Partial-Label Learning with Conformal Candidate Cleaning

Tobias Fuchs[1] **Florian Kalinke**[1]

[1]Information Systems, Karlsruhe Institute of Technology, Karlsruhe, Germany

Abstract

Real-world data is often ambiguous; for example, human annotation produces instances with multiple conflicting class labels. Partial-label learning (PLL) aims at training a classifier in this challenging setting, where each instance is associated with a set of candidate labels and one correct, but unknown, class label. A multitude of algorithms targeting this setting exists and, to enhance their prediction quality, several extensions that are applicable across a wide range of PLL methods have been introduced. While many of these extensions rely on heuristics, this article proposes a novel enhancing method that incrementally prunes candidate sets using conformal prediction. To work around the missing labeled validation set, which is typically required for conformal prediction, we propose a strategy that alternates between training a PLL classifier to label the validation set, leveraging these predicted class labels for calibration, and pruning candidate labels that are not part of the resulting conformal sets. In this sense, our method alternates between empirical risk minimization and candidate set pruning. We establish that our pruning method preserves the conformal validity with respect to the unknown ground truth. Our extensive experiments on artificial and real-world data show that the proposed approach significantly improves the test set accuracies of several state-of-the-art PLL classifiers.

1 INTRODUCTION

Real-world data is often noisy and ambiguous. In crowdsourcing, for example, different annotators can assign several conflicting class labels to the same instance. Other examples with ambiguous data include web mining (Guillaumin et al., 2010; Zeng et al., 2013) and audio classification (Briggs et al., 2012). While such datasets can be manually cleaned, sanitizing data is costly, especially for large-scale datasets. Partial-label learning (PLL; Jin and Ghahramani 2002; Lv et al. 2020; Xu et al. 2021; Tian et al. 2024) provides a principled way of dealing with such conflicting data. More specifically, in PLL, instances are annotated with sets of candidate labels of which only one unknown label is the correct class label. PLL permits training a multi-class classifier in this weakly-supervised setting.

Many algorithms targeting the PLL problem exist. Recently, several extensions (Bao et al., 2021, 2022; Wang and Zhang, 2022; Zhang et al., 2022b; Xu et al., 2023) that can be combined with a wide range of PLL methods have been proposed, which aim at further improving their predictive performance. Typically, different PLL classifiers perform best on different datasets. In this sense, having extensions that are applicable to a multitude of different PLL algorithms is extremely beneficial. These extensions include feature selection and candidate cleaning techniques, which clean the instance space and candidate label space, respectively. However, many of these extensions depend on heuristics.

In contrast, this article proposes a novel method that alternates between training a PLL classifier through empirical risk minimization and pruning the candidate sets using conformal prediction, which output sets of possible labels that contain the correct label with a specified confidence level (Lei, 2014; Sadinle et al., 2019). In our pruning step, we remove candidate labels if they are not part of these predicted conformal sets. This principled way of reducing the candidate set ambiguity benefits the training of the PLL classifier when compared to the existing heuristic thresholds. Our extension significantly improves the prediction quality of several state-of-the-art PLL methods across a variety of datasets and experimental settings. To guarantee the validity of the conformal classifier used in the pruning step, one usually requires a labeled validation set for the calibration of the coverage guarantee. In the PLL setting, however, ground truth is unavailable. To resolve this seri-

ous issue, we propose a strategy that trains a PLL classifier, uses its predictions to label the validation set, calibrates the conformal sets with the validation set, and prunes candidate labels that are not part of these conformal sets. We show that our method preserves the conformal validity with respect to the unknown ground truth.

Our **contributions** can be summarized as follows.

- *Algorithm.* We propose a novel candidate cleaning method that alternates between training a PLL classifier and pruning the PLL candidate sets. Our algorithm significantly improves the predictive performance of several state-of-the-art PLL approaches.
- *Experiments.* Extensive experiments on artificial and real-world partially labeled data support our claims. An ablation study further demonstrates the usefulness of the proposed strategy. We make our source code and data openly available at `github.com/mathefuchs/pll-with-conformal-candidate-cleaning`.
- *Theoretical analysis.* We analyze our method and show that the pruning step yields valid conformal sets.

Structure of the paper. Section 2 establishes our notations and states the partial-label learning problem, Section 3 discusses related work, Section 4 details our contributions, and Section 5 shows our experimental setup and results. All proofs are deferred to Appendix A. Appendix D lists all hyperparameters used within our experiments in detail and Appendix E contains additional experiments.

2 NOTATIONS

This section establishes notations used throughout our work as well as states the partial-label learning problem.

Given a d-dimensional real-valued feature space $\mathcal{X} = \mathbb{R}^d$ and a set $\mathcal{Y} = [k] := \{1, \dots, k\}$ of $3 \leq k \in \mathbb{N}$ classes, a partially-labeled training dataset $\mathcal{D} = \{(x_i, s_i) \in \mathcal{X} \times 2^{\mathcal{Y}} : i \in [n]\}$ contains n training instances with associated feature vectors $x_i \in \mathcal{X}$ and candidate labels $s_i \subseteq \mathcal{Y}$ for each $i \in [n]$. Their respective ground-truth labels $y_i \in \mathcal{Y}$ are unknown during training, but $y_i \in s_i$. We split the dataset $\mathcal{D}$ into a training set $\mathcal{D}_t$ and a dataset $\mathcal{D}_v$ for calibration.

Let $\Omega = \mathcal{X} \times \mathcal{Y} \times 2^{\mathcal{Y}}$. Underlying partial-label learning (PLL) is the probability triplet $(\Omega, \mathcal{B}(\Omega), \mathbb{P})$ with $\mathcal{B}$ denoting the Borel σ-algebra. We denote by $X : \Omega \to \mathcal{X}$, $Y : \Omega \to \mathcal{Y}$, and $S : \Omega \to 2^{\mathcal{Y}}$ the random variables governing the occurrence of an instance's features, ground-truth label, and its candidate labels, respectively. Their realizations are denoted by x_i, y_i, and s_i. We denote by $\mathbb{P}_X$ the marginal and by $\mathbb{P}_{XY}$ and $\mathbb{P}_{XS}$ the joint distribution of (X, Y) and (X, S), respectively. $\mathbb{P}_{XY}$ coincides with the probability measure usually underlying the supervised setting. We denote with $\mathbb{P}_n := \mathbb{P}_{XS}^n$ the n-fold product of $\mathbb{P}_{XS}$.

The cumulative distribution function of the random variable X is $F_X(t) = \mathbb{P}_X(X \leq t)$ and its empirical counterpart is $\hat{F}_X(t) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{X_i \leq t\}}$, where $X_i, \dots, X_n \overset{i.i.d.}{\sim} \mathbb{P}_X$.

Let $\ell : [0,1]^k \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ denote a measurable loss function, e.g., the log-loss. PLL aims to train a probabilistic classifier $f : \mathcal{X} \to [0,1]^k$ with $\sum_{j=1}^k f_j(x) = 1$, for $x \in \mathcal{X}$, that minimizes the risk $R(f) = \mathbb{E}_{XS}[\sum_{y=1}^k W_{X,S,y}\, \ell(f(X), y)]$, where $W_{X,S,y}$ are label weights to control the influence of different loss terms. $f_y(x)$ denotes the y-th entry of the vector $f(x) \in [0,1]^k$.

Common instantiations for $W_{X,S,y}$ include the average strategy $W_{X,S,y}^{(\text{avg})} = \mathbb{1}_{\{y \in S\}} / |S|$ (Hüllermeier and Beringer, 2005; Cour et al., 2011) and the minimum strategy

$$W_{X,S,y}^{(\min)} = \frac{\mathbb{P}_{Y|X}(Y = y)}{\sum_{j \in S} \mathbb{P}_{Y|X}(Y = j)} \tag{1}$$

(Lv et al., 2020; Feng et al., 2020), which weights the loss based on the relevancy of each label.

For the minimum strategy in (1), the true risk takes the form

$$R(f) = \mathbb{E}_{XS}\big[\sum_{y=1}^k \frac{\mathbb{P}_{Y|X}(Y = y)}{\sum_{j \in S} \mathbb{P}_{Y|X}(Y = j)} \ell(f(X), y)\big]. \tag{2}$$

The empirical version of the risk is obtained by substituting the expectation with a sample mean:

$$\hat{R}(f) = \frac{1}{n} \sum_{i=1}^n \sum_{y=1}^k w_{ij} \ell(f(x_i), y), \tag{3}$$

where $(x_i, s_i) \in \mathcal{D}$ and $w_{ij} \in [0,1]$ approximates the label relevancy $W_{X,S,y}^{(\min)}$ in (1) using

$$w_{ij} = \begin{cases} f_j(x_i) / \sum_{j' \in s_i} f_{j'}(x_i) & \text{if } j \in s_i, \\ 0 & \text{else,} \end{cases} \tag{4}$$

using a trained classifier $f : \mathcal{X} \to [0,1]^k$.

Let $\mathcal{H} = \{f : \mathcal{X} \to [0,1]^k \mid f \text{ measurable}, \forall x \in \mathcal{X} : \sum_{j=1}^k f_j(x) = 1\}$ denote the hypothesis space, $f^* = \arg\min_{f \in \mathcal{H}} R(f)$ the true risk minimizer, and $\hat{f} = \arg\min_{f \in \mathcal{H}} \hat{R}(f)$ the empirical risk minimizer. An optimal multi-class classifier must be of the form $f_y^*(x) = \mathbb{P}_{Y|X=x}(Y = y)$ (Yu et al., 2018, Lemma 1). We make the common assumption that the hypothesis space $\mathcal{H}$ is well-specified, that is, $f^* \in \mathcal{H}$ (Tsybakov, 2004; van Erven et al., 2015). The class label of each instance $x \in \mathcal{X}$ with the highest probabilistic prediction, that is, $\hat{y}_x = \arg\max_{y \in \mathcal{Y}} \hat{f}_y(x)$, is called *pseudo-label*.

3 EXISTING WORK

Partial-label learning is one out of many weakly-supervised learning frameworks (Bylander, 1994; Hady and Schwenker,

2013; Ishida et al., 2019), where training instances are annotated with multiple candidate labels. Section 3.1 discusses related work regarding partial-label learning and Section 3.2 discusses related work regarding set-valued prediction-making, which is a natural fit for representing the ambiguity of the PLL candidate sets.

3.1 PARTIAL-LABEL LEARNING (PLL)

PLL is a weakly-supervised learning problem that has gained significant attention over the last decades. Most approaches adapt common supervised classification algorithms to the PLL setting. Examples include a logistic regression formulation (Grandvalet, 2002), expectation-maximization strategies (Jin and Ghahramani, 2002; Liu and Dietterich, 2012), nearest-neighbor methods (Hüllermeier and Beringer, 2005; Zhang and Yu, 2015; Fuchs et al., 2025), support-vector classifiers (Nguyen and Caruana, 2008; Cour et al., 2011; Yu and Zhang, 2017), custom stacking and boosting ensembles (Zhang et al., 2017; Tang and Zhang, 2017; Wu and Zhang, 2018), and label propagation strategies (Zhang and Yu, 2015; Zhang et al., 2016; Xu et al., 2019; Wang et al., 2019; Feng and An, 2019).

Recent state-of-the-art methods (Lv et al., 2020; Feng et al., 2020; Xu et al., 2021; Zhang et al., 2022a; Wang et al., 2022; Xu et al., 2023; Tian et al., 2024) minimize variations of (3) with the weights as in (4) using different deep learning approaches. The minimum loss reweighs the loss terms to only include the most likely class labels. Gong et al. (2024) extend this idea by introducing a smoothing component.

Lv et al. (2020); Feng et al. (2020) iteratively refine the PLL candidate sets by alternating between training a model $f : \mathcal{X} \to [0,1]^k$ using empirical risk minimization on (3) and updating the label weights w_{ij} in (4) using the trained classifier f. At the beginning, the weights w_{ij} are initialized with uniform weights on the respective candidate sets: $w_{ij} = 1/|s_i|$ if $j \in s_i$, else 0, which coincides with the average strategy (Hüllermeier and Beringer, 2005; Cour et al., 2011). They further show that the resulting classifier is risk consistent with the Bayes classifier f^*, if the small-ambiguity-degree condition holds (Cour et al., 2011; Liu and Dietterich, 2012). The condition requires that there is no incorrect label $\bar{y} \neq y$, which co-occurs with the correct label y in a candidate set with a probability of one. Formally, one imposes that $\sup_{x\in\mathcal{X},y\in\mathcal{Y},\bar{y}\in\mathcal{Y},\bar{y}\neq y} \mathbb{P}_{S|X=x,Y=y}(\bar{y} \in S) < 1$.

Because of the huge variety of PLL methods, there are recent algorithms that can be combined with any of the above to improve prediction performance further. Wang and Zhang (2022) propose a feature augmentation technique based on class prototypes and Bao et al. (2021, 2022); Zhang et al. (2022b) propose feature selection strategies for PLL data. Existing state-of-the-art methods achieve significantly better accuracies when trained on these modified feature sets.

Xu et al. (2023) propose the method POP, which gradually removes unlikely class labels from the candidate sets if the margin between the most likely and the second-most likely class label exceeds some heuristic threshold. In contrast, our method gradually removes unlikely class labels based on the set-valued conformal prediction framework, which provides a more principled way of cleaning the candidate sets. Our method significantly improves the test set accuracies of several state-of-the-art methods including the method POP.

3.2 SET-VALUED PREDICTIONS

Recent methods in supervised multi-class classification (Lei, 2014; Barber et al., 2023; Mozannar et al., 2023; Mao et al., 2024; Narasimhan et al., 2024) explore training set-valued predictors $C : \mathcal{X} \to 2^{\mathcal{Y}}$ rather than single-label classifiers $f : \mathcal{X} \to \mathcal{Y}$ as they offer more flexibility in representing the uncertainty involved in prediction-making. Set-valued prediction-making involves a variety of problem formulations including reject options and conformal prediction. Reject options allow one to abstain from individual predictions if unsure alleviating the cost of misclassifications; see Fuchs et al. (2025) for a recent study of reject options in PLL.

In conformal prediction, classifiers output sets of class labels $C(x) \subseteq \mathcal{Y}$. *Valid* conformal predictors guarantee that

$$\mathbb{P}_{XY}(Y \in C(X)) \geq 1 - \alpha, \tag{5}$$

which means that the correct label is part of a conformal set with a given error level of at most $\alpha \in (0,1)$. The conformal predictor C that outputs $C(x) = \mathcal{Y}$, for $x \in \mathcal{X}$, is trivially valid as it covers the correct label with a probability of one. To avoid this case, one searches for conformal predictors C with minimal expected cardinality $\mathbb{E}_X\,|C(X)|$, while still being valid. In the supervised setting, this is captured by the following optimization problem (Sadinle et al., 2019):

$$\begin{aligned} \min_{C:\mathcal{X}\to 2^{\mathcal{Y}}} \quad & \mathbb{E}_X\,|C(X)|, \\ \text{subject to} \quad & \mathbb{P}_{XY}(Y \in C(X)) \geq 1 - \alpha. \end{aligned} \tag{6}$$

Optimal solutions to (6) are of the form $C(x) = \{y \in \mathcal{Y} : \mathbb{P}_{Y|X=x}(Y = y) \geq t_\alpha\}$, for $x \in \mathcal{X}$, where t_α is set to

$$\begin{aligned} t_\alpha = \sup\big\{t \in [0,1] : \mathbb{P}_{XY}\big[(x,y) : \\ \mathbb{P}_{Y|X=x}(Y = y) \geq t\big] \geq 1 - \alpha\big\}, \end{aligned} \tag{7}$$

where we assume that the quantile function of $\mathbb{P}_{Y|X=x}(Y = y)$ is continuous at t_α.[1] In practice, one approximates t_α by computing the empirical distribution function on a hold-out validation set. One splits the dataset $\mathcal{D}$ into a dataset $\mathcal{D}_t$ for model training and $\mathcal{D}_v$ for calibrating the conformal predictor C with respect to the confidence level α. The validation set $\mathcal{D}_v$ is assumed to be exchangeable with respect to the joint distribution $\mathbb{P}_{XY}$.

[1] See Sadinle et al. (2019, Theorem 1) for the general case.

Conformal prediction is also a natural fit to partial-label learning as both deal with sets of class labels. Javanmardi et al. (2023) examine different ways of achieving valid conformal sets in the PLL context. However, they do not propose any new PLL method against which we can compare. Rather, they analyze the properties of different nonconformity measures in this context. In contrast, our focus is on constructing new PLL methods and evaluating them.

In the following section, we propose a novel candidate cleaning method that is based on conformal prediction and adapts (6) to the PLL setting to yield valid conformal sets. The optimization problem (6) cannot directly be transferred to the PLL context as ground truth for the calibration of the validity property is unavailable. We propose a strategy that uses the PLL classifier f to label the validation set and then leverages these pseudo-labels for calibration. We show that this preserves the validity with respect to the ground truth.

4 PLL WITH CONFORMAL CLEANING

We propose a novel candidate cleaning strategy that iteratively cleans the candidate sets of the PLL dataset $\mathcal{D}$ by reducing the candidate set cardinalities. Our method alternates between training a PLL classifier through empirical risk minimization and pruning the candidate sets based on conformal prediction. Conformal predictors $C : \mathcal{X} \to 2^{\mathcal{Y}}$ cover the correct label y_i of instance x_i with a specified probability; see (5). This coverage property is calibrated using a separate validation set of exchangeable PLL data points that are labeled using the trained PLL algorithm. As the classifier can give wrong predictions, however, we propose a novel correction strategy that accounts for possible misclassifications when calibrating the coverage of the correct labels against the validation set, which maintains the validity guarantee. We remove class labels from the candidate sets s_i if they are not part of the predicted conformal set $C(x_i)$ since the correct label y_i is in $C(x_i)$ with a specified confidence level.

This procedure iteratively removes noise from incorrect candidate labels, which benefits the training of the PLL classifier by having to account for less and less noise in each training step. Many PLL algorithms (Lv et al., 2020; Xu et al., 2023; Tian et al., 2024) proceed in a similar manner. They have in common that they alternate between training a PLL classifier and using its predictions to refine the candidate label weights. This can equivalently be expressed from an expectation-maximization perspective (Wang et al., 2022, Section 5). These label propagation strategies are state-of-the-art in many weakly-supervised learning domains. In contrast to the existing heuristic update rules, however, our proposed method provides a principled way of iteratively cleaning the candidate sets using conformal predictors C.

In the following sections, we discuss our method in detail. Section 4.1 elaborates on the notion of conformal validity in the PLL context, Section 4.2 details how to correct for the ambiguity in PLL compared to the supervised setting, Section 4.3 outlines the proposed algorithm, Section 4.4 discusses the method's runtime complexity, and Section 4.5 discusses the placement of our method with respect to related work.

4.1 PLL VALIDITY

Since we use the conformal predictions $C(x_i)$ to clean the associated candidate sets s_i, for $(x_i, s_i) \in \mathcal{D}$, we require that $s_i \cap C(x_i)$ is nonempty with a specified confidence level as otherwise $C(x_i)$ does not contain the unknown correct label y_i. Hence, we adapt (5) to our setting and consider a conformal classifier C valid with respect to the PLL candidate sets if it holds that

$$\mathbb{P}_{XS}(S \cap C(X) \neq \emptyset) \geq 1 - \alpha, \tag{8}$$

for a given error level $\alpha \in (0, 1)$. In other words, conformal predictions $C(x_i)$ need to cover the observed ambiguously labeled candidate sets s_i with a specified probability. Recall that $C(x) = \mathcal{Y}$, for $x \in \mathcal{X}$, trivially satisfies (8). One therefore also wants to minimize the cardinalities $|C(x)|$. Given the standard PLL assumption that the correct label y_i is within the respective candidate set s_i, which implies that $\mathbb{P}_{S|X=x,Y=y}(y \in S) = 1$ for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, an optimal solution to (6) is also valid in the sense of (8). Theorem 4.1 captures this relationship and underpins our proposed cleaning method, which we detail in Section 4.3.

Theorem 4.1. *Assume that* $\mathbb{P}_{S|X=x,Y=y}(y \in S) = 1$, *for any* $(x, y) \in \mathcal{X} \times \mathcal{Y}$, *and* $\alpha \in (0, 1)$. *Then, an optimal solution* C *of* (6) *satisfies* (8)*:* $\mathbb{P}_{XS}(S \cap C(X) \neq \emptyset) \geq 1 - \alpha$.

4.2 CORRECTING FOR MISCLASSIFICATION

Recall that, in the PLL setting, the ground-truth labels y are unavailable during training, which hinders the approximation of (7) needed for the solution of (6). Because a solution to (6) is, however, also desirable in the PLL setting (Theorem 4.1), we make use of existing PLL algorithms to generate pseudo-labels. This strategy iteratively learns a prediction model $f : \mathcal{X} \to [0, 1]^k$ that minimizes the empirical risk in (3). We use the trained model f to predict the labels on the validation set $\mathcal{D}_v$, which in turn is used for the calibration of the validity guarantee. Notably, this strategy results in a valid conformal predictor (Theorem 4.4). We note that it remains open to establish the minimality of the resulting conformal sets (analogous to solutions of (6)).

At first glance, it might be counter-intuitive to use the trained model f to label the validation set and build conformal sets based on it. However, we want to recall that the used base PLL classifier is risk consistent (Feng et al., 2020, Theorem 4). With this result and additional mild assumptions,

we can prove that the PLL classifier's predictions cannot be arbitrarily bad (Lemma 4.3) and, leveraging this, that our conformal predictor is valid for some adapted threshold and error level (Theorem 4.4).

One of our central assumptions is a Bernstein condition (Audibert, 2004; Bartlett and Mendelson, 2006; Grünwald and Mehta, 2020) on the loss difference (Assumption 4.2). The Bernstein condition is defined as follows.

Assumption 4.2 (Bernstein Condition)**.** Let $B > 0$, $\beta \in (0, 1]$, $f^* = \arg\min_{f \in \mathcal{H}} R(f)$ the true risk minimizer, and $\ell : [0, 1]^k \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ a loss function. We assume that the excess loss $L_f(x, y) := \ell(f(x), y) - \ell(f^*(x), y)$ satisfies the (β, B)-Bernstein condition, that is, for $f \in \mathcal{H}$,

$$\mathbb{E}_{XY}\left[L_f(X, Y)^2\right] \leq B \left(\mathbb{E}_{XY}[L_f(X, Y)]\right)^{\beta}.$$

Assumption 4.2 is frequently made in ERM as it allows controlling the variance of the resulting losses, since $\mathrm{Var}_{XY}[\ell(f(X), Y) - \ell(f^*(X), Y)] \leq \mathbb{E}_{XY}[(\ell(f(X), Y) - \ell(f^*(X), Y))^2]$. In other words, the tail of the distribution of the excess loss must be well-behaved.

Building upon Assumption 4.2, we prove the results in the following Lemma 4.3, which are the main building blocks underlying the proof of our main result.

Lemma 4.3. *Let $\hat{f} = \arg\min_{f \in \mathcal{H}} \hat{R}(f)$ the empirical risk minimizer, $f^* = \arg\min_{f \in \mathcal{H}} R(f)$ the true risk minimizer, $\hat{y}_x = \arg\max_y \hat{f}_y(x)$, $y^*_x = \arg\max_y f^*_y(x)$, and Assumption 4.2 hold for the excess loss $L_{\hat{f}}$.*

(i) Then, for any $\delta_1 \in (0, 1)$ and some constant $M_1 > 0$, it holds, with $\mathbb{P}_n$-probability at least $1 - \delta_1$, that

$$\mathbb{E}_{XY}\left[\left|\hat{f}_Y(X) - f^*_Y(X)\right|\right] \leq M_1 \left(\frac{\log(1/\delta_1)}{n}\right)^{\frac{1}{4}\beta},$$

assuming that $\ell : (0, 1]^k \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$, $(p, y) \mapsto -\log p_y$ is the log-loss.

(ii) Also, for any $\delta_2 \in (0, 1)$ and some constant $M_2 > 0$, it holds, with $\mathbb{P}_n$-probability at least $1 - \delta_2$, that

$$\mathbb{P}_X[\hat{y}_X \neq y^*_X] \leq M_2 \left(\frac{\log(1/\delta_2)}{n}\right)^{\frac{1}{2}\beta},$$

given that, for any $x \in \mathcal{X}$ and some constant $\delta_5 \in [0, 1)$, $\mathbb{P}_{Y|X=x}(Y \in \{\hat{y}_x, y^*_x\}) \geq 1 - \delta_5$.

Intuitively, Lemma 4.3 (i) and (ii) state that, under mild assumptions, a consistent PLL classifier cannot, in expectation, provide arbitrarily bad predictions. More precisely, Lemma 4.3 (i) states that the expected absolute difference in the probabilistic predictions of the empirical and true risk minimizer are upper-bounded. Lemma 4.3 (ii) states that the probability of class label predictions of the empirical and true risk minimizer not matching is upper-bounded. Note that, for $n \to \infty$, both upper-bounds tend to zero.

In the following, we comment on the assumptions made. Lemma 4.3 (i) requires the loss function to be the log-loss as it is a local proper loss function (Gneiting and Raftery, 2007), that is, ℓ is a proper loss function that only uses the y-th entry of the vector p in the computation of $\ell(p, y)$, which we use in our proof. Lemma 4.3 (ii) requires that the correct label y^*_x and pseudo-label $\hat{y}_x$ have some lower-bound for their conditional probability mass. Intuitively, the assumption captures that the true class posterior of the correct label y^*_x must have a probability mass that is not arbitrarily close to zero.

Based on the upper bounds in Lemma 4.3, one can adapt the threshold and confidence levels in (6) and (7) such that the conformal guarantee is still valid when using the pseudo-labels on the validation set. Theorem 4.4 states this result.

Theorem 4.4. *Assume the setting of Lemma 4.3 (i) and (ii) and, for any $\delta_6 \in (0, 1)$, $\mathbb{P}_{Y|X=x}(Y = y^*_X) \geq 1 - \delta_6$ with $y^*_X = \arg\max_{y' \in \mathcal{Y}} f^*_y(X)$. For any $\alpha \in (0, 1)$, let*

$$t_\alpha = \sup\{t \in [0, 1] \mid \hat{F}_{\hat{f}_{\hat{y}_X}(X)}(t) \leq \alpha\}, \tag{9}$$

with $\hat{y}_x = \arg\max_{y \in \mathcal{Y}} \hat{f}_y(x)$. Then, the conformal set

$$C(x) = \{y \in \mathcal{Y} \mid \hat{f}_y(x) \geq t_\alpha - \delta_3\} \tag{10}$$

is valid, that is, $\mathbb{P}_X(y^*_X \in C(X)) \geq 1 - \alpha'_n$ holds with a $\mathbb{P}_n$-probability of at least $1 - (\delta_1 + \delta_2 + \delta_4)$, where, for any $\delta_1, \delta_2, \delta_4 \in (0, 1)$ and some constants $\beta \in (0, 1]$, $B, \delta_3, M_1, M_2 > 0$,

$$\alpha'_n := \frac{1}{\delta_3(1 - \delta_6)} M_1 \left(\frac{\log(1/\delta_1)}{n}\right)^{\frac{1}{4}\beta} + M_2 \left(\frac{\log(1/\delta_2)}{n}\right)^{\frac{1}{2}\beta} + \alpha + \left(\frac{\log(2/\delta_4)}{2n}\right)^{\frac{1}{2}}.$$

Intuitively, the tighter the upper bounds in Lemma 4.3 are, the smaller the necessary correction of the threshold and confidence level in Theorem 4.4. In other words, loose upper bounds in Lemma 4.3 lead to high cardinalities of $C(x)$ in (10). In contrast, tight upper bounds in Lemma 4.3 lead to small cardinalities of $C(x)$ in (10). The following Remark 4.5 details how to obtain conformal validity for a fixed error level.

Remark 4.5. Alternatively, one obtains a fixed error level $\alpha_2 \in (0, 1)$ in Theorem 4.4, that is, $\mathbb{P}_X(y^*_X \in C(X)) \geq 1 - \alpha_2$, by using

$$\alpha'' = \alpha_2 - \frac{1}{\delta_3(1 - \delta_6)} M_1 \left(\frac{\log(1/\delta_1)}{n}\right)^{\frac{1}{4}\beta} - M_2 \left(\frac{\log(1/\delta_2)}{n}\right)^{\frac{1}{2}\beta} - \left(\frac{\log(2/\delta_4)}{2n}\right)^{\frac{1}{2}}, \tag{11}$$

in the computation of the threshold $t_{\alpha''}$ in (9).

While Remark 4.5 follows from a simple substitution, it explicitly links Theorem 4.4 to the setting usually considered in conformal prediction: One wants to have a conformal predictor that is valid regarding some specified confidence level α_2, which Remark 4.5 achieves by using an altered α'' in the computation of $t_{\alpha''}$. If $\alpha'' \leq 0$, the resulting conformal predictor defaults to $C(x) = \mathcal{Y}$, for $x \in \mathcal{X}$, which is valid. In contrast, given some confidence level α, Theorem 4.4 gives a conformal predictor that is valid with the confidence level $\alpha'_n \neq \alpha$.

Theorem 4.4 enables our proposed algorithm. When using a consistent PLL classifier to label the validation set, a conformal predictor with a threshold set based on these pseudo-labels still satisfies a conformal validity guarantee for an adapted threshold and error level. The subsequent section discusses our approach.

4.3 PROPOSED ALGORITHM

Based on the conformal predictor in Theorem 4.4, we propose a novel candidate cleaning strategy that alternates between training a neural-network-based PLL classifier and pruning the candidate labels by conformal prediction. We outline our method in Algorithm 1. In the following, we provide an overview. Thereafter, we discuss all parts in detail.

First, we randomly partition the dataset $\mathcal{D}$ into $\mathcal{D}_t$ for training the model and $\mathcal{D}_v$ for calibrating the conformal predictor C based on the current state of the prediction model f (Line 1). The training set consists of $80\,\%$ and the validation set of $20\,\%$ of all instances. We initialize the model f and the label weights w_{ij} in Lines 3–4. Lines 5–23 contain the main training loop, which can be divided into four phases: (1) Updating the predictions on the validation set $\mathcal{D}_v$ for calibration (Lines 6–7), (2) updating the model's weights θ through back-propagation (Lines 8–10), (3) cleaning the candidate sets s_i based on the predicted conformal sets $C(x_i)$ (Lines 11–20), and (4) updating the label weights w_{ij} (Lines 21–22). We detail these phases in the following.

In **phase 1** (Lines 6–7), we use the current model f to predict the labels on the hold-out validation set $\mathcal{D}_v$, which are required for the computations in phase 3.

In **phase 2** (Lines 8–10), we update the weights θ of the neural network f by performing back-propagation on the risk term (3). As our candidate cleaning method is agnostic to the concrete PLL classifier used, one can also use other commonly-used PLL strategies instead.

In **phase 3** (Lines 11–18), we compute the conformal predictor C, which is used to clean the candidate sets. After completing R_{warmup} warm-up epochs, we start with our pruning procedure. In Line 13, we compute α_r for the current epoch r. While it is desirable to use the exact value of α'' in (11) in Line 13 of Algorithm 1, its computation is unfortunately infeasible as the constants B and β, for which the Bernstein condition (Assumption 4.2) holds, cannot be known unless the true distribution $\mathbb{P}_{XY}$ is known. As the employed PLL classifiers are consistent, that is, they converge to the Bayes classifier with enough samples, we approximate the estimation error terms in (11) by the probability mass that the PLL classifier allocates on false class labels, that is, class labels that are not part of the candidate sets and hence cannot be the correct label. Given $(x_i, s_i) \in \mathcal{D}_t$, we set $\alpha_r = \frac{1}{n'} \sum_{i=1}^{n'} \sum_{j \notin s_i} f_j(x_i)$ with $n' = |\,\mathcal{D}_t\,|$. Then, we compute the conformal prediction sets $C(x_i)$ in Line 15 for all training instances $(x_i, s_i) \in \mathcal{D}_t$ using the empirical distribution function of the adapted scores on the validation set; this conformal predictor C is valid by Theorem 4.4. We use the conformal sets $C(x_i)$ to prune the candidate sets s_i. If $C(x_i)$ and s_i have a nonempty intersection, which is implied with high probability by the conformal validity (Theorem 4.1), we assign $s_i \cap C(x_i)$ to s_i in Line 17.

Finally, in **phase 4** (Lines 21–22), we update the label weights w_{ij} based on the cleaned candidate sets s_i with (4).

Algorithm 1 Conformal Candidate Cleaning

Input: PLL dataset $\mathcal{D} = \{(x_i, s_i) \in \mathcal{X} \times 2^{\mathcal{Y}} : i \in [n]\}$; conformal error level $\alpha \in (0, 1)$; number of epochs R; number of warm-up rounds R_{warmup};
Output: Predictor $f : \mathcal{X} \to [0, 1]^k$, $\sum_{y \in \mathcal{Y}} f_y(x) = 1$;

1: $(\mathcal{D}_t, \mathcal{D}_v) \leftarrow$ Partition $\mathcal{D}$ into $\mathcal{D}_t$ for model training and $\mathcal{D}_v$ for calibrating the conformal sets;
2: $n' \leftarrow |\,\mathcal{D}_t\,|$;
3: $(f, \theta) \leftarrow$ Initialize model f and its weights θ;
4: $(w_{ij})_{i \in [n'], j \in [k]} \leftarrow 1/|s_i|$ if $j \in s_i$, else 0;
5: **for** $r = 1, \ldots, R$ **do**
6: ▷ *Update predictions on the validation set*
7: $\mathcal{S} \leftarrow \{\max_{y \in s_i} f_y(x_i; \theta) : (x_i, s_i) \in \mathcal{D}_v\}$;
8: ▷ *Update f's weights θ*
9: $\hat{R}(f; w, \theta) \leftarrow -\frac{1}{n'} \sum_{i=1}^{n'} \sum_{j=1}^{k} w_{ij} \log f_j(x_i; \theta)$;
10: Update θ by back-propagation on $-\nabla \hat{R}(f; w, \theta)$;
11: ▷ *Clean candidate sets s_i*
12: **if** $r \geq R_{\text{warmup}}$ **then**
13: $\alpha_r \leftarrow$ Estimate the adapted error level in (11);
14: **for** $(x_i, s_i) \in \mathcal{D}_t$ **do**
15: $C(x_i) \leftarrow$ Construct the conformal predictor as defined in (10) using $\mathcal{S}$ and α_r;
16: **if** $s_i \cap C(x_i) \neq \emptyset$ **then**
17: $s_i \leftarrow s_i \cap C(x_i)$;
18: ▷ *Update label weights w_{ij}*
19: $(w_{ij})_{i \in [n'], j \in [k]} \leftarrow \frac{f_j(x_i)}{\sum_{j' \in s_i} f_{j'}(x_i)}$ if $j \in s_i$, else 0;
20: **return** predictor $f(\,\cdot\,; \theta)$;

Table 1: Average test-set accuracies (± std.) on the real-world datasets (top) and the supervised datasets with added incorrect candidate labels (bottom). We benchmark our strategy (CONF+) combined with all existing methods.

Method	*bird-song*	*lost*	*mir-flickr*	*msrc-v2*	*soccer*	*yahoo-news*
PRODEN (2020)	75.55 (± 1.08)	78.94 (± 3.01)	**67.05** (± 1.18)	54.33 (± 1.76)	54.18 (± 0.55)	65.25 (± 1.00)
CONF+P. (no)	76.27 (± 0.94)	79.56 (± 1.96)	66.07 (± 1.63)	53.00 (± 2.24)	54.63 (± 0.81)	65.42 (± 0.36)
CONF+PRODEN	**76.99** (± 0.90)	**80.09** (± 4.40)	66.91 (± 1.57)	**54.60** (± 3.42)	**54.77** (± 0.84)	**65.93** (± 0.42)
CC (2020)	74.49 (± 1.57)	78.23 (± 2.11)	62.39 (± 1.87)	50.96 (± 2.03)	55.28 (± 0.96)	**65.03** (± 0.51)
CONF+CC	**75.01** (± 1.84)	**79.38** (± 1.79)	**63.37** (± 0.45)	**52.45** (± 3.64)	**55.52** (± 0.74)	64.35 (± 0.64)
VALEN (2021)	**72.30** (± 1.83)	**70.18** (± 3.44)	**67.05** (± 1.48)	**49.20** (± 1.37)	**53.20** (± 0.88)	**62.25** (± 0.45)
CONF+VALEN	71.22 (± 1.03)	68.41 (± 2.95)	61.61 (± 2.79)	48.37 (± 2.24)	52.49 (± 1.00)	62.16 (± 0.74)
CAVL (2022)	69.78 (± 3.00)	**72.12** (± 1.08)	**65.02** (± 1.34)	**52.67** (± 2.32)	**55.06** (± 0.48)	61.91 (± 0.46)
CONF+CAVL	**72.00** (± 1.22)	71.24 (± 3.81)	64.42 (± 0.89)	51.63 (± 5.03)	54.85 (± 0.92)	**62.43** (± 0.43)
POP (2023)	75.17 (± 1.04)	77.79 (± 2.11)	**67.93** (± 1.44)	53.83 (± 0.69)	55.31 (± 0.71)	65.09 (± 0.64)
CONF+POP	**77.58** (± 1.01)	**78.41** (± 2.13)	66.21 (± 2.19)	**54.82** (± 3.60)	**56.49** (± 1.10)	**65.25** (± 0.23)
CROSEL (2024)	75.11 (± 1.79)	**81.24** (± 3.68)	**67.58** (± 1.16)	52.23 (± 2.83)	52.64 (± 1.21)	**67.72** (± 0.32)
CONF+CROSEL	**77.76** (± 0.50)	81.15 (± 2.57)	65.93 (± 1.94)	**54.10** (± 2.75)	**54.97** (± 0.65)	67.55 (± 0.22)

Method	*mnist*	*fmnist*	*kmnist*	*svhn*	*cifar10*	*cifar100*
PRODEN (2020)	87.21 (± 0.83)	71.18 (± 2.95)	59.31 (± 1.22)	83.71 (± 0.37)	**86.42** (± 0.39)	**61.58** (± 0.20)
CONF+P. (no)	**91.74** (± 0.34)	**78.38** (± 0.50)	**66.88** (± 0.76)	**87.31** (± 0.30)	85.39 (± 0.49)	61.50 (± 0.20)
CONF+PRODEN	91.55 (± 0.23)	78.09 (± 0.33)	66.43 (± 0.38)	86.99 (± 0.41)	85.29 (± 0.44)	61.45 (± 0.49)
CC (2020)	**86.29** (± 2.18)	**66.19** (± 2.77)	**58.29** (± 0.32)	83.40 (± 0.42)	**85.61** (± 0.27)	60.43 (± 0.53)
CONF+CC	85.20 (± 4.16)	59.75 (± 2.68)	57.07 (± 0.66)	**84.32** (± 0.31)	84.10 (± 0.38)	**60.49** (± 0.37)
VALEN (2021)	**78.91** (± 0.80)	66.53 (± 2.65)	58.48 (± 0.45)	54.87 (± 15.83)	**84.83** (± 0.23)	58.67 (± 0.17)
CONF+VALEN	74.20 (± 21.99)	**69.09** (± 2.71)	**60.95** (± 2.59)	**78.31** (± 3.15)	84.35 (± 0.22)	**59.57** (± 0.71)
CAVL (2022)	71.11 (± 3.92)	**59.85** (± 6.49)	48.15 (± 5.07)	**72.57** (± 3.14)	**84.00** (± 0.94)	**61.97** (± 0.25)
CONF+CAVL	**71.86** (± 4.57)	59.54 (± 6.62)	**52.14** (± 3.89)	70.53 (± 2.94)	82.82 (± 1.58)	61.79 (± 0.36)
POP (2023)	87.08 (± 0.58)	72.30 (± 2.63)	60.63 (± 1.15)	83.69 (± 0.28)	**86.76** (± 0.29)	61.27 (± 0.60)
CONF+POP	**91.19** (± 0.29)	**79.15** (± 1.23)	**67.37** (± 0.28)	**85.89** (± 0.48)	85.32 (± 0.38)	**61.38** (± 0.30)
CROSEL (2024)	91.84 (± 0.44)	76.34 (± 1.21)	**65.55** (± 0.81)	75.95 (± 3.91)	**87.32** (± 0.22)	63.69 (± 0.29)
CONF+CROSEL	**91.85** (± 0.61)	**77.31** (± 0.46)	64.73 (± 1.52)	**77.70** (± 3.84)	87.05 (± 0.09)	**64.55** (± 0.31)

4.4 RUNTIME COMPLEXITY

The main runtime cost of our cleaning method arises from the computation of the conformal sets $C(x_i)$ in Line 15 of Algorithm 1. Finding the rank of $f_y(x_i)$ within $\mathcal{S}$ can be done by first sorting $\mathcal{S}$ and then using a binary search. This requires a total runtime of $\mathcal{O}(Rn \log n)$, as we prune candidate labels in each epoch and, both, the training set $\mathcal{D}_t$ and validation set $\mathcal{D}_v$ have a size of $\mathcal{O}(n)$. Note that the runtime of our method is not dependent on the number of feature dimensions d as the considered scores $\mathcal{S}$ are scalars.

4.5 PLACEMENT REGARDING RELATED WORK

In this section, we provide a brief comparison of our cleaning strategy with the one employed by POP (Xu et al., 2023). POP uses level sets, which we sketch in the following. Let $e > 0$, $(x_i, s_i) \in \mathcal{D}$, the predicted label $\hat{y}_{x_i} = \arg\max_{j \in s_i} f_j(x_i)$, and the second-most likely label $\hat{o}_{x_i} = \arg\max_{j \in s_i, j \neq \hat{y}} f_j(x_i)$. The level sets are of the form $L(e) = \{x \in \mathcal{X} : f_{\hat{y}_x}(x) - f_{\hat{o}_x}(x) \geq e\}$ to gradually clean the candidate labels for instances in $L(e)$. In other words, one is confident in the predicted labels if the distance between the most likely and second-most likely label exceeds some margin. Given $x \in L(e)$, this implies

$$\begin{aligned} & f_{\hat{y}_x}(x) - f_{\hat{o}_x}(x) \geq e \\ \overset{(\dagger)}{\Leftrightarrow} \quad & 2f_{\hat{y}_x}(x) - 1 + \underbrace{\sum_{j' \in \mathcal{Y} \setminus \{\hat{y}_x, \hat{o}_x\}} f_{j'}(x)}_{\leq 1} \geq e \\ \Rightarrow \quad & f_{\hat{y}_x}(x) \geq \frac{1}{2}e, \end{aligned} \tag{12}$$

with (†) holding as $f_{\hat{o}_x}(x) = 1 - \sum_{j' \in \mathcal{Y} \setminus \{\hat{y}_x, \hat{o}_x\}} f_{j'}(x) - f_{\hat{y}_x}(x)$. POP gradually decreases the value of e to enlarge the reliable region $L(e)$, which in turn requires $f_{\hat{y}_x}(x) \geq \frac{1}{2}e$ by (12). In contrast, in Theorem 4.4, we find an appropriate value t such that $f_{\hat{y}_x}(x) \geq t$ holds with a specified probability. The conformal predictor C can therefore also be interpreted as a level set. However, our approach satisfies the conformal validity guarantee.

5 EXPERIMENTS

Section 5.1 lists all PLL methods that we compare against, Section 5.2 summarizes the experimental setup, and Section 5.3 shows our results.

5.1 ALGORITHMS FOR COMPARISON

In our experiments, we benchmark six state-of-the-art PLL methods. These are PRODEN (Lv et al., 2020), CC (Feng et al., 2020), VALEN (Xu et al., 2021), CAVL (Zhang et al., 2022a), POP (Xu et al., 2023), and CROSEL (Tian et al., 2024). For each dataset, we use the same base models across all approaches. For the colored-image datasets, we use a ResNet-9 architecture (He et al., 2016). Else, we use a standard d-300-300-300-k MLP (Werbos, 1974). We train all models from scratch. An in-depth overview of all hyperparameters is in Appendix D. Appendix E contains additional experiments, including the use of the pre-trained BLIP-2 model (Li et al., 2023) on the vision datasets.

5.2 EXPERIMENTAL SETUP

Data. Using the standard PLL experimentation protocol (Lv et al., 2020; Zhang et al., 2022a; Tian et al., 2024), we perform experiments on real-world PLL datasets and on supervised datasets with artificially added incorrect candidate labels. To report averages and standard deviations, we repeat all experiments five times with different seeds. For the supervised multi-class datasets, we use *mnist* (LeCun et al., 1999), *fmnist* (Xiao et al., 2018), *kmnist* (Clanuwat et al., 2018), *cifar10* (Krizhevsky, 2009), *cifar100* (Krizhevsky, 2009), and *svhn* (Netzer et al., 2011). Regarding the real-world PLL datasets, we use *bird-song* (Briggs et al., 2012), *lost* (Cour et al., 2011), *mir-flickr* (Huiskes and Lew, 2008), *msrc-v2* (Liu and Dietterich, 2012), *soccer* (Zeng et al., 2013), and *yahoo-news* (Guillaumin et al., 2010). An overview of the dataset characteristics is in Appendix D.

Candidate generation. As is common in related work, we use two kinds of candidate label generation methods to augment labeled multi-class data with partial labels: Uniform (Hüllermeier and Beringer, 2005; Liu and Dietterich, 2012) and instance-dependent (Xu et al., 2021). For *cifar10* and *cifar100*, we use the uniform generation strategy as in Wang et al. (2022) and the instance-dependent strategy for all other datasets. For adding instance-dependent candidate labels, we first train a supervised MLP classifier $g : \mathcal{X} \to [0, 1]^k$. Then, given an instance $x \in \mathcal{X}$ with correct label $y \in \mathcal{Y}$, we add the incorrect label $\bar{y} \in \mathcal{Y} \setminus \{y\}$ to the candidate set s with a binomial flipping probability of $\xi_{\bar{y}}(x) = g_{\bar{y}}(x) / \max_{y' \in \mathcal{Y} \setminus \{y\}} g_{y'}(x)$. For *cifar10*, we use a constant flipping probability of $\xi_{\bar{y}}(x) = 0.1$. In the *cifar100* dataset, all class labels $\mathcal{Y}$ are partitioned into 20 meta-categories (for example, aquatic mammals consisting of the labels beaver, dolphin, otter, seal, and whale) and we use a constant flipping probability of $\xi_{\bar{y}}(x) = 0.1$ if $\bar{y}$ and y belong to the same meta-category, else we set $\xi_{\bar{y}}(x) = 0$.

Table 2: Number of significant differences compared to all 6 methods on all 12 datasets using a paired t-test (level 5 %).

Comparison vs. all others	Wins	Ties	Losses
PRODEN (2020)	26	**36**	10
CONF+PRODEN (no correction)	**37**	24	11
CONF+PRODEN	**44**	21	7
CC (2020)	17	**36**	19
CONF+CC	19	**28**	25
VALEN (2021)	3	31	**38**
CONF+VALEN	4	26	**42**
CAVL (2022)	8	28	**36**
CONF+CAVL	5	29	**38**
POP (2023)	27	**38**	7
CONF+POP	**44**	22	6
CROSEL (2024)	**36**	29	7
CONF+CROSEL	**49**	19	4

5.3 RESULTS

Predictive performance. Table 1 presents the average test-set accuracies for all competitors on all datasets. We benchmark our conformal candidate cleaning technique combined with all approaches in Section 5.1, which is marked by CONF+METHOD. An overview of significant differences is in Table 2. There, we compare the respective method to all the other approaches. All significance tests use a paired student t-test with a confidence level of 5 %.

The approaches CONF+PRODEN, CONF+POP, and CONF+CROSEL that combine the respective approaches with our candidate cleaning strategy win most often (Table 2). Conformal candidate cleaning makes PRODEN win 18 more direct comparisons, POP win 17 more direct comparisons, and CROSEL win 13 more direct comparisons advancing the state-of-the-art prediction performance. These methods significantly benefit from our pruning.

The approaches CC, VALEN, and CAVL yield similar performances when combined with conformal candidate cleaning. For VALEN and CAVL, we attribute this to the fact that their methods already use pseudo-labeling internally, that is, they treat the most likely label as the possible correct label, which diminishes the positive effect of pruning candidates.

Ablation study. Additionally, we perform an ablation experiment regarding our correction method proposed in Theorem 4.4. The approach CONF+PRODEN (no correction) uses conformal predictions based on the labels provided by the PLL classifier without our proposed correction method, which is equivalent to a fixed α_r. Table 2 shows that, while CONF+PRODEN (no correction) is already a significant improvement over PRODEN, our PLL correction strategy improves performance even further by incorporating the possible approximation error of the trained classifier. We limit our ablation study to PRODEN due to runtime constraints.

Our experiments show that the proposed method yields significant improvements over a wide range of existing PLL models (PRODEN, POP, and CROSEL) and advances the state-of-the-art prediction performance with the method CONF+CROSEL.

Acknowledgements

This work was supported by the German Research Foundation (DFG) Research Training Group GRK 2153: *Energy Status Data — Informatics Methods for its Collection, Analysis and Exploitation* and by the pilot program Core-Informatics of the Helmholtz Association (HGF).

References

Jean-Yves Audibert. *PAC-Bayesian statistical learning theory*. PhD thesis, Université Paris, 2004.

Wei-Xuan Bao, Jun-Yi Hang, and Min-Ling Zhang. Partial label dimensionality reduction via confidence-based dependence maximization. In *Conference on Knowledge Discovery and Data Mining*, pages 46–54, 2021.

Wei-Xuan Bao, Jun-Yi Hang, and Min-Ling Zhang. Submodular feature selection for partial label learning. In *Conference on Knowledge Discovery and Data Mining*, pages 26–34, 2022.

Rina Foygel Barber, Emmanuel J. Candes, Aaditya Ramdas, and Ryan J. Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.

Peter L. Bartlett and Shahar Mendelson. Empirical minimization. *Probability Theory and Related Fields*, 135(3): 311–334, 2006.

Forrest Briggs, Xiaoli Z. Fern, and Raviv Raich. Rank-loss support instance machines for MIML instance annotation. In *Conference on Knowledge Discovery and Data Mining*, pages 534–542, 2012.

Tom Bylander. Learning linear threshold functions in the presence of classification noise. In *Conference on Computational Learning Theory*, pages 340–347, 1994.

Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *CoRR*, abs/1812.01718, 2018.

Timothée Cour, Benjamin Sapp, and Ben Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12:1501–1536, 2011.

Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *Annals of Mathematical Statistics*, 27(3):642–669, 1956.

Lei Feng and Bo An. Partial label learning with self-guided retraining. In *AAAI Conference on Artificial Intelligence*, pages 3542–3549, 2019.

Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. In *Advances in Neural Information Processing Systems*, 2020.

Tobias Fuchs, Florian Kalinke, and Klemens Böhm. Partial-label learning with a reject option. *Transactions on Machine Learning Research*, January 2025.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Xiuwen Gong, Nitin Bisht, and Guandong Xu. Does label smoothing help deep partial label learning? In *International Conference on Machine Learning*, 2024.

Yves Grandvalet. Logistic regression for partial labels. In *Information Processing and Management of Uncertainty in Knowledge-based Systems*, 2002.

Peter D. Grünwald and Nishant A. Mehta. Fast rates for general unbounded loss functions: From ERM to generalized Bayes. *Journal of Machine Learning Research*, 21: 56:1–56:80, 2020.

Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *European Conference on Computer Vision*, pages 634–647, 2010.

Mohamed Farouk Abdel Hady and Friedhelm Schwenker. Semi-supervised learning. In *Handbook on Neural Information Processing*, volume 49, pages 215–239. Springer, 2013.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Shuo He, Chaojie Wang, Guowu Yang, and Lei Feng. Candidate label set pruning: A data-centric perspective for deep partial-label learning. In *International Conference on Learning Representations*, 2024.

Mark J. Huiskes and Michael S. Lew. The MIR flickr retrieval evaluation. In *International Conference on Multimedia Information Retrieval*, pages 39–43, 2008.

Eyke Hüllermeier and Jürgen Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):168–179, 2005.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

Takashi Ishida, Gang Niu, Aditya Krishna Menon, and Masashi Sugiyama. Complementary-label learning for arbitrary losses and models. In *International Conference on Machine Learning*, pages 2971–2980, 2019.

Alireza Javanmardi, Yusuf Sale, Paul Hofman, and Eyke Hüllermeier. Conformal prediction with partially labeled data. In *Conformal and Probabilistic Prediction with Applications*, pages 251–266, 2023.

Rong Jin and Zoubin Ghahramani. Learning with multiple labels. In *Advances in Neural Information Processing Systems*, pages 897–904, 2002.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `http://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf`.

Yann LeCun, Corinna Cortes, and Christopher J. C. Burges. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist, 1999.

Jing Lei. Classification with confidence. *Biometrika*, 101 (4):755–769, 2014.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pages 19730–19742, 2023.

Li-Ping Liu and Thomas G. Dietterich. A conditional multinomial mixture model for superset label learning. In *Advances in Neural Information Processing Systems*, pages 557–565, 2012.

Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *International Conference on Machine Learning*, pages 6500–6510, 2020.

Anqi Mao, Mehryar Mohri, and Yutao Zhong. Predictor-rejector multi-class abstention: Theoretical analysis and algorithms. In *International Conference on Algorithmic Learning Theory*, pages 822–867, 2024.

Hussein Mozannar, Hunter Lang, Dennis Wei, Prasanna Sattigeri, Subhro Das, and David A. Sontag. Who should predict? Exact algorithms for learning to defer to humans. In *International Conference on Artificial Intelligence and Statistics*, pages 10520–10545, 2023.

Michael Naaman. On the tight constant in the multivariate Dvoretzky-Kiefer-Wolfowitz inequality. *Statistics and Probability Letters*, 173:1–8, 2021.

Harikrishna Narasimhan, Aditya Krishna Menon, Wittawat Jitkrittum, Neha Gupta, and Sanjiv Kumar. Learning to reject meets long-tail learning. In *International Conference on Learning Representations*, 2024.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Nam Nguyen and Rich Caruana. Classification with partial labels. In *Conference on Knowledge Discovery and Data Mining*, pages 551–559, 2008.

Mauricio Sadinle, Jing Lei, and Larry A. Wasserman. Least ambiguous set-valued classifiers with bounded error levels. *Journal of the American Statistical Association*, 114 (525):223–234, 2019.

Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, pages 369–386, 2019.

Cai-Zhi Tang and Min-Ling Zhang. Confidence-rated discriminative partial label learning. In *AAAI Conference on Artificial Intelligence*, pages 2611–2617, 2017.

Shiyu Tian, Hongxin Wei, Yiqun Wang, and Lei Feng. CroSel: Cross selection of confident pseudo labels for partial-label learning. In *Conference on Computer Vision and Pattern Recognition*, 2024.

Alexander B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1): 135–166, 2004.

Tim van Erven, Peter D. Grünwald, Nishant A. Mehta, Mark D. Reid, and Robert C. Williamson. Fast rates in statistical and online learning. *Journal of Machine Learning Research*, 16:1793–1861, 2015.

Deng-Bao Wang, Li Li, and Min-Ling Zhang. Adaptive graph guided disambiguation for partial label learning. In *International Conference on Knowledge Discovery and Data Mining*, pages 83–91, 2019.

Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. PiCO: Contrastive label disambiguation for partial label learning. In *International Conference on Learning Representations*, 2022.

Wei Wang and Min-Ling Zhang. Partial label learning with discrimination augmentation. In *Conference on Knowledge Discovery and Data Mining*, pages 1920–1928, 2022.

Paul Werbos. *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. PhD thesis, Harvard University, 1974.

Xuan Wu and Min-Ling Zhang. Towards enabling binary decomposition for partial label learning. In *International Joint Conference on Artificial Intelligence*, pages 2868–2874, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2018.

Ning Xu, Jiaqi Lv, and Xin Geng. Partial label learning via label enhancement. In *AAAI Conference on Artificial Intelligence*, pages 5557–5564, 2019.

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

Ning Xu, Biao Liu, Jiaqi Lv, Congyu Qiao, and Xin Geng. Progressive purification for instance-dependent partial label learning. In *International Conference on Machine Learning*, pages 38551–38565, 2023.

Fei Yu and Min-Ling Zhang. Maximum margin partial label learning. In *Asian Conference on Machine Learning*, pages 573–593, 2017.

Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. Learning with biased complementary labels. In *European Conference on Computer Vision*, pages 69–85, 2018.

Zinan Zeng, Shijie Xiao, Kui Jia, Tsung-Han Chan, Shenghua Gao, Dong Xu, and Yi Ma. Learning by associating ambiguously labeled images. In *Conference on Computer Vision and Pattern Recognition*, pages 708–715, 2013.

Fei Zhang, Lei Feng, Bo Han, Tongliang Liu, Gang Niu, Tao Qin, and Masashi Sugiyama. Exploiting class activation value for partial-label learning. In *International Conference on Learning Representations*, 2022a.

Min-Ling Zhang and Fei Yu. Solving the partial label learning problem: An instance-based approach. In *International Joint Conference on Artificial Intelligence*, pages 4048–4054, 2015.

Min-Ling Zhang, Bin-Bin Zhou, and Xu-Ying Liu. Partial label learning via feature-aware disambiguation. In *International Conference on Knowledge Discovery and Data Mining*, pages 1335–1344, 2016.

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

Min-Ling Zhang, Jing-Han Wu, and Wei-Xuan Bao. Disambiguation enabled linear discriminant analysis for partial label dimensionality reduction. *Transactions on Knowledge Discovery from Data*, 16(4):72:1–72:18, 2022b.

Partial-Label Learning with Conformal Candidate Cleaning (Supplementary Material)

Tobias Fuchs[1] **Florian Kalinke**[1]

[1]Information Systems, Karlsruhe Institute of Technology, Karlsruhe, Germany

A PROOFS

This section collects our proofs. Section A.1 contains the proof of Theorem 4.1, Section A.2 that of Lemma 4.3, and Section A.3 that of Theorem 4.4.

A.1 PROOF OF THEOREM 4.1

Let C be an optimal solution of (6). Then, we have

$$\begin{aligned}
\mathbb{P}_{XS}(S \cap C(X) \neq \emptyset) &= 1 - \mathbb{P}_{XS}(S \cap C(X) = \emptyset) = 1 - \mathbb{P}_{XS}(\forall y \in S, y \notin C(X)) \geq 1 - \mathbb{P}_{XS}(\exists y \in S, y \notin C(X)) \\
&\overset{(a)}{=} 1 - \sum_{y \in \mathcal{Y}} \mathbb{P}(Y = y, y \in S, y \notin C(X)) = 1 - \mathbb{P}(Y \in S, Y \notin C(X)) \\
&\overset{(b)}{=} 1 - \int_{\mathcal{X} \times \mathcal{Y}} \mathbb{P}_{S|X=x,Y=y}(y \in S, y \notin C(x)) \, \mathrm{d}\, \mathbb{P}_{XY}(x, y) \\
&\overset{(c)}{=} 1 - \int_{\mathcal{X} \times \mathcal{Y}} \underbrace{\mathbb{P}_{S|X=x,Y=y}(y \in S)}_{\overset{(d)}{=} 1} \mathbb{P}_{S|X=x,Y=y,y \in S}(y \notin C(x)) \, \mathrm{d}\, \mathbb{P}_{XY}(x, y) \\
&= 1 - \int_{\mathcal{X} \times \mathcal{Y}} \mathbb{P}_{S|X=x,Y=y,y \in S}(y \notin C(x)) \, \mathrm{d}\, \mathbb{P}_{XY}(x, y) \\
&\overset{(e)}{=} 1 - \int_{\mathcal{X} \times \mathcal{Y}} \mathbb{P}_{S|X=x,Y=y}(y \notin C(x)) \, \mathrm{d}\, \mathbb{P}_{XY}(x, y) \\
&\overset{(f)}{=} 1 - \int_{\mathcal{X} \times \mathcal{Y}} \mathbb{1}_{\{y \notin C(x)\}} \, \mathrm{d}\, \mathbb{P}_{XY}(x, y) \\
&= 1 - \mathbb{P}_{XY}(Y \notin C(X)) = \mathbb{P}_{XY}(Y \in C(X)) \overset{(g)}{\geq} 1 - \alpha,
\end{aligned}$$

where (a) is implied by the law of total probability holding for the discrete Y taking mutually exclusive values in $y \in \mathcal{Y}$, (b) holds by the tower rule, (c) holds by the chain rule of conditional probability, (d) holds as $\mathbb{P}_{S|X=x,Y=y}(y \in S) = 1$ for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, (e) holds by independence, (f) holds as $\mathbb{P}_{S|X=x,Y=y}(y \notin C(x))$ is either one if $y \notin C(x)$ or zero if $y \in C(x)$, and (g) holds by our imposed assumption.

A.2 PROOF OF LEMMA 4.3

We prove parts (i) and (ii) separately in the following.

Proof of (i)**.** To proof the result, we first show that for any $\hat{f}$, one has the expectation bound

$$\mathbb{E}_{XY}\left[\left|\hat{f}_Y(X) - f_Y^*(X)\right|\right] \leq \frac{\lambda\sqrt{B}}{2^{\frac{1}{2}\beta}}\left(R(\hat{f}) - R(f^*)\right)^{\frac{1}{2}\beta}, \tag{13}$$

for some constants $\beta \in (0,1]$ and $B, \lambda > 0$. We then apply a known result (recalled in Theorem C.4) to obtain the stated concentration inequality. The details are as follows.

To prove (13), notice that

$$\begin{aligned}
\mathbb{E}_{XY}\left[\left|\hat{f}_Y(X) - f_Y^*(X)\right|\right] &= \left(\left(\mathbb{E}_{XY}\left[\left|\hat{f}_Y(X) - f_Y^*(X)\right|\right]\right)^2\right)^{\frac{1}{2}} \overset{(a)}{\leq} \left(\mathbb{E}_{XY}\left[\left|\hat{f}_Y(X) - f_Y^*(X)\right|^2\right]\right)^{\frac{1}{2}} \\
&\overset{(b)}{\leq} \lambda\left(\mathbb{E}_{XY}\left[\left|\ell(\hat{f}(X),Y) - \ell(f^*(X),Y)\right|^2\right]\right)^{\frac{1}{2}} = \lambda\left(\mathbb{E}_{XY}\left[\left(\ell(\hat{f}(X),Y) - \ell(f^*(X),Y)\right)^2\right]\right)^{\frac{1}{2}} \\
&\overset{(c)}{\leq} \lambda\sqrt{B}\left(\mathbb{E}_{XY}\left[\ell(\hat{f}(X),Y) - \ell(f^*(X),Y)\right]\right)^{\frac{1}{2}\beta} \overset{(d)}{=} \lambda\sqrt{B}\left(\mathbb{E}_{XY}\left[\ell(\hat{f}(X),Y)\right] - \mathbb{E}_{XY}[\ell(f^*(X),Y)]\right)^{\frac{1}{2}\beta} \\
&\overset{(e)}{=} \lambda\sqrt{B}\frac{1}{2^{\frac{1}{2}\beta}}\left(R(\hat{f}) - R(f^*)\right)^{\frac{1}{2}\beta},
\end{aligned}$$

using the following observations. (a) is implied by Jensen's inequality. Next, we note that $z \mapsto -\log(z)$ satisfies the λ-bi-Lipschitz condition on $[\epsilon, 1]$ (Lemma B.1), implying that

$$\left|\hat{f}_Y(X) - f_Y^*(X)\right| \leq \lambda\left|-\log\left(\hat{f}_Y(X)\right) - \left(-\log\left(f_Y^*(X)\right)\right)\right| = \left|\ell\left(\hat{f}(X),Y\right) - \ell\left(f^*(X),Y\right)\right|, \tag{14}$$

where we used the definition of ℓ for the equality. Using (14) together with the monotonicity of the L_2-norm yields (b). (c) holds by the assumed (β, B)-Bernstein condition. The linearity of expectations gives (d) and an identity recalled in Theorem C.3 yields (e).

Now, to obtain the probabilistic bound, we observe that

$$\begin{aligned}
&\mathbb{P}_n\left(\mathbb{E}_{XY}\left[|\hat{f}_Y(X) - f_Y^*(X)|\right] \leq M_1\left(\frac{\log(1/\delta_1)}{n}\right)^{\frac{1}{4}\beta}\right) \\
&\overset{(13)}{\geq} \mathbb{P}_n\left(\lambda\sqrt{B}\frac{1}{2^{\frac{1}{2}\beta}}\left(R(\hat{f}) - R(f^*)\right)^{\frac{1}{2}\beta} \leq M_1\left(\frac{\log(1/\delta_1)}{n}\right)^{\frac{1}{4}\beta}\right) \\
&\overset{(a)}{=} \mathbb{P}_n\left(\lambda^{\frac{2}{\beta}}B^{\frac{1}{\beta}}\frac{1}{2}\left(R(\hat{f}) - R(f^*)\right) \leq M_1\left(\frac{\log(1/\delta_1)}{n}\right)^{\frac{1}{2}}\right) \overset{(b)}{\geq} 1 - \delta_1.
\end{aligned}$$

In (a), we notice that both sides of the inequality are nonnegative and apply the function $z \mapsto z^{2/\beta}$, which is monotonically increasing on $\mathbb{R}^+$. We conclude the proof of part (i) with an application of Theorem C.4 in (b), where we let $M_1 = M\lambda^{\frac{2}{\beta}}B^{\frac{1}{\beta}}\frac{1}{2}$ (with M defined in the external result).

Proof of (ii)**.** The proof of the lemma proceeds in three steps. In step 1, we will show that

$$\mathbb{E}_{XY}\left[\mathbb{1}_{\{\arg\max_{j\in\mathcal{Y}}\hat{f}_j(X) \neq \arg\max_{j\in\mathcal{Y}} f_j^*(X)\}}\right] \leq \frac{1}{1-\delta_5}\mathbb{E}_{XY}\left[\left(\mathbb{1}_{\{\arg\max_{j\in\mathcal{Y}}\hat{f}_j(x)\neq y\}} - \mathbb{1}_{\{\arg\max_{j\in\mathcal{Y}} f_j^*(x)\neq y\}}\right)^2\right],$$

which will allow us to obtain, in step 2, that for some constants $\beta \in (0,1]$, $B > 0$ and $\delta_5 \in [0,1)$, one has

$$\mathbb{P}_X\left[\arg\max_{j\in\mathcal{Y}}\hat{f}_j(X) \neq \arg\max_{j\in\mathcal{Y}} f_j^*(X)\right] \leq \frac{B}{1-\delta_5}\left(R(\hat{f}) - R(f^*)\right)^{\beta}. \tag{15}$$

The result will then follow by an application of Theorem C.4, which we elaborate in step 3. The details are as follows.

Step 1. Note that we have

$$1 \overset{(a)}{\leq} \frac{1}{1-\delta_5}\mathbb{P}_{Y|X=x}\left(Y \in \{\hat{y}_x, y_x^*\}\right) \overset{(b)}{\leq} \frac{1}{1-\delta_5}\sum_{y\in\{\hat{y}_x, y_x^*\}}\mathbb{P}_{Y|X=x}\left(Y = y\right), \tag{16}$$

with the assumption used in (a) and a union bound implying (b).

To conclude the first step, we obtain

$$\begin{aligned}
&\mathbb{E}_{XY}\left[\mathbb{1}_{\{\arg\max_{j\in\mathcal{Y}}\hat{f}_j(X)\neq\arg\max_{j\in\mathcal{Y}}f_j^*(X)\}}\right] \\
&\quad\overset{(a)}{\leq}\frac{1}{1-\delta_5}\,\mathbb{E}_X\left[\sum_{y\in\{y_X^*,\hat{y}_X\}}\mathbb{P}_{Y|X}(Y=y)\mathbb{1}_{\{\hat{y}_X\neq y_X^*\}}\right] \\
&\quad\overset{(b)}{=}\frac{1}{1-\delta_5}\,\mathbb{E}_X\left[\sum_{y\in\{y_X^*,\hat{y}_X\}}\mathbb{P}_{Y|X}(Y=y)(\mathbb{1}_{\{\hat{y}_X\neq y\}}-\mathbb{1}_{\{y_X^*\neq y\}})^2\right] \\
&\quad\overset{(c)}{\leq}\frac{1}{1-\delta_5}\,\mathbb{E}_X\left[\sum_{y\in\mathcal{Y}}\mathbb{P}_{Y|X}(Y=y)(\mathbb{1}_{\{\hat{y}_X\neq y\}}-\mathbb{1}_{\{y_X^*\neq y\}})^2\right] \\
&\quad\overset{(d)}{=}\frac{1}{1-\delta_5}\,\mathbb{E}_{XY}\left[\left(\mathbb{1}_{\{\arg\max_{j\in\mathcal{Y}}\hat{f}_j(x)\neq y\}}-\mathbb{1}_{\{\arg\max_{j\in\mathcal{Y}}f_j^*(x)\neq y\}}\right)^2\right],
\end{aligned}\tag{17}$$

where (a) is implied by (16) and the indicator function being nonnegative. For (b), we must show that $\mathbb{1}_{\{\hat{y}_x\neq y_x^*\}} = (\mathbb{1}_{\{\hat{y}_x\neq y\}}-\mathbb{1}_{\{y_x^*\neq y\}})^2$ for any (fixed) $x\in\mathcal{X}$ and $y\in\{\hat{y}_x,y_x^*\}$; it suffices to check the three cases.

- If $y=\hat{y}_x=y_x^*$, then $0=0$,
- if $\hat{y}_x\neq y_x^*$ and $y=\hat{y}_x$, then $1=1$, and
- if $\hat{y}_x=y_x^*$ and $y\neq\hat{y}_x$, then $1=1$.

In (c), we add nonnegative terms and (d) holds by the definition of the expectation.

Step 2. We relax the l.h.s. in (15) to

$$\begin{aligned}
\mathbb{P}_X\left[\arg\max_{j\in\mathcal{Y}}\hat{f}_j(X)\neq\arg\max_{j\in\mathcal{Y}}f_j^*(X)\right]&\overset{(a)}{=}\mathbb{E}_X\left[\mathbb{1}_{\{\arg\max_{j\in\mathcal{Y}}\hat{f}_j(X)\neq\arg\max_{j\in\mathcal{Y}}f_j^*(X)\}}\right] \\
&\overset{(b)}{=}\mathbb{E}_{XY}\left[\mathbb{1}_{\{\arg\max_{j\in\mathcal{Y}}\hat{f}_j(X)\neq\arg\max_{j\in\mathcal{Y}}f_j^*(X)\}}\right] \\
&\overset{(c)}{\leq}\frac{1}{1-\delta_5}\,\mathbb{E}_{XY}\left[(\mathbb{1}_{\{\arg\max_{j\in\mathcal{Y}}\hat{f}_j(X)\neq Y\}}-\mathbb{1}_{\{\arg\max_{j\in\mathcal{Y}}f_j^*(X)\neq Y\}})^2\right] \\
&\overset{(d)}{=}\frac{1}{1-\delta_5}\,\mathbb{E}_{XY}\left[(\ell(\hat{f}(X),Y)-\ell(f^*(X),Y))^2\right]\overset{(e)}{\leq}\frac{B}{1-\delta_5}\left(R(\hat{f})-R(f^*)\right)^\beta,
\end{aligned}$$

obtaining the r.h.s. and establishing (15). The details are as follows. In (a), we use that a probability can be written as the expectation of an indicator function. We notice in (b) that the integrand does not depend on Y. Regarding (c), with $\hat{y}_x, y_x^*$ defined as in the statement, we use (17) obtained in step 1. Defining $\ell:[0,1]^k\to\mathbb{R}_{\geq 0}$, $(p,y)\mapsto\mathbb{1}_{\{\arg\max_{y'\in\mathcal{Y}}p_{y'}\neq y\}}$ as the usual 0-1-loss yields (d) and the (β,B)-Bernstein condition gives (e).

Step 3. It remains to obtain the probabilistic bound. We have that

$$\begin{aligned}
&\mathbb{P}_n\left(\mathbb{P}_X\left[\arg\max_{j\in\mathcal{Y}}\hat{f}_j(X)\neq\arg\max_{j\in\mathcal{Y}}f_j^*(X)\right]\leq M_2\left(\frac{\log(1/\delta_2)}{n}\right)^{\frac{1}{2}\beta}\right) \\
&\quad\overset{(15)}{\geq}\mathbb{P}_n\left(\frac{B}{1-\delta_5}\left(R(\hat{f})-R(f^*)\right)^\beta\leq M_2\left(\frac{\log(1/\delta_2)}{n}\right)^{\frac{1}{2}\beta}\right) \\
&\quad\overset{(a)}{=}\mathbb{P}_n\left(\left(\frac{B}{1-\delta_5}\right)^{\frac{1}{\beta}}\left(R(\hat{f})-R(f^*)\right)\leq M_2^{\frac{1}{\beta}}\left(\frac{\log(1/\delta_2)}{n}\right)^{\frac{1}{2}}\right)\overset{(b)}{\geq}1-\delta_2,
\end{aligned}$$

where we apply the monotonically increasing $z\mapsto z^{1/\beta}$ in (a). In (b), we set $M_2^{\frac{1}{\beta}}=M\left(\frac{B}{1-\delta_5}\right)^{\frac{1}{\beta}}$ and apply Theorem C.4 (with M given there). This concludes the proof of (ii).

A.3 PROOF OF THEOREM 4.4

To obtain the statement, we first show that one has the following decomposition. For any $\alpha \in (0,1)$ and some $\delta_3 \in (0,1)$,

$$\mathbb{P}_X\Big[f^*_{y^*_X}(X) \geq t_\alpha - \delta_3\Big] \geq \underbrace{\mathbb{P}_X\Big[f^*_{y^*_X}(X) \geq \hat{f}_{y^*_X}(X) - \delta_3\Big]}_{=:t_1} + \underbrace{\mathbb{P}_X\Big[\hat{f}_{y^*_X}(X) = \hat{f}_{\hat{y}_X}(X)\Big]}_{=:t_2} + \underbrace{\mathbb{P}_X\Big[\hat{f}_{\hat{y}_X}(X) \geq t_\alpha\Big]}_{=:t_3} - 2. \quad (18)$$

We will then obtain lower bounds on the individual terms t_1, t_2, and t_3, and show that their combination implies the stated result.

Decomposition. Let $A_1 = \{X : f^*_{y^*_X}(X) \geq \hat{f}_{y^*_X}(X) - \delta_3\}$, $A_2 = \{X : \hat{f}_{y^*_X}(X) = \hat{f}_{\hat{y}_X}(X)\}$, $A_3 = \{X : \hat{f}_{\hat{y}_X}(X) \geq t_\alpha\}$, and $B = \{X : f^*_{y^*_X}(X) \geq t_\alpha - \delta_3\}$. Using these definitions, we obtain that

$$\begin{aligned}
\mathbb{P}_X\Big[f^*_{y^*_X}(X) \geq t_\alpha - \delta_3\Big] &\overset{(a)}{=} \mathbb{P}_X[B] \overset{(b)}{\geq} \mathbb{P}_X[A_1 \cap A_2 \cap A_3] \overset{(c)}{=} 1 - \mathbb{P}_X[(A_1 \cap A_2 \cap A_3)^c] \overset{(d)}{=} 1 - \mathbb{P}_X[A_1^c \cup A_2^c \cup A_3^c] \\
&\overset{(e)}{\geq} 1 - \mathbb{P}_X[A_1^c] - \mathbb{P}_X[A_2^c] - \mathbb{P}_X[A_2^c] \overset{(f)}{=} 1 - (1 - \mathbb{P}_X[A_1]) - (1 - \mathbb{P}_X[A_2]) - (1 - \mathbb{P}_X[A_3]) \\
&\overset{(g)}{=} \underbrace{\mathbb{P}_X[A_1]}_{=t_1} + \underbrace{\mathbb{P}_X[A_2]}_{=t_2} + \underbrace{\mathbb{P}_X[A_3]}_{=t_3} - 2, \qquad (19)
\end{aligned}$$

with the following details. (a) is by the preceeding definition of the B set. For (b), we have to show that $B \supseteq A_1 \cap A_2 \cap A_3$, which implies that $\mathbb{P}_X[B] \geq \mathbb{P}_X[A_1 \cap A_2 \cap A_3]$. Let $x \in A_1 \cap A_2 \cap A_3$, then

$$f^*_{y^*_x}(x) \geq \hat{f}_{y^*_x}(x) - \delta_3 = \hat{f}_{\hat{y}_x}(x) - \delta_3 \geq t_\alpha - \delta_3 \quad \Longrightarrow \quad f^*_{y^*_x}(x) \geq t_\alpha - \delta_3.$$

Therefore, $x \in B$, proving (b). (c) considers complementary events and De Morgan's laws yield (d). In (e), we use the inclusion-exclusion principle, where we ignore a few positive terms to obtain the inequality. Considering complementary events gives (f) and cancellations yield (g). This proves (18).

Term t_1. To obtain a bound on the first term, we obtain an expectation bound, which together with Markov's inequality and Lemma 4.3 (i) will give the result. The expectation bound is

$$\begin{aligned}
\mathbb{E}_X\Big[|\hat{f}_{y^*_X}(X) - f^*_{y^*_X}(X)|\Big] &\overset{(a)}{\leq} \mathbb{E}_X\left[\frac{\mathbb{P}_{Y|X}(Y = y^*_X)}{1 - \delta_6}\,|\hat{f}_{y^*_X}(X) - f^*_{y^*_X}(X)|\right] \\
&\overset{(b)}{\leq} \frac{1}{1 - \delta_6}\,\mathbb{E}_X\Big[\sum_{y \in \mathcal{Y}} \mathbb{P}_{Y|X}(Y = y)\,|\hat{f}_y(X) - f^*_y(X)|\Big] \overset{(c)}{=} \frac{1}{1 - \delta_6}\,\mathbb{E}_{XY}\Big[|\hat{f}_Y(X) - f^*_Y(X)|\Big], \qquad (20)
\end{aligned}$$

with (a) implied by the assumption $\mathbb{P}_{Y|X}(Y = y^*_X) \geq 1 - \delta_6$ guaranteeing that $1 \leq \frac{\mathbb{P}_{Y|X}(Y = y^*_X)}{1 - \delta_6}$. In (b), we use that $y^*_X \in \mathcal{Y}$ and that all terms in the sum are nonnegative. Using a property of the expectation of a joint distribution yields (c).

Next, Markov's inequality (recalled in Lemma C.5) implies that

$$\mathbb{P}_X\Big[|\hat{f}_{y^*_X}(X) - f^*_{y^*_X}(X)| \geq \delta_3\Big] \overset{\text{C.5}}{\leq} \frac{1}{\delta_3}\,\mathbb{E}_X\Big[|\hat{f}_{y^*_X}(X) - f^*_{y^*_X}(X)|\Big] \overset{(20)}{\leq} \frac{1}{\delta_3(1 - \delta_6)}\,\mathbb{E}_{XY}\Big[|\hat{f}_Y(X) - f^*_Y(X)|\Big]. \quad (21)$$

Finally, we have that

$$\begin{aligned}
&\mathbb{P}_n\left[\mathbb{P}_X\left[f^*_{y^*_X}(X) \geq \hat{f}_{y^*_X}(X) - \delta_3\right] \geq 1 - \frac{1}{\delta_3(1-\delta_6)} M_1 \left(\frac{\log(1/\delta_1)}{n}\right)^{\frac{1}{4}\beta}\right] \\
&\quad \overset{(a)}{=} \mathbb{P}_n\left[\mathbb{P}_X\left[\hat{f}_{y^*_X}(X) - f^*_{y^*_X}(X) \leq \delta_3\right] \geq 1 - \frac{1}{\delta_3(1-\delta_6)} M_1 \left(\frac{\log(1/\delta_1)}{n}\right)^{\frac{1}{4}\beta}\right] \\
&\quad \overset{(b)}{\geq} \mathbb{P}_n\left[\mathbb{P}_X\left[\left|\hat{f}_{y^*_X}(X) - f^*_{y^*_X}(X)\right| \leq \delta_3\right] \geq 1 - \frac{1}{\delta_3(1-\delta_6)} M_1 \left(\frac{\log(1/\delta_1)}{n}\right)^{\frac{1}{4}\beta}\right] \\
&\quad \overset{(c)}{\geq} \mathbb{P}_n\left[1 - \mathbb{P}_X\left[\left|\hat{f}_{y^*_X}(X) - f^*_{y^*_X}(X)\right| \leq \delta_3\right] \leq \frac{1}{\delta_3(1-\delta_6)} M_1 \left(\frac{\log(1/\delta_1)}{n}\right)^{\frac{1}{4}\beta}\right] \\
&\quad \overset{(d)}{\geq} \mathbb{P}_n\left[\mathbb{P}_X\left[\left|\hat{f}_{y^*_X}(X) - f^*_{y^*_X}(X)\right| > \delta_3\right] \leq \frac{1}{\delta_3(1-\delta_6)} M_1 \left(\frac{\log(1/\delta_1)}{n}\right)^{\frac{1}{4}\beta}\right] \\
&\quad \overset{(21)}{\geq} \mathbb{P}_n\left[\frac{1}{\delta_3(1-\delta_6)} \mathbb{E}_{XY}\left[|\hat{f}_Y(X) - f^*_Y(X)|\right] \leq \frac{1}{\delta_3(1-\delta_6)} M_1 \left(\frac{\log(1/\delta_1)}{n}\right)^{\frac{1}{4}\beta}\right] \\
&\quad \overset{(e)}{\geq} \mathbb{P}_n\left[\mathbb{E}_{XY}\left[|\hat{f}_Y(X) - f^*_Y(X)|\right] \leq M_1 \left(\frac{\log(1/\delta_1)}{n}\right)^{\frac{1}{4}\beta}\right] \overset{(f)}{\geq} 1 - \delta_1,
\end{aligned} \tag{22}$$

where we rearrange the l.h.s. of the inequality in (a). In (b), we consider the absolute value, decreasing the overall probability. In (c), we subtract 1 on both sides and multiply by -1. In (d), we consider the complement of the event. In (e), we simplify and Lemma 4.3(i) yields (f).

Term t_2. The observation $\mathbb{P}_X\left[\hat{f}_{y^*_X}(X) = \hat{f}_{\hat{y}_X}(X)\right] \geq \mathbb{P}_X[y^*_X = \hat{y}_X]$ implies that

$$\begin{aligned}
&\mathbb{P}_n\left[\mathbb{P}_X\left[\hat{f}_{y^*_X}(X) = \hat{f}_{\hat{y}_X}(X)\right] \geq 1 - M_2\left(\frac{\log(1/\delta_2)}{n}\right)^{\frac{1}{2}\beta}\right] \overset{(a)}{\geq} \mathbb{P}_n\left[\mathbb{P}_X[y^*_X = \hat{y}_X] \geq 1 - M_2\left(\frac{\log(1/\delta_2)}{n}\right)^{\frac{1}{2}\beta}\right] \\
&\quad \overset{(b)}{=} \mathbb{P}_n\left[\mathbb{P}_X[y^*_X \neq \hat{y}_X] \leq M_2\left(\frac{\log(1/\delta_2)}{n}\right)^{\frac{1}{2}\beta}\right] \overset{(c)}{\geq} 1 - \delta_2,
\end{aligned} \tag{23}$$

where (a) holds by the preceding observation. In (b), we subtract 1 on both sides, multiply by -1, and consider the complement of the l.h.s. Inequality (c) was shown in Lemma 4.3(ii).

Term t_3. For bounding the third term, we use the well-known Dvoretzky-Kiefer-Wolfowitz inequality (recalled in Theorem C.1) In particular, we have

$$\begin{aligned}
&\mathbb{P}_n\left[\mathbb{P}_X[\hat{f}_{\hat{y}_X}(X) \geq t_\alpha] \geq 1 - \left(\alpha + \sqrt{\frac{\log(2/\delta_4)}{2n}}\right)\right] \overset{(a)}{=} \mathbb{P}_n\left[1 - F_{\hat{f}_{\hat{y}_X}(X)}(t_\alpha) \geq 1 - \left(\alpha + \sqrt{\frac{\log(2/\delta_4)}{2n}}\right)\right] \\
&\quad \overset{(b)}{=} \mathbb{P}_n\left[F_{\hat{f}_{\hat{y}_X}(X)}(t_\alpha) - \alpha \leq \sqrt{\frac{\log(2/\delta_4)}{2n}}\right] \overset{(c)}{\geq} \mathbb{P}_n\left[F_{\hat{f}_{\hat{y}_X}(X)}(t_\alpha) - \hat{F}_{\hat{f}_{\hat{y}_X}(X)}(t_\alpha) \leq \sqrt{\frac{\log(2/\delta_4)}{2n}}\right] \\
&\quad \overset{(d)}{\geq} \mathbb{P}_n\left[\sup_{t_\alpha}\left|F_{\hat{f}_{\hat{y}_X}(X)}(t_\alpha) - \hat{F}_{\hat{f}_{\hat{y}_X}(X)}(t_\alpha)\right| \overset{(e)}{\leq} \sqrt{\frac{\log(2/\delta_4)}{2n}}\right] \overset{(e)}{\geq} 1 - \delta_4,
\end{aligned} \tag{24}$$

where (a) holds as

$$\mathbb{P}_X[\hat{f}_{\hat{y}_X}(X) \geq t_\alpha] = 1 - \mathbb{P}_X[\hat{f}_{\hat{y}_X}(X) \leq t_\alpha] = 1 - F_{\hat{f}_{\hat{y}_X}(X)}(t_\alpha).$$

We rearrange in (b). For obtaining (c), we observe that $\hat{F}_{\hat{f}_{\hat{y}_X}(X)}(t_\alpha) \leq \alpha$. In (d), we consider the supremum, reducing the probability as the inequality becomes more strict. Theorem C.1 gives (e).

Combination of t_1, t_2, and t_3. The desired result is obtained by combining the intermediate results using that

$$\mathbb{P}_n[\mathbb{P}_X[y_X^* \in C(X)] \geq 1 - \alpha_n'] \overset{(19a)}{=} \mathbb{P}_n[\mathbb{P}_X[B] \geq 1 - \alpha_n'] \overset{(19)}{\geq} \mathbb{P}_n[\mathbb{P}_X[A_1] + \mathbb{P}_X[A_2] + \mathbb{P}_X[A_3] - 2 \geq 1 - \alpha_n'] \\ \overset{(a)}{\geq} 1 - (\delta_1 + \delta_2 + \delta_4),$$

where we use a union bound in (a) and the results obtained in (22), (23), and (24); further, we observe that

$$\left(1 - \frac{1}{\delta_3(1-\delta_6)} M_1 \left(\frac{\log(1/\delta_1)}{n}\right)^{\frac{1}{4}\beta}\right) + \left(1 - M_2 \left(\frac{\log(1/\delta_2)}{n}\right)^{\frac{1}{2}\beta}\right) + \left(1 - \alpha - \sqrt{\frac{\log(2/\delta_4)}{2n}}\right) - 2 \\ = 1 - \left(\frac{1}{\delta_3(1-\delta_6)} M_1 \left(\frac{\log(1/\delta_1)}{n}\right)^{\frac{1}{4}\beta} + M_2 \left(\frac{\log(1/\delta_2)}{n}\right)^{\frac{1}{2}\beta} + \alpha + \sqrt{\frac{\log(2/\delta_4)}{2n}}\right) = 1 - \alpha_n'.$$

B AUXILIARY RESULTS

This section collects our auxiliary results.

Lemma B.1. *Let $\varepsilon \in (0,1)$ and $f : [\varepsilon, 1] \to [0, -\log \varepsilon]$, $z \mapsto -\log z$. Then, f is $\frac{1}{\varepsilon}$-bi-Lipschitz, that is, for any $x_1, x_2 \in [\varepsilon, 1]$, it holds that $\varepsilon|x_1 - x_2| \leq |f(x_1) - f(x_2)| \leq \frac{1}{\varepsilon}|x_1 - x_2|$.*

Proof. f is continuous on $[\varepsilon, 1]$ and differentiable on $(\varepsilon, 1)$. Hence, by the mean value theorem, for any $x_1, x_2 \in [\varepsilon, 1]$, there exists $\xi \in (x_1, x_2)$ such that

$$|f(x_1) - f(x_2)| = |x_1 - x_2| \, |f'(\xi)| \, .$$

Using that $|f'(\xi)| = \frac{1}{\xi}$ satisfies $\varepsilon \leq f'(\xi) \leq \frac{1}{\varepsilon}$ as $\varepsilon \leq \xi \leq 1$ yields the stated stated claim. □

C EXTERNAL RESULTS

This section briefly summarizes external results that are necessary to prove our theorems. Theorem C.1 states the Dvoretzky-Kiefer-Wolfowitz inequality, Assumption C.2 describes the candidate generation model used in Theorem C.3, which relates the PLL risk (2) to the risk in the supervised setting. Theorem C.4 provides the estimation-error bound on which we build in our Lemma 4.3. We recall Markov's inequality in Lemma C.5.

Theorem C.1 (Dvoretzky et al. 1956; Naaman 2021, Dvoretzky-Kiefer-Wolfowitz Inequality)**.** *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and $X, X_1, \ldots, X_n \overset{i.i.d.}{\sim} \mathbb{P}$ real-valued random variables on Ω. Then, for any $\delta \in (0,1)$,*

$$\mathbb{P}_X\left(\sup_{x \in \mathbb{R}} \left|\hat{F}_X(x) - F_X(x)\right| \leq \sqrt{\frac{\log(2/\delta)}{2n}}\right) \geq 1 - \delta,$$

with $\hat{F}_X(x) = \frac{1}{n}\sum_{i=1}^n \mathbb{1}_{\{X_i \leq x\}}$ and $F_X(x) = \mathbb{P}(X \leq x)$.

Assumption C.2 (Feng et al. 2020, Eq. (5))**.** In the PLL setting (Section 2), assume that $\mathbb{P}_{XS}$ and $\mathbb{P}_{XY}$ have Lebesgue densities p_{XS} and p_{XY}, respectively, $p_{S|X,Y}(S) = p_{S|Y}(S)$, and the candidate generation model is of the form

$$p_{XS}(X, S) = \sum_{y=1}^{k} p_{S|Y=y}(S) p_{XY}(X, Y = y), \qquad \text{with} \qquad p_{S|Y=y}(S) = \begin{cases} \frac{1}{2^{k-1}-1} & \text{if } y \in S, \\ 0 & \text{else.} \end{cases}$$

The following theorem collects an identity by Feng et al. (2020).

Theorem C.3 (Feng et al. 2020, Eq. (6), (7), and (8))**.** *Let Assumption C.2 hold, $R(f)$ as in* (2)*, and the true risk of the supervised classification setting $R_{\sup}(f) := \mathbb{E}_{XY}[\ell(f(X), Y)]$. Then, $R_{\sup}(f) = \frac{1}{2}R(f)$.*

Table 3: Overview of dataset characteristics grouped into real-world partially labeled datasets (top) and supervised multi-class classification datasets with added candidate labels (bottom).

Dataset	#Instances n	#Features d	#Classes k	Avg. candidates
bird-song	4 966	38	12	2.146
lost	1 122	108	14	2.216
mir-flickr	2 778	1 536	12	2.756
msrc-v2	1 755	48	22	3.149
soccer	17 271	279	158	2.095
yahoo-news	22 762	163	203	1.915
mnist	70 000	784	10	6.304
fmnist	70 000	784	10	5.953
kmnist	70 000	784	10	6.342
svhn	99 289	3 072	10	4.878
cifar10	60 000	3 072	10	1.900
cifar100	60 000	3 072	100	1.399

Theorem C.4 (Feng et al. 2020, Theorem 4)**.** *Let $\ell : [0,1]^k \times \mathcal{Y} \to [0, M]$ be a bounded and λ-Lipschitz loss function in the first argument ($\lambda > 0$), that is, $\sup_{y \in \mathcal{Y}} |\ell(\mathbf{p}, y) - \ell(\mathbf{q}, y)| \leq \lambda \|\mathbf{p} - \mathbf{q}\|_2$ for $\mathbf{p}, \mathbf{q} \in [0,1]^k$. Further, let $\mathcal{H} = \{f : \mathcal{X} \to [0,1]^k \mid f$ measurable, $\forall x \in \mathcal{X} : \sum_{j=1}^k f_j(x) = 1\}$, $f^* = \arg\min_{f \in \mathcal{H}} R(f)$ be the true risk minimizer and $\hat{f} = \arg\min_{f \in \mathcal{H}} \hat{R}(f)$ be the empirical risk minimizer of the risks in* (2) *and* (3)*, respectively. Then, for any $\delta \in (0,1)$, with $\mathbb{P}_n$-probability of at least $1 - \delta$,*

$$R(\hat{f}) - R(f^*) \leq 4\sqrt{2}\lambda \sum_{y=1}^{k} \mathfrak{R}_n(\mathcal{H}_y) + C\sqrt{\frac{\log(2/\delta)}{2n}},$$

where $\mathfrak{R}_n(\mathcal{H}_y)$ is the empirical Rademacher complexity of $\mathcal{H}_y := \{f_y \mid f \in \mathcal{H}\}$ and some constant $C > 0$. Further, using that $\mathfrak{R}_n(\mathcal{H}_y) \leq C_{\mathcal{H}}/\sqrt{n}$ for some constants $C_{\mathcal{H}}, M > 0$, it holds with the same probability that

$$R(\hat{f}) - R(f^*) \leq M\sqrt{\frac{\log(1/\delta)}{n}}.$$

Lemma C.5 (Markov inequality)**.** *For a real-valued random variable X with probability distribution $\mathbb{P}$ and $a > 0$, it holds that*

$$\mathbb{P}(|X| \geq a) \leq \frac{\mathbb{E}(|X|)}{a}.$$

D ADDITIONAL SETUP

In our experiments, we consider twelve datasets of which Table 3 summarizes the characteristics. As mentioned in Section 5.1, we consider six state-of-the-art PLL approaches and our novel candidate cleaning technique. We choose their parameters as recommended by the respective authors.

- PRODEN (Lv et al., 2020): For a fair comparison, we use the same base models for each particular dataset. For the colored-image datasets, we use a ResNet-9 architecture (He et al., 2016). For all other image and non-image datasets, we use a standard d-300-300-300-k MLP (Werbos, 1974) with batch normalization (Ioffe and Szegedy, 2015) and ReLU activations (Glorot et al., 2011). We choose the *Adam* optimizer for training over a total of 200 epochs and use the one-cycle learning rate scheduler (Smith and Topin, 2019). Also, we use mini-batched training with a batch size of 16 for the small-scale datasets (less than 5000 samples) and of 256 for the large-scale datasets (more than 5000 samples). This balances training duration and predictive quality.
- CC (Feng et al., 2020): We use the same base models and training procedures as mentioned above for PRODEN. Otherwise, there are no additional hyperparameters for CC.

Table 4: Average test-set accuracies (± std.) on the real-world datasets. We benchmark our strategy (CONF+) as well as the cleaning method CLSP combined with all existing methods.

Method	*bird-song*	*lost*	*mir-flickr*	*msrc-v2*	*soccer*	*yahoo-news*
PRODEN (2020)	75.55 (± 1.08)	78.94 (± 3.01)	67.05 (± 1.18)	54.33 (± 1.76)	54.18 (± 0.55)	65.25 (± 1.00)
CLSP+PRODEN	74.61 (± 0.84)	61.95 (± 2.80)	60.53 (± 2.95)	51.74 (± 1.77)	31.93 (± 29.15)	50.92 (± 0.66)
CONF+P. (no)	76.27 (± 0.94)	79.56 (± 1.96)	66.07 (± 1.63)	53.00 (± 2.24)	54.63 (± 0.81)	65.42 (± 0.36)
CONF+PRODEN	76.99 (± 0.90)	80.09 (± 4.40)	66.91 (± 1.57)	54.60 (± 3.42)	54.77 (± 0.84)	65.93 (± 0.42)
CC (2020)	74.49 (± 1.57)	78.23 (± 2.11)	62.39 (± 1.87)	50.96 (± 2.03)	55.28 (± 0.96)	65.03 (± 0.51)
CLSP+CC	74.37 (± 0.91)	60.88 (± 3.71)	59.79 (± 2.29)	49.64 (± 2.06)	53.71 (± 0.99)	49.89 (± 0.30)
CONF+CC	75.01 (± 1.84)	79.38 (± 1.79)	63.37 (± 0.45)	52.45 (± 3.64)	55.52 (± 0.74)	64.35 (± 0.64)
VALEN (2021)	72.30 (± 1.83)	70.18 (± 3.44)	67.05 (± 1.48)	49.20 (± 1.37)	53.20 (± 0.88)	62.25 (± 0.45)
CLSP+VALEN	74.95 (± 0.27)	59.03 (± 2.67)	60.11 (± 1.95)	49.92 (± 1.80)	53.31 (± 0.84)	49.50 (± 0.76)
CONF+VALEN	71.22 (± 1.03)	68.41 (± 2.95)	61.61 (± 2.79)	48.37 (± 2.24)	52.49 (± 1.00)	62.16 (± 0.74)
CAVL (2022)	69.78 (± 3.00)	72.12 (± 1.08)	65.02 (± 1.34)	52.67 (± 2.32)	55.06 (± 0.48)	61.91 (± 0.46)
CLSP+CAVL	73.13 (± 1.23)	58.76 (± 1.75)	59.86 (± 2.92)	48.65 (± 2.31)	53.48 (± 0.76)	49.48 (± 0.37)
CONF+CAVL	72.00 (± 1.22)	71.24 (± 3.81)	64.42 (± 0.89)	51.63 (± 5.03)	54.85 (± 0.92)	62.43 (± 0.43)
POP (2023)	75.17 (± 1.04)	77.79 (± 2.11)	67.93 (± 1.44)	53.83 (± 0.69)	55.31 (± 0.71)	65.09 (± 0.64)
CLSP+POP	74.25 (± 0.89)	60.18 (± 2.48)	59.61 (± 1.84)	50.58 (± 1.47)	32.08 (± 29.29)	50.77 (± 0.42)
CONF+POP	77.58 (± 1.01)	78.41 (± 2.13)	66.21 (± 2.19)	54.82 (± 3.60)	56.49 (± 1.10)	65.25 (± 0.23)
CROSEL (2024)	75.11 (± 1.79)	81.24 (± 3.68)	67.58 (± 1.16)	52.23 (± 2.83)	52.64 (± 1.21)	67.72 (± 0.32)
CLSP+CROSEL	76.53 (± 1.34)	63.72 (± 2.23)	59.75 (± 2.79)	51.29 (± 1.69)	52.24 (± 0.84)	53.53 (± 0.93)
CONF+CROSEL	77.76 (± 0.50)	81.15 (± 2.57)	65.93 (± 1.94)	54.10 (± 2.75)	54.97 (± 0.65)	67.55 (± 0.22)

- VALEN (Xu et al., 2021): We use the same base models and training procedures as mentioned above for PRODEN. Additionally, we use ten warm-up epochs and the three nearest neighbors to calculate the adjacency matrix.
- CAVL (Zhang et al., 2022a): We use the same base models and training procedures as mentioned above for PRODEN. Otherwise, there are no additional hyperparameters for CAVL.
- POP (Xu et al., 2023): We use the same base models and training procedures as mentioned above for PRODEN. Also, we set $e_0 = 0.001$, $e_{end} = 0.04$, and $e_s = 0.001$. We abstain from using the data augmentations discussed in the paper for a fair comparison.
- CROSEL (Tian et al., 2024): We use the same base models and training procedures as mentioned above for PRODEN. We use 10 warm-up epochs using CC and $\lambda_{cr} = 2$. We abstain from using the data augmentations discussed in the paper for a fair comparison.
- CONF+Other method (our proposed approach): Our conformal candidate cleaning technique uses the same base models and training procedures as mentioned above for PRODEN. We use $R_{\text{warmup}} = 10$ warm-up epochs, a validation set size of 20 %, and $\alpha_r = \frac{1}{n'} \sum_{i=1}^{n'} \sum_{j \notin s_i} f_j(x_i)$. Otherwise, we use one of the given PLL classifiers for prediction-making.

We have implemented all approaches in PYTHON using the PYTORCH library. Running all experiments requires approximately three days on a machine with 48 cores and one NVIDIA GeForce RTX 4090. All our source code and data is available at `github.com/mathefuchs/pll-with-conformal-candidate-cleaning`.

E ADDITIONAL EXPERIMENTS

In addition to our cleaning method (CONF), we also benchmark the existing cleaning method CLSP (He et al., 2024) on all datasets in Table 4, 5, and 6, similar to the experiments in Section 5. Instead of training a ResNet-9 base model from scratch as done in Section 5.1, we use the pre-trained BLIP-2 model (Li et al., 2023) for the experiments in Table 5 below. We repeat all experiments five times and report means and standard deviations.

We observe that the CLSP models perform well on the image datasets (e.g., *cifar100*) but poorly on the real-world tabular PLL datasets shown in Table 4. We attribute this to the fact that CLSP relies on the latent representation of large-scale vision

Table 5: Average test-set accuracies (± std.) on the supervised datasets with added incorrect candidate labels. We benchmark our strategy (CONF) as well as the cleaning method CLSP combined with all existing methods. We use the pre-trained BLIP-2 model for all results in this table.

Method	*mnist*	*fmnist*	*kmnist*	*cifar10*	*cifar100*
PRODEN	87.21 (± 0.83)	71.18 (± 2.95)	59.31 (± 1.22)	99.07 (± 0.05)	90.51 (± 0.18)
CLSP+PRODEN	85.91 (± 2.42)	72.11 (± 2.81)	62.61 (± 1.00)	99.03 (± 0.07)	90.31 (± 0.13)
CONF+P. (no correction)	91.74 (± 0.34)	78.38 (± 0.50)	66.88 (± 0.76)	99.03 (± 0.04)	91.16 (± 0.10)
CONF+PRODEN	91.55 (± 0.23)	78.09 (± 0.33)	66.43 (± 0.38)	99.03 (± 0.04)	91.16 (± 0.10)
CC	86.29 (± 2.18)	66.19 (± 2.77)	58.29 (± 0.32)	99.07 (± 0.05)	73.43 (± 1.40)
CLSP+CC	85.46 (± 1.93)	71.37 (± 2.34)	61.37 (± 1.09)	99.03 (± 0.06)	89.00 (± 1.56)
CONF+CC	85.20 (± 4.16)	59.75 (± 2.68)	57.07 (± 0.66)	99.04 (± 0.03)	71.45 (± 0.94)
VALEN	78.91 (± 0.80)	66.53 (± 2.65)	58.48 (± 0.45)	92.17 (± 0.54)	67.24 (± 2.49)
CLSP+VALEN	84.72 (± 3.10)	68.84 (± 1.49)	60.76 (± 0.76)	98.17 (± 0.17)	84.53 (± 1.23)
CONF+VALEN	74.20 (± 21.99)	69.09 (± 2.71)	60.95 (± 2.59)	42.63 (± 19.92)	60.44 (± 1.91)
CAVL	71.11 (± 3.92)	59.85 (± 6.49)	48.15 (± 5.07)	41.78 (± 21.40)	31.95 (± 1.80)
CLSP+CAVL	83.72 (± 3.57)	67.38 (± 2.59)	62.06 (± 2.12)	87.34 (± 12.71)	68.02 (± 1.96)
CONF+CAVL	71.86 (± 4.57)	59.54 (± 6.62)	52.14 (± 3.89)	29.97 (± 15.73)	37.34 (± 2.59)
POP	87.08 (± 0.58)	72.30 (± 2.63)	60.63 (± 1.15)	99.06 (± 0.04)	90.50 (± 0.21)
CLSP+POP	85.43 (± 2.60)	72.05 (± 2.41)	62.49 (± 0.90)	99.04 (± 0.07)	90.37 (± 0.06)
CONF+POP	91.19 (± 0.29)	79.15 (± 1.23)	67.37 (± 0.28)	99.05 (± 0.04)	91.12 (± 0.09)
CROSEL	91.84 (± 0.44)	76.34 (± 1.21)	65.55 (± 0.81)	99.07 (± 0.02)	75.86 (± 2.26)
CLSP+CROSEL	91.70 (± 0.62)	74.42 (± 1.02)	67.93 (± 1.07)	99.08 (± 0.02)	88.80 (± 0.85)
CONF+CROSEL	91.85 (± 0.61)	77.31 (± 0.46)	64.73 (± 1.52)	99.07 (± 0.03)	77.26 (± 0.98)

Table 6: Number of significant differences aggregated from Table 4 and 6 using a paired t-test (level $5\,\%$).

Comparison vs. all others	Wins	Ties	Losses
PRODEN	27	37	8
CLSP+PRODEN	18	27	27
CONF+PRODEN (no correction)	41	25	6
CONF+PRODEN	50	20	2
CC	18	39	15
CLSP+CC	18	22	32
CONF+CC	22	30	20
VALEN	5	30	37
CLSP+VALEN	17	15	40
CONF+VALEN	5	24	43
CAVL	5	26	41
CLSP+CAVL	9	19	44
CONF+CAVL	4	26	42
POP	29	39	4
CLSP+POP	17	23	32
CONF+POP	49	22	1
CROSEL	30	33	9
CLSP+CROSEL	25	15	32
CONF+CROSEL	44	22	6

models. In contrast, our method CONF gives strong results on, both, real-world and image data. This hypothesis is supported by Table 6: The approaches CONF+PRODEN, CONF+POP, and CONF+CROSEL that combine the respective approaches with our candidate cleaning strategy win most frequently.