>76.26±0.46%</td>
<td>69.33±0.32%</td>
<td>49.47±0.43%</td>
<td>58.93±0.10%</td>
<td>70.69±0.14%</td>
<td>56.02±0.59%</td>
<td>63.51±0.20%</td>
</tr>
<tr>
<td>CC</td>
<td>63.54±0.25%</td>
<td>69.90±0.58%</td>
<td>41.50±0.44%</td>
<td>58.81±0.54%</td>
<td>69.53±0.34%</td>
<td>49.07±0.36%</td>
<td>54.86±0.48%</td>
</tr>
<tr>
<td>LW</td>
<td>73.13±0.32%</td>
<td>51.45±0.26%</td>
<td>49.85±0.49%</td>
<td>54.50±0.81%</td>
<td>59.34±0.25%</td>
<td>50.24±0.45%</td>
<td>48.21±0.29%</td>
</tr>
<tr>
<td>CAVL</td>
<td>73.96±0.51%</td>
<td>69.63±0.93%</td>
<td>46.62±1.29%</td>
<td>57.13±0.10%</td>
<td>65.82±0.06%</td>
<td>52.92±0.40%</td>
<td>60.97±0.13%</td>
</tr>
<tr>
<td>CLPL</td>
<td>63.39±0.12%</td>
<td>62.90±3.33%</td>
<td>37.8±0.71%</td>
<td>58.87±0.10%</td>
<td>64.25±0.29%</td>
<td>48.23±0.03%</td>
<td>49.42±0.13%</td>
</tr>
</tbody>
</table>

(2009). CIFAR-100 Krizhevsky & Hinton (2009). These datasets are manually corrupted into ID partially labeled versions. Specifically, we set the flipping probability of each incorrect label corresponding to an instance \( x \) by using the confidence prediction of a neural network trained using supervised data parameterized by \( \hat{\theta} \). The flipping probability \( \xi_i(x) = \frac{f_j(x; \hat{\theta})}{\max_{j \in Y} f_j(x; \hat{\theta})} \), where \( Y_i \) is the set of all incorrect labels except for the true label of \( x_i \). The average number of candidate labels (avg. #CLS) for each benchmark dataset corrupted by the ID generation process is recorded in Appendix A.4.

In addition, five real-world PLL datasets which are collected from different application domains are used, including Lost Cour et al. (2011), Soccer Player Zeng et al. (2013), Yahoo!News Guillaumin et al. (2010), MSRCv2 Liu & Dietterich (2012), and BirdSong Briggs et al. (2012). The average number of candidate labels (avg. #CLS) for each real-world PLL dataset is also recorded in Appendix A.4.

4.2 Baselines

The performance of POP is compared against five deep PLL approaches:

- **PRODEN** [Lv et al. (2020)]: A progressive identification approach which approximately minimizes a risk estimator and identifies the true labels in a seamless manner;
- **RC** [Feng et al. (2020b)]: A risk-consistent approach which employs the loss correction strategy to establish the true risk by only using the partially labeled data;
- **CC** [Feng et al. (2020b)]: A classifier-consistent approach which also uses the loss correction strategy to learn the classifier that approaches the optimal one;
- **VALEN** [Yao et al. (2020a)]: An ID PLL approach which recovers the latent label distribution via variational inference methods;
- **LW** [Wen et al. (2021)]: A risk-consistent approach which proposes a leveraged weighted loss to trade off the losses on candidate labels and non-candidate ones.
- **CAVL** [Zhang et al. (2021a)]: A progressive identification approach which exploits the class activation value to identify the true label in candidate label sets.
- **CLPL** [Cour et al. (2011)]: A averaging-based disambiguation approach based on a convex learning formulation.
Table 3: Classification accuracy (mean±std) of each comparing approach on benchmark datasets corrupted by the ID generation process.

<table>
<thead>
<tr>
<th></th>
<th>MNIST</th>
<th>Kuzushiji-MNIST</th>
<th>Fashion-MNIST</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODEN</td>
<td>97.70±0.03%</td>
<td>87.60±0.23%</td>
<td>87.21±0.11%</td>
<td>76.77±0.63%</td>
<td>55.12±0.12%</td>
</tr>
<tr>
<td>PRODEN+PO</td>
<td>97.87±0.04%</td>
<td>88.70±0.02%</td>
<td>87.62±0.04%</td>
<td>79.00±0.28%</td>
<td>57.68±0.14%</td>
</tr>
<tr>
<td>RC+PO</td>
<td>97.72±0.02%</td>
<td>87.25±0.06%</td>
<td>87.06±0.14%</td>
<td>76.49±0.52%</td>
<td>55.18±0.70%</td>
</tr>
<tr>
<td>RC</td>
<td>98.08±0.03%</td>
<td>87.78±0.09%</td>
<td>87.45±0.05%</td>
<td>78.89±0.17%</td>
<td>57.66±0.11%</td>
</tr>
<tr>
<td>CC</td>
<td>97.25±0.11%</td>
<td>83.31±0.07%</td>
<td>86.01±0.13%</td>
<td>72.87±0.82%</td>
<td>55.56±0.23%</td>
</tr>
<tr>
<td>CC+PO</td>
<td>97.99±0.06%</td>
<td>83.98±0.10%</td>
<td>86.32±0.06%</td>
<td>77.03±0.58%</td>
<td>56.18±0.06%</td>
</tr>
<tr>
<td>LW</td>
<td>96.80±0.07%</td>
<td>84.46±0.22%</td>
<td>86.25±0.01%</td>
<td>46.77±0.65%</td>
<td>48.00±0.16%</td>
</tr>
<tr>
<td>LW+PO</td>
<td>97.47±0.06%</td>
<td>84.71±0.07%</td>
<td>86.40±0.05%</td>
<td>48.54±0.04%</td>
<td>49.61±0.27%</td>
</tr>
<tr>
<td>CAVL</td>
<td>96.25±0.40%</td>
<td>79.38±0.69%</td>
<td>84.66±0.05%</td>
<td>62.69±0.65%</td>
<td>47.35±0.16%</td>
</tr>
<tr>
<td>CAVL+PO</td>
<td>96.71±0.11%</td>
<td>79.83±0.12%</td>
<td>85.04±0.10%</td>
<td>63.12±0.23%</td>
<td>47.61±0.06%</td>
</tr>
<tr>
<td>CLPL</td>
<td>96.11±0.21%</td>
<td>83.31±0.24%</td>
<td>83.16±0.25%</td>
<td>53.61±0.31%</td>
<td>22.31±0.11%</td>
</tr>
<tr>
<td>CLPL+PO</td>
<td>96.51±0.22%</td>
<td>83.63±0.11%</td>
<td>83.71±0.15%</td>
<td>54.22±0.51%</td>
<td>23.37±0.29%</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>Lost</th>
<th>BirdSong</th>
<th>MSRCv2</th>
<th>MIRfilickr</th>
<th>Malagasy</th>
<th>Soccer Player</th>
<th>Yahoo! News</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODEN</td>
<td>76.47±0.25%</td>
<td>73.44±0.12%</td>
<td>45.10±0.16%</td>
<td>59.59±0.52%</td>
<td>69.34±0.09%</td>
<td>54.05±0.15%</td>
<td>66.14±0.10%</td>
</tr>
<tr>
<td>PRODEN+PO</td>
<td>78.57±0.45%</td>
<td>74.47±0.36%</td>
<td>45.86±0.28%</td>
<td>61.09±0.10%</td>
<td>72.29±0.33%</td>
<td>54.48±0.10%</td>
<td>66.38±0.07%</td>
</tr>
<tr>
<td>RC</td>
<td>76.26±0.46%</td>
<td>69.33±0.32%</td>
<td>49.47±0.43%</td>
<td>58.93±0.10%</td>
<td>70.69±0.14%</td>
<td>56.02±0.59%</td>
<td>63.51±0.20%</td>
</tr>
<tr>
<td>RC+PO</td>
<td>78.56±0.45%</td>
<td>70.77±0.26%</td>
<td>51.18±0.59%</td>
<td>59.65±0.52%</td>
<td>71.04±0.10%</td>
<td>56.49±0.03%</td>
<td>63.86±0.22%</td>
</tr>
<tr>
<td>CC</td>
<td>63.54±0.25%</td>
<td>69.90±0.58%</td>
<td>41.50±0.44%</td>
<td>58.81±0.54%</td>
<td>69.53±0.34%</td>
<td>49.07±0.36%</td>
<td>54.86±0.48%</td>
</tr>
<tr>
<td>CC+PO</td>
<td>65.47±0.93%</td>
<td>71.50±0.06%</td>
<td>43.21±0.43%</td>
<td>59.89±0.48%</td>
<td>71.19±0.40%</td>
<td>49.36±0.02%</td>
<td>55.22±0.05%</td>
</tr>
<tr>
<td>LW</td>
<td>73.13±0.32%</td>
<td>51.45±0.26%</td>
<td>49.85±0.49%</td>
<td>54.50±0.81%</td>
<td>59.34±0.25%</td>
<td>50.24±0.45%</td>
<td>48.21±0.29%</td>
</tr>
<tr>
<td>LW+PO</td>
<td>75.30±0.26%</td>
<td>52.35±0.26%</td>
<td>52.42±0.86%</td>
<td>55.46±0.27%</td>
<td>60.85±0.57</td>
<td>50.94±0.47%</td>
<td>48.6±0.12%</td>
</tr>
<tr>
<td>CAVL</td>
<td>73.96±0.51%</td>
<td>69.63±0.93%</td>
<td>46.62±1.29%</td>
<td>57.13±0.10%</td>
<td>65.82±0.06%</td>
<td>52.92±0.40%</td>
<td>60.97±0.13%</td>
</tr>
<tr>
<td>CAVL+PO</td>
<td>75.32±0.11%</td>
<td>70.13±0.22%</td>
<td>46.92±0.13%</td>
<td>58.63±0.48%</td>
<td>67.70±0.19%</td>
<td>53.44±0.10%</td>
<td>61.37±0.11%</td>
</tr>
<tr>
<td>CLPL</td>
<td>63.39±0.12%</td>
<td>62.90±3.33%</td>
<td>37.8±0.71%</td>
<td>58.87±0.10%</td>
<td>64.25±0.29%</td>
<td>48.23±0.03%</td>
<td>49.42±0.13%</td>
</tr>
<tr>
<td>CLPL+PO</td>
<td>64.73±0.14%</td>
<td>64.06±0.48%</td>
<td>39.32±0.24%</td>
<td>60.31±0.27%</td>
<td>66.04±0.25%</td>
<td>49.11±0.21%</td>
<td>50.33±0.18%</td>
</tr>
</tbody>
</table>

- **PICO** [Wang et al. (2022b)]: a data-augmentation-based method which identifies the true label via contrastive-learning with learned prototypes for image datasets.
- **RCR** [Wu et al. (2022)]: a data-augmentation-based method which identifies the true label via consistency regularization with random augmented instances for image datasets.

For the benchmark datasets, we use the same data augmentation strategy for the data-augmentation-free methods (VALEN, PRODEN, RC, CC, LW and CAVL) to make fair comparisons with the data-augmentation-based methods (PICO and RCR). However, data augmentation cannot be employed on the realworld datasets that contain extracted feature from audio and video data, we just compared our methods with the data-augmentation-free methods on realworld datasets.

For all the deep approaches, We used the same training/validation setting, models, and optimizer for fair comparisons. Specifically, a 5-layer LeNet is trained on MNIST, Kuzushiji-MNIST and Fashion-MNIST, the Wide-ResNet-28-2 [Zagoruyko & Komodakis (2016)] is trained on CIFAR-10 and CIFAR-100, and the linear model is trained on real-world PLL datasets, respectively. The hyper-parameters are selected so as to maximize the accuracy on a validation set (10% of the training set). We run 5 trials on the benchmark datasets and the real-world PLL datasets. The mean accuracy as well as standard deviation are recorded for all comparing approaches. All the comparing methods are implemented with PyTorch.

4.3 Experimental Results

Table 1 and Table 2 report the classification accuracy of each approach on benchmark datasets corrupted by the ID generation process and the real-world PLL datasets, respectively. Due to the inability of data augmentation to be employed on extracted feature, we didn’t compare our methods with PICO and RCR on realworld datasets. The best results are highlighted in bold. We can observe
that POP achieves the best performance against other approaches in most cases and the performance advantage of POP over comparing approaches is stable under varying the number of candidate labels.

In addition, to analysis the purified region in Theorem 1, we employ the confidence predictions of $f(x, \theta)$ (the network in Section 4.1) as the posterior and plot the curve of the estimated purified region in every epoch on Lost in Figure 1. We can see that although the estimated purified region would be not accurate enough, the curve could show that the trend of continuous increase for the purified region.

4.4 FURTHER ANALYSIS

As the framework of POP is flexible for the loss function, we integrate the proposed method with the previous methods for instance-independent PLL including PRODEN, RC, CC, LW, CAVL and CLPL. In this subsection, we empirically prove that the previous methods for instance-independent PLL could be promoted to achieve better performance after integrating with POP.

Table 3 and Table 4 report the classification accuracy of each method for instance-independent PLL and its variant integrated with POP on benchmark datasets corrupted by the ID generating procedure and the real-world datasets, respectively. We didn’t use any data augmentation on benchmark datasets in this part of experiments. As shown in Table 3 and Table 4, the approaches integrated with POP including PRODEN+POP, RC+POP, CC+POP, LW+POP, CAVL+POP and CLPL+POP achieve superior performance against original method, which clearly validates the usefulness of POP framework for improving performance for ID PLL.

Figure 3 illustrates the variant integrated with POP performs under different hyper-parameter configurations on CIFAR-10 while similar observations are also made on other data sets. The hyper-parameter sensitivity on other datasets could be founded in Appendix A.4. As shown in Figure 3, it is obvious that the performance of the variant integrated with POP is relatively stable across a broad range of each hyper-parameter. This property is quite desirable as POP framework could achieve robust classification performance.

5 CONCLUSION

In this paper, the problem of partial label learning is studied where a novel approach POP is proposed. We consider ID partial label learning and propose a theoretically-guaranteed approach, which could train the classifier with progressive purification of the candidate labels and is theoretically guaranteed to eventually approximates the Bayes optimal classifier for ID PLL. Experiments on benchmark and real-world datasets validate the effectiveness of the proposed method. If PLL methods become very effective, the need for exactly annotated data would be significantly reduced. As a result, the employment of data annotators might be decreased which could lead to a negative societal impact.
REFERENCES


A Appendix

A.1 Proofs of Theorem 1

Assume that there exists a set \( L(e) \) for all \( x \in L(e) \) which satisfies \( y^x = \arg \max_j f_j(x) \) and \( p(y^x|x) - p(o|x) \geq e \), we have

\[
\mathbb{E}_{(z,S) \sim \hat{p}(z,S_{\text{new}})} \left[ \mathbf{1}_{j \neq h^*(z)} \right] p(y^z|z) - p(o|z) \geq p(y^x|x) - p(o|x), \quad j \in S_z \quad (9)
\]

Let \( e_{\text{new}} \) be the new boundary and \( \frac{\partial}{\partial e} (p(y^x|x) - e) \leq e - e_{\text{new}} \leq \frac{\partial}{\partial e} (p(y^x|x) - e) \). As the probability density function \( d(u) \) of the margin \( u(x) = p(y^x|x) - p(o|x) \) is bounded by \( c < \)
Due to that \( d(\mu) < \epsilon^* \), we have the following result for \( x \) that satisfies \( p(y^x|x) - p(o|x) \geq \epsilon_{new} \):

\[
E_{(z,S) \sim \tilde{p}(z,S_{new})} \left[ 1_{j \neq h^*(z)} \right] j \in S_z, p(y^\tilde{z}|z) - p(o|z) \geq p(y^\tilde{z}|x) - p(o|x)
\]

\[
\leq E_{(z,S) \sim \tilde{p}(z,S_{new})} \left[ 1_{j \neq h^*(z)} \right] p(y^\tilde{z}|z) - p(o|z) \geq p(y^\tilde{z}|x) - p(o|x)
\]

\[
= p_a \left[ j \neq h^*(z) \right] p(y^\tilde{z}|z) - p(o|z) \geq p(y^\tilde{z}|x) - p(o|x)
\]

\[
\leq p_a \left[ j \neq h^*(z) \right] p(y^\tilde{z}|z) - p(o|z) \geq \epsilon
\]

\[
+ \frac{p_a \left[ j \neq h^*(z), \epsilon_{new} \leq p(y^\tilde{z}|z) - p(o|z) < \epsilon \right] \epsilon_{new}}{p_a \left[ p(y^\tilde{z}|z) - p(o|z) \geq p(y^\tilde{z}|x) - p(o|x) \right]}
\]

\[
= \frac{c^* (e - \epsilon_{new})}{c_a (p(y^\tilde{z}|x) - e)}
\]

Due to that \( \frac{1}{\alpha} (p(y^\tilde{z}|x) - e) \leq e - \epsilon_{new} \leq \frac{1}{\alpha} (p(y^\tilde{z}|x) - e) \) holds, we can further relax Eq. (10) as follows:

\[
E_{(z,S) \sim \tilde{p}(z,S_{new})} \left[ 1_{(h(z) \neq y^* \neq)} \right] p(y^\tilde{z}|z) - p(o|z) \geq p(y^\tilde{z}|x) - p(o|x)
\]

\[
\leq \frac{c^* (e - \epsilon_{new})}{c_a (p(y^\tilde{z}|x) - e)}
\]

\[
\leq \frac{c^*}{c_a (p(y^\tilde{z}|x) - e)} \frac{\epsilon}{3\alpha} (p(y^\tilde{z}|x) - e)
\]

\[
= \frac{\epsilon}{3\alpha}.
\]

Then, we can find that the assumption that the gap between \( f_j(x) \) and \( f_j^*(x) \) should be controlled by the risk at point \( z \) implies:

\[
\left| f_j(x) - f_j^*(z) \right|
\]

\[
\leq \alpha E_{(z,S) \sim \tilde{p}(z,S_{new})} \left[ 1_{(h(z) \neq y^*)} \right] p(y^\tilde{z}|z) - p(o|z) \geq p(y^\tilde{z}|x) - p(o|x) + \frac{\epsilon}{6}
\]

\[
= \alpha \frac{\epsilon}{3\alpha} + \frac{\epsilon}{6}
\]

(12)

Hence, for \( x \) s.t. \( p(y^\tilde{z}|x) - p(o|x) \geq \epsilon_{new} \), according to Eq. (12) we have

\[
f_{y^\tilde{z}}(x) - f_{y^\tilde{z}}(x) \geq (p(y = y^\tilde{z}|x) - \frac{\epsilon}{2}) - (p(y = j|x) + \frac{\epsilon}{2})
\]

\[
= p(y = y^\tilde{z}|x) - p(y = j|x) - \epsilon
\]

\[
\geq p(y = y^\tilde{z}|x) - p(o|x) - \epsilon
\]

\[
\geq \epsilon_{new} - \epsilon
\]

\[
\geq 0,
\]

Details of Eq. (3) in the paper submission
To begin with, we prove that there exists at least a level set $L(e_{\text{new}})$ is pure for $f$. Meanwhile, the choice of $e_{\text{new}}$ ensures that

$$p(y^a|\mathbf{x}) - e_{\text{new}} \geq p(y^m|\mathbf{x}) - (e - \frac{\epsilon}{6l\alpha}(p(y^a|\mathbf{x}) - e))$$

$$= p(y^a|\mathbf{x}) - e + \frac{\epsilon}{6l\alpha}(p(y^a|\mathbf{x}) - e)$$

$$(14)$$

Here, the proof of Theorem 1 has been completed.

A.2 Details of Eq. (5)

If $f_m(\mathbf{x}) - f_j \neq m \geq e + \epsilon$, according to Eq. (12), we have:

$$p(y^a|\mathbf{x}) \geq p(y = m|\mathbf{x})$$

$$= p(y = j|\mathbf{x}) + p(y = m|\mathbf{x}) - p(y = j|\mathbf{x})$$

$$\geq p(y = j|\mathbf{x}) + p(y = m|\mathbf{x}) - p(y = j|\mathbf{x})$$

$$\geq p(y = j|\mathbf{x}) + (f_m(\mathbf{x}) - \epsilon) - (f_j(\mathbf{x}) + \epsilon)$$

$$(15)$$

A.3 Proofs of Theorem 2

To begin with, we prove that there exists at least a level set $L(e_0)$ pure to $f_0$. Considering $\mathbf{x}$ satisfies $p(y^a|\mathbf{x}) - p(o|\mathbf{x}) \geq e_0$, we have $\mathbb{P}_z \left[ j \neq h^*(z) \left| j \in S_z, p(y^a|z) - p(o|z) \geq e_0 \right. \right] \leq p(y^a|z) - e_0 + \xi^j(z)$.

Due to the assumption $|f_j(\mathbf{x}) - f^*_j(\mathbf{x})| \leq \alpha \mathbb{E}_{(z,S) \sim \tilde{p}(z,S)} \left[ 1_{(j \neq h^*(z))} \right] j \in S_z, p(y^a|z) - p(o|z) \geq p(y^a|z) - p(o|z) \right]$ + $\xi^j(z)$, it suffices to satisfy $\alpha \mathbb{E}_{(z,S) \sim \tilde{p}(z,S)} \left[ 1_{(j \neq h^*(z))} \right] j \in S_z, p(y^a|z) - p(o|z) \geq p(y^a|z) - p(o|z) \right] + \xi^j(z)$ to ensure that $f_j(\mathbf{x})$ has the same prediction with $h^*$ when $p(y^a|\mathbf{x}) - p(o|\mathbf{x}) \geq e_0$. Since we have $\xi^j(\mathbf{x}) \leq p(y = j|\mathbf{x}) t \leq p(y^a|\mathbf{x}) t$, by choosing $e_0 \geq \frac{1+\epsilon}{1+\alpha} \geq \frac{1+\epsilon}{1+\alpha} p(y^a|\mathbf{x}) t$ one can ensure that initial $f_0$ has a pure $L(e_0)$-level set.

Then in the rest of the iterations we ensure the level set $p(y^a|z) - p(o|z) \geq e$ is pure. We decrease $e$ by a reasonable factor to avoid incurring too many corrupted labels while ensuring enough progress in label purification, i.e. $\frac{1}{6l\alpha}(p(y^a|\mathbf{x}) - e) \leq e - e_{\text{new}} \leq \frac{1}{6l\alpha}(p(y^a|\mathbf{x}) - e)$, such that in the

Figure 3: Hyper-parameter sensitivity on Lost.
level set \( p(y^\pi|x) - p(o|x) \geq \epsilon_{new} \), we have \( |f_j^*(x) - f_j^*(x)| \leq \frac{\epsilon}{2} \). This condition ensures the correctness of flipping when \( \epsilon \geq \epsilon \). The the purified region cannot be improved once \( \epsilon < \epsilon \) since there is no guarantee that \( f_j^*(x) \) has consistent label with \( h^* \) when \( p(y^\pi|x) - p(o|x) < \epsilon \) and \( |f_j^*(x) - f_j^*(x)| \leq \frac{\epsilon}{2} \). To get the largest purified region, we can set \( \epsilon_{end} = \epsilon \). Since the probability density function \( d(u) \) of the margin \( u(x) = p(y^\pi|x) - p(o|x) \) is bounded by \( c_* \leq d(u) \leq c_* \), we have:

\[
\mathbb{P}_{x \sim D}[y_{final}(x) \neq h^*] \leq \mathbb{P}[p(y^\pi|x) - p(o|x) < \epsilon_{end}]
\]

\[
= \mathbb{P}_{x \sim D}[p(y^\pi|x) - p(o|x) < \epsilon]
\]

\[
\leq c_* \epsilon. \tag{16}
\]

Then \( \mathbb{P}_{x \sim D}[y_{final}(x) = h^*] = 1 - \mathbb{P}_{x \sim D}[y_{final}(x) \neq h^*] \geq 1 - c_* \epsilon \).

The rest of the proof is the total round \( R \geq \frac{6\alpha}{\epsilon} \log \left( \frac{1 - \epsilon_{end}}{1 - \epsilon_{0}} \right) \), which follows from the fact that each round of label flipping improves the the purified region by a factor of \( 1 + \frac{\epsilon}{6\alpha} \):

\[
\left( 1 + \frac{\epsilon}{6\alpha} \right)^R \left( p(y^\pi|x) - \epsilon_{end} \right) \geq p(y^\pi|x) - \epsilon
\]

\[
\Rightarrow \left( 1 + \frac{\epsilon}{6\alpha} \right)^R \geq \frac{p(y^\pi|x) - \epsilon}{p(y^\pi|x) - \epsilon_{end}}
\]

\[
\Rightarrow R \geq \log \left( 1 + \frac{\epsilon}{6\alpha} \right) \geq \log \left( \frac{p(y^\pi|x) - \epsilon}{p(y^\pi|x) - \epsilon_{end}} \right)
\]

\[
\Rightarrow R \geq \frac{6\alpha}{\epsilon} \log \left( \frac{p(y^\pi|x) - \epsilon}{p(y^\pi|x) - \epsilon_{end}} \right) \geq \frac{6\alpha}{\epsilon} \log \left( \frac{1 - \epsilon_{end}}{1 - \epsilon_{0}} \right). \tag{17}
\]

### A.4 Details of Experiments

We collect four widely used benchmark datasets including MNIST [LeCun et al. (1998)], Kuzushiji-MNIST [Clanuwat et al. (2018)], Fashion-MNIST [Xiao et al. (2017)], CIFAR-10 [Krizhevsky & Hinton (2009)], CIFAR-100 [Krizhevsky & Hinton (2009)]. In addition, five real-world PLL datasets are adopted, which are collected from several application domains including Lost [Cour et al. (2011)], Soccer Player [Zeng et al. (2013)] and Yahoo! News [Guillaumin et al. (2010)] for automatic face naming from images or videos, MSRCv2 [Liu & Dietterich (2012)] for object classification, and...
BirdSong \cite{Briggs2012} for bird song classification. Figure \ref{figure:} illustrates the variant integrated with POP performs under different hyper-parameter configurations on Lost.

The average number of candidate labels (avg. #CLs) for each benchmark dataset corrupted by the ID generation process is recorded in Table \ref{table:} and the average number of candidate labels (avg. #CLs) for each real-world PLL dataset is recorded in Table \ref{table:}.