CCMN: A General Framework for Learning with Class-Conditional Multi-Label Noise

Ming-Kun Xie and Sheng-Jun Huang

Abstract—Class-conditional noise commonly exists in machine learning tasks, where the class label is corrupted with a probability depending on its ground-truth. Many research efforts have been made to improve the model robustness against the class-conditional noise. However, they typically focus on the single label case by assuming that only one label is corrupted. In real applications, an instance is usually associated with multiple labels, which could be corrupted simultaneously with their respective conditional probabilities. In this paper, we formalize this problem as a general framework of learning with Class-Conditional Multi-label Noise (CCMN for short). We establish two unbiased estimators with error bounds for solving the CCMN problems, and further prove that they are consistent with commonly used multi-label loss functions. Finally, a new method for partial multi-label learning is implemented with the unbiased estimator under the CCMN framework. Empirical studies on multiple datasets and various evaluation metrics validate the effectiveness of the proposed method.

Index Terms—Class-conditional noise, class-conditional multi-label noise, unbiased estimator, partial multi-label learning.

1 Introduction

In ordinary supervised classification problems, a common assumption is that the class labels of training data are always correct. However, in practice, the assumption hardly holds since the training examples are usually corrupted due to unavoidable reasons, such as measurement error, subjective labeling bias or human labeling error. Learning in presence of label noise has been a problem of theoretical as well as practical interest in machine learning communities [1]. In various applications, such as the image annotation [2] and text classification [3], many successful methods have been applied to train robust models against label noise.

In general, to deal with noise-corrupted data, it is crucial to discover the causes of label noise. A natural and simple formulation of label noise is that the labels are corrupted by a random noise process [4], which can be described by the random classification noise (RCN) framework [5]. RCN framework assumes that each label is flipped independently with a specific probability $\rho \in [0, \frac{1}{2})$. Despite the great impact that RCN framework has made, it is too simple to deal with practical tasks, where the cause of label noise may not follow a random process. In order to deal with such problems, in [6], [7], authors propose a class-conditional random label noise (CCN) framework, where the probability of a label flipping depends on its true label. Unfortunately, CCN framework only considers single label case and fail to deal with multiple noisy labels, where multiple labels assigned to one instance may be corrupted simultaneously.

In this paper, we extend CCN to a more general framework of Class Conditional Multi-label Noise (CCMN) to learn with multi-class and multi-label classification tasks. In CCMN framework, each instance is associated with multiple labels and each of class labels may be flipped with a probability ρ^j_{+1} or ρ^j_{-1}, which depends on its true label $y_j \in \{+1, -1\}$. It is worth noting that a great deal of real-world applications in weak-supervised setting [8] can be regarded as special cases of this framework. Examples include learning from partial-labeled data [9], [10], [11] or learning with missing labels [12], which we will discuss detailedly in following content. To the best of our knowledge, general theoretical results in this setting have not been developed before.

To tackle corrupted data with class-conditional multiple noisy labels, we derive a modified loss function for learning a multi-label classifier with risk minimization. Theoretically, we show that the empirical risk minimization with the modified loss functions can be in an unbiased fashion from independent and dependent perspectives. We then provide the estimation error bound for the two unbiased estimators and further show that learning with class-conditional multiple noisy labels can be multi-label consistent to two commonly used losses, i.e., hamming loss and ranking loss, respectively. Finally, CCMN can be regarded as a general framework of various weakly-supervised learning scenarios, such as partial label learning [9], partial multi-label learning [10] and weak label learning [12]. Among them, we take PML as an examples, and propose a new approach for partial multi-label learning with unbiased estimator. Comprehensive empirical studies demonstrate the effectiveness of the proposed methods.

Our main contributions are summarized as follows:

- A general framework of learning from class-conditional multi-label noise is proposed. Varied weakly-supervised learning scenarios can be cast under CCMN framework.
- We propose two unbiased estimators for solving CCMN problems in both independent and dependent fashion. These two estimators are proven to be multi-label con-

- *Ming-Kun Xie and Sheng-Jun Huang are with the MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, Nanjing University of Aeronautics and Astronautics, Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 211106, China. E-mail: {mkxie, huangsj}@nuaa.edu.cn*
- *Sheng-Jun Huang is the corresponding author.*

sistent to hamming loss and ranking loss, respectively.
- A novel approach for Partial Multi-label Learning with unbiased estimator (uPML) is proposed under the CCMN framework.

2 Related Work

There are a great deal of previous works aiming to learn a robust classification model in presence of label noise. Most of these methods only consider single label case and cannot tackle class-conditional multi-label noise.

There has been a long line of studies in machine learning community on random label noise. The earliest work by Angluin and Laird [5] first propose the *random classification noise* (RCN) model. In [13], authors propose a boosting-based method and empirically show its robustness to random label noise. A theoretical analysis proposed in [14] proves that any method based on convex surrogate losses is inherently ill-suited to random label noise. In [4], authors consider random and adversarial label noise and try to deal with label noise by utilizing a robust SVM. Besides, there are many other heuristic methods without theoretical justification, such as confidence weighted learning [15], AROW [16] and the NHERD algorithm [17]. Noise-tolerance property is proposed in [18] to examine whether a loss function is robust to label noise.

The first attempt to tackle class-conditional label noise is in [6] where authors propose a variant of SVM to deal with noisy labels with theoretical guarantee. In [7], authors propose unbiased estimators to solve class-conditional label noise in the binary classification setting. Based on the assumption that the class-conditional distributions may overlap, a method called mixture proportion estimation is proposed in [19] to estimate the maximal proportion of one distribution that is present in another. Furthermore, authors extend the method into multi-class setting in [20].

Thanks to the great development of deep learning, there are various methods raised for utilizing deep neural networks to handle noisy labels, such as label correction methods [21], [22], [23], loss correction methods [24], [25], sample reweighting methods [26], [27], [28] and robust loss methods [29], [30].

In this paper, we also employ the CCMN framework to solve partial multi-label learning problems. In order to deal with partial-labeled data, the most commonly used strategy is disambiguation, which recovers ground-truth labeling information for candidate labels. Some methods perform disambiguation strategy by estimating a confidence for each candidate label [10], [31], [32], [33]. Other methods utilize decomposition scheme [34] or adversarial training [35]. Besides, label compression technique is utilized to deal with the large label space in PML tasks [36]. However, aforementioned methods never consider the generation process of noisy labels in candidate label set, which is a essential information for solving PML problems. In [37], authors first consider modeling the relationship between noisy labels and feature representations. Some studies aim to extend the PML framework to novel settings, such as semi-supervised learning [38] and multi-view learning [39]. Nevertheless, all of these methods failed to prove the consistency of the proposed method.

3 Formulation of CCMN

Let $\boldsymbol{x} \in \mathcal{X}$ be a feature vector and $\boldsymbol{y} \in \mathcal{Y}$ be its corresponding label vector, where $\mathcal{X} \subset \mathbb{R}^d$ is the feature space and $\mathcal{Y} \subset \{-1, 1\}^q$ is the target space with q possible class labels. For notational convenience, the label y_j can be denoted by its index j. In the setting, $y_j = 1$ indicates the j-th label is a true label for instance $\boldsymbol{x}$; $y_j = -1$, otherwise. In this paper, we focus on the multi-label learning problem, where each instance may be assigned with more than one label, i.e., $\sum_{j=1}^{q} I(y_j = 1) \geq 1$ holds, where I is the indicator function. Let $S = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), ..., (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$ be the given training data set, drawn *i.i.d.* according to the true distribution $\mathcal{D}$. We also use $[q]$ to denote the integer set $\{1, ..., q\}$.

In this paper, for instance $\boldsymbol{x}$, its corresponding label $\boldsymbol{y}$ can be corrupted and may be flipped into $\tilde{\boldsymbol{y}}$ following a class-conditional multi-label noise model as follows:

$$
\begin{aligned}
p(\tilde{y}_j = -1 | y_j = +1) &= \rho_{+1}^j, \\
p(\tilde{y}_j = +1 | y_j = -1) &= \rho_{-1}^j, \\
\forall j \in [q], \rho_{+1}^j + \rho_{-1}^j &< 1.
\end{aligned}
$$

where ρ_{+1}^j and ρ_{-1}^j are noise rates for each class label and are assumed to be known to the learner. In section 6, we will discuss how to estimate true noise rates in detail.

After injecting random noise into original samples S, the observed data set $S_\rho = \{(\boldsymbol{x}_1, \tilde{\boldsymbol{y}}_1), ..., (\boldsymbol{x}_n, \tilde{\boldsymbol{y}}_n)\}$ are drawn *i.i.d.*, according to distribution $\mathcal{D}_\rho$. Our goal is to learn a prediction function $\boldsymbol{h} : \mathcal{X} \to \mathcal{Y}$ can accurately predict labels for any unseen instance. In general, it is not easy to learn $\boldsymbol{h}$ directly, and alternatively, one usually learns a real-valued decision function $\boldsymbol{f} : \mathcal{X} \to \mathbb{R}^K$. Note that, for each instance $\boldsymbol{x}$, even though its final prediction depends on $\boldsymbol{h}(\boldsymbol{x})$, we also call $\boldsymbol{f}$ itself the classifier. As mentioned before, CCMN can be used to solve multi-class and multi-label learning problems. In this paper, we focus on the multi-label classification task without loss of generality.

In general, CCMN framework has implications in varied real-world applications. In the following content, we discuss on two popular weakly-supervised learning scenarios, i.e., partial multi-label learning and weak label learning, which can be regarded as special cases of CCMN framework.

In PML problems, each instance is associated with a candidate label set Y_c, which contains multiple relevant labels and some other noisy labels. One intuitive strategy to solve the task is that treats all labels in Y_c as relevant labels and transforms Y_c into $\tilde{Y}$. Here, we use $\tilde{Y}$, since there may exist irrelevant labels in $\tilde{Y}$. Then, a new training set is obtained, where each instance is associated with the label set $\tilde{Y}$. Besides relevant labels, $\tilde{Y}$ is also injected into some irrelevant labels, which can be regarded as class-conditional multi-label noise, i.e., irrelevant labels are flipped into relevant labels with $\rho_{-1}^j > 0$ while $\rho_{+1}^j = 0$.

In weak label learning, also known as multi-label learning with missing labels, relevant labels of each instance are partially known. Specifically, each instance is associated with a relevant label set Y_{+1} while $\tilde{Y}_{-1}$ is the irrelevant label set. Here, we use $\tilde{Y}_{-1}$, since there may exist missing labels in $\tilde{Y}_{-1}$, i.e., relevant labels are missed. Similarly, one can treat all labels in $\tilde{Y}_{-1}$ as irrelevant labels. Therefore, besides irrelevant labels, $\tilde{Y}_{-1}$ is also injected into some relevant labels which

can be regarded as class-conditional multi-label noise with $\rho_{+1}^j > 0$ while $\rho_{-1}^j = 0$.

4 Learning with CCMN

In this section, we first provide some necessary preliminaries, and then derive two CCMN solvers for independent and dependent cases. In the independent case, we solve each binary classification task of the CCMN problem independently while in the dependent case, we solve the CCMN problem by considering label correlations.

4.1 Preliminaries

To derive our results for solving CCMN problems, we introduce some notations and the property of multi-label consistency.

There are many multi-label loss functions (also called evaluation metrics), such as *hamming loss*, *ranking loss*, *coverage* and *average precision* [40], etc. In this paper, we focus on two well known loss functions, i.e., hamming loss and ranking loss, and leave the discussion on other loss functions in the future work.

The hamming loss considers how many instance-label pairs are misclassified. Given the decision function $\boldsymbol{f}$ and prediction function F, the hamming loss can be defined by:

$$L_h(F(\boldsymbol{f}(\boldsymbol{x})), \boldsymbol{y}) = \frac{1}{q}\sum_{j=1}^{q} I(\hat{y}_j \neq y_j), \tag{1}$$

where $\hat{\boldsymbol{y}} = F(\boldsymbol{f}(\boldsymbol{x})) = [\hat{y}_1, ..., \hat{y}_q]$.

The ranking loss considers label pairs that are ordered reversely for an instance. Given a real-value decision function $\boldsymbol{f} = [f_1, f_2, ..., f_q]$, the ranking loss can be defined by:

$$L_r(\boldsymbol{f}, \boldsymbol{y}) = \sum_{1\leq j<k\leq q} I(y_j < y_k)\ell(j,k) + I(y_j > y_k)\ell(k,j), \tag{2}$$

where

$$\ell(j,k) = I(f_j > f_k) + \frac{1}{2}I(f_j = f_k).$$

The risk of $\boldsymbol{f}$ with respect to loss L is given by $R(\boldsymbol{f}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}[L(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y})]$ and the minimal risk (also called the Bayes risk) can be defined by $R^* = \inf_{\boldsymbol{f}} R(\boldsymbol{f})$. For an instance $\boldsymbol{x}$, the conditional risk of $\boldsymbol{f}$ can be defined as

$$l(\boldsymbol{p}, \boldsymbol{f}) = \sum_{\boldsymbol{y}\in\mathcal{Y}} p_{\boldsymbol{y}} L(\boldsymbol{f}, \boldsymbol{y}),$$

where $p_{\boldsymbol{y}} = [p(\boldsymbol{y}|\boldsymbol{x})]_{\boldsymbol{y}\in\mathcal{Y}}$ is a vector of conditional probability over $\boldsymbol{y} \in \mathcal{Y}$.

Note that the above mentioned two loss functions are discontinuous and computationally NP-hard, which makes the corresponding optimization problems hard to solve. In practice, a feasible solution is to consider alternatively a surrogate loss function $\mathcal{L}$ which can be optimized efficiently. We will give the specific definition of $\mathcal{L}$ in the next section. The $\mathcal{L}$-risk and Bayes $\mathcal{L}$-risk can be defined as $R_{\mathcal{L}}(\boldsymbol{f}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}[\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y})]$ and $R_{\mathcal{L}}^* = \inf_{\boldsymbol{f}} R_{\mathcal{L}}(\boldsymbol{f})$, respectively. Accordingly, we define the empirical $\mathcal{L}$-risk as $\widehat{R}_{\mathcal{L}}(\boldsymbol{f}) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{y}_i)$.

Furthermore, we define the conditional $\mathcal{L}$-risk of $\boldsymbol{f}$

$$W(\boldsymbol{p}, \boldsymbol{f}) = \sum_{\boldsymbol{y}\in\mathcal{Y}} p_{\boldsymbol{y}}\mathcal{L}(\boldsymbol{f}, \boldsymbol{y}),$$

and the conditional Bayes $\mathcal{L}$-risk

$$W^*(\boldsymbol{p}) = \inf_{\boldsymbol{f}} W(\boldsymbol{p}, \boldsymbol{f}).$$

Our goal is to learn a good classification model with the modified loss function $\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}), \tilde{\boldsymbol{y}})$ from noise-corrupted data by minimizing empirical $\tilde{\mathcal{L}}$-risk:

$$\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}\in\mathcal{F}} \widehat{R}_{\tilde{\mathcal{L}}}(\boldsymbol{f}),$$

where $\mathcal{F}$ is a function class.

4.2 Independent Case

In order to solve multi-label learning problems, the most straightforward method is to decompose the task into q independent binary classification problems [41], where the goal is to learn q functions, $\boldsymbol{f} = (f_1, f_2, ..., f_q)$. However, as mentioned before, it is difficult to directly optimize the hamming loss due to its discontinuity. Alternatively, we consider the following surrogate loss

$$\mathcal{L}_h(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y}) = \sum_{j=1}^{q}\phi(y_j f_j(\boldsymbol{x})), \tag{3}$$

where ϕ is a convex loss function. The common choices are least square loss $\phi(t) = (1-t)^2$ and hinge loss $\phi(t) = (1-t)_+$ in [42].

The modified loss function under class-conditional multi-label noise in the independent case can be defined as follows:

$$\tilde{\mathcal{L}}_h(\boldsymbol{f}(\boldsymbol{x}), \tilde{\boldsymbol{y}}) = \sum_{j=1}^{q}\tilde{\phi}(y_j f_j(\boldsymbol{x})), \tag{4}$$

where,

$$\tilde{\phi}(y_j f_j) = \kappa_j\left[(1-\rho_{-y_j})\phi(y_j f_j) - \rho_{y_j}\phi(-y_j f_j)\right].$$

Here, $\kappa_j = \frac{1}{1-\rho_{+1}^j-\rho_{-1}^j}$ is a constant and we omit the superscript j of ρ_{y_j}.

We extend the results in [7] to have the following lemma, which shows unbiasedness of the estimator defined by Eq.(4).

Lemma 1. *For any $y_j, \forall j \in [q]$, let $\phi(y_j f_j(\boldsymbol{x}))$ be any bounded loss function. Then, if $\tilde{\mathcal{L}}_h(\boldsymbol{f}, \tilde{\boldsymbol{y}})$ can be defined by Eq.(4), we have $\mathbb{E}_{\tilde{\boldsymbol{y}}}\left[\tilde{\mathcal{L}}_h(\boldsymbol{f}, \tilde{\boldsymbol{y}})\right] = \mathcal{L}_h(\boldsymbol{f}, \boldsymbol{y})$.*

Let $\boldsymbol{\sigma} = \{\sigma_1, ..., \sigma_n\}$ be n Rademacher variables with σ_i independently uniform random variable taking value in $\{-1, +1\}$. Then, the Rademacher complexity with respect to function class $\mathcal{F}$ and unbiased estimator $\tilde{\mathcal{L}}$ can be formulated as follows:

$$\mathcal{R}_n(\tilde{\mathcal{L}}\circ\mathcal{F}) = \mathbb{E}_{\boldsymbol{x},\tilde{\boldsymbol{y}},\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}_i), \tilde{\boldsymbol{y}}_i)\right].$$

Accordingly, the performance guarantee for the unbiased estimator can be derived as following theorems.

Theorem 1. *Let $\mu = \max_j \frac{1}{1-\rho_{-1}^j-\rho_{+1}^j}, \forall j \in [q]$. Then, for the loss function $\phi(\cdot)$ bounded by Θ, with probability at least $1-\delta$, we have*

$$R_{\mathcal{L}_h}(\hat{\boldsymbol{f}}) - \min_{\boldsymbol{f}\in\mathcal{F}} R_{\mathcal{L}_h}(\boldsymbol{f}) \leq 4qK_\rho\mathcal{R}_n(\mathcal{F}) + 2q\mu\Theta\sqrt{\frac{\ln\frac{1}{\delta}}{2n}},$$

where $\hat{\boldsymbol{f}}$ is trained by minimizing $\widehat{R}_{\tilde{\mathcal{L}}_h}(\boldsymbol{f})$ and K_ρ is the Lipschitz constant of $\tilde{\mathcal{L}}_h$.

Theorem 2. *If ϕ is convex function with $\phi'(0) < 0$, then learning from CCMN examples with the modified surrogate loss $\tilde{\mathcal{L}}_h$ defined by Eq.(4) is consistent w.r.t hamming loss, i.e., there exists a non-negative concave function ξ with $\xi(0) = 0$, such that*

$$R_{L_h}(\hat{\boldsymbol{f}}) - R^*_{L_h} \leq \xi(R_{\mathcal{L}_h}(\hat{\boldsymbol{f}}) - R^*_{\mathcal{L}_h}).$$

As shown in Theorem 1, the generalization error is dependent to the noise rates ρ_{-1} and ρ_{+1}. It is intuitive that smaller noise rates lead to a better generalization performance owing to a smaller μ. By combining Theorem 1 with Theorem 2, we obtain a performance guarantee for learning from class-conditional multiple noisy labels with respect to hamming loss. As $n \to \infty$, we have the consistency: if $R_{\mathcal{L}_h}(\hat{\boldsymbol{f}}_n) \to R^*_{\mathcal{L}_h}$ (as shown in Theorem 1), then $R_{L_h}(\hat{\boldsymbol{f}}_n) \to R^*_{L_h}$, since $\mathcal{R}_n(\mathcal{F}) \to 0$ for all parametric models with a bounded norm such as deep networks trained with weight decay [43].

4.3 Dependent Case

In multi-label learning problems, a common assumption is that there exist label correlations among labels [44], [45]. Therefore, it is fundamental to learn with class-conditional multi-label noise in a dependent fashion. The ranking loss considers the second-order label correlation and its surrogate loss is commonly defined as

$$\begin{aligned} &\mathcal{L}_r(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y}) \\ &= \sum_{1\leq j<k\leq q} I(y_j > y_k)\phi(f_{jk}) + I(y_j < y_k)\phi(f_{kj}) \\ &= \sum_{1\leq j<k\leq q} \phi(y_{jk}(f_j - f_k)), \end{aligned} \tag{5}$$

where $y_{jk} = \frac{y_j - y_k}{2}$ and $f_{jk} = f_j - f_k$.

If $y_j \neq y_k$, let

$$a = (1 - \rho^j_{-y_{jk}})(1 - \rho^k_{y_{jk}}), b = \rho^j_{y_{jk}}\rho^k_{-y_{jk}},$$

and if $y_j = y_k$, let

$$c = \rho^j_{y_j}(1 - \rho^k_{-y_k}), d = \rho^k_{y_k}(1 - \rho^j_{-y_j}).$$

Then, the modified loss function under class-conditional multi-label noise in the dependent case can be defined as

$$\tilde{\mathcal{L}}_r(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y}) = \sum_{1\leq j<k\leq q} \tilde{\phi}((f_j, f_k), (y_j, y_k)), \tag{6}$$

where $\tilde{\phi}((f_j, f_k), (y_j, y_k))$ can be abbreviated by $\tilde{\phi}(j, k)$,

$$\tilde{\phi}(j,k) = \begin{cases} \kappa_{jk}\left[a\phi(y_{jk}f_{jk}) + b\phi(-y_{jk}f_{jk})\right] & \text{if } y_j \neq y_k \\ -\kappa_{jk}\left[c\phi(-y_j f_{jk}) + d\phi(y_j f_{jk})\right] & \text{if } y_j = y_k \end{cases}$$

Here, $\kappa_{jk} = \frac{1}{(1-\rho^j_{+1}-\rho^j_{-1})(1-\rho^k_{+1}-\rho^k_{-1})}$ is a constant.

The unbiasedness of the estimator defined by Eq.(6) can be shown as following lemma.

Lemma 2. *For any $y_j, y_k, \forall j, k \in [q]$, let $\phi(\cdot)$ be any bounded loss function. Then, if $\tilde{\mathcal{L}}_r(\boldsymbol{f}, \tilde{\boldsymbol{y}})$ can be defined by Eq.(6), we have $\mathbb{E}_{\tilde{\boldsymbol{y}}}\left[\tilde{\mathcal{L}}_r(\boldsymbol{f}, \tilde{\boldsymbol{y}})\right] = \mathcal{L}_r(\boldsymbol{f}, \boldsymbol{y})$.*

Accordingly, the performance guarantee for the unbiased estimator can be derived as following theorems.

Theorem 3. *Let $\mu = \max_j \frac{1+|\rho^j_{-1}-\rho^j_{+1}|}{(1-\rho^j_{-1}-\rho^j_{+1})^2}, \forall j \in [q]$. Then, for the loss function $\phi(\cdot)$ bounded by Θ, with probability at least $1-\delta$, we have*

$$\begin{aligned} &R_{\mathcal{L}_r}(\hat{\boldsymbol{f}}) - \min_{\boldsymbol{f}\in\mathcal{F}} R_{\mathcal{L}_r}(\boldsymbol{f}) \\ &\leq 4q(q-1)K_\rho\mathcal{R}_n(\mathcal{F}) + 2q(q-1)\mu\Theta\sqrt{\frac{\ln\frac{1}{\delta}}{2n}}, \end{aligned}$$

where $\hat{\boldsymbol{f}}$ is trained by minimizing $\widehat{R}_{\tilde{\mathcal{L}}_r}(\boldsymbol{f})$ and K_ρ is the Lipschitz constant of $\tilde{\mathcal{L}}_r$.

Theorem 4. *If ϕ is a differential and non-increasing function with $\phi'(0) < 0$ and $\phi(t)+\phi(-t) = 2\phi(0)$, then learning from CCMN examples with the modified surrogate loss $\tilde{\mathcal{L}}_r$ defined by Eq.(6) is consistent w.r.t ranking loss, i.e., there exists a non-negative concave function ξ with $\xi(0) = 0$, such that*

$$R_{L_r}(\hat{\boldsymbol{f}}) - R^*_{L_r} \leq \xi(R_{\mathcal{L}_r}(\hat{\boldsymbol{f}}) - R^*_{\mathcal{L}_r}).$$

From Theorem 3, it can be observed that besides the noise rates ρ_{-1} and ρ_{+1}, the generalization error is also dependent to the difference between noise rates, i.e., $|\rho_{-1} - \rho_{+1}|$. Generally, a smaller difference also leads to a better generalization performance owing to a smaller μ. By combining Theorem 3 with Theorem 4, we obtain a performance guarantee for learning from class-conditional multiple noisy labels with respect to ranking loss. As $n \to \infty$, we have the consistency: if $R_{\mathcal{L}_r}(\hat{\boldsymbol{f}}_n) \to R^*_{\mathcal{L}_r}$ (as shown in Theorem 3), then $R_{L_r}(\hat{\boldsymbol{f}}_n) \to R^*_{L_r}$, since $\mathcal{R}_n(\mathcal{F}) \to 0$ for all parametric models with a bounded norm such as deep networks trained with weight decay [43].

Testing Phase. Regarding the classifier $\hat{\boldsymbol{f}}$ trained by minimizing the loss function $\tilde{\mathcal{L}}_h$ or $\mathcal{L}_h$, for each testing instance $\boldsymbol{x}_t$, we use the $\mathrm{sgn}(f_j(\boldsymbol{x}_t))$ to predict its labels, where $\mathrm{sgn}(a)$ is a function which outputs $+1$ if $a >= 0$; -1, otherwise.

However, regarding classifier $\boldsymbol{f}$ trained by minimizing the loss function $\tilde{\mathcal{L}}_r$ or $\mathcal{L}_r$, it is unreasonable to use 0 as a threshold directly. Instead, to perform predictions, in training phase, we introduce a *dummy* label $y_0 = 0$ for each instance, and then, in testing phase, the output of the classifier for label y_0 is used as the threshold to decide the label assignment for each instance.

Specifically, for instance $\boldsymbol{x}$, suppose that its corresponding label vector can be represented by $\boldsymbol{y} = [y_0, y_1, ..., y_q]$, where y_0 is indexed by 0. In the training phase, the cumulative loss $\tilde{\mathcal{L}}_0$ with respect to y_0 can be written as follows:

$$\tilde{\mathcal{L}}_0(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y}) = \sum_{j=1}^{q} \tilde{\phi}(f_{j0}(\boldsymbol{x}), y_{j0}), \tag{7}$$

where $f_{j0}(\boldsymbol{x}) = f_j(\boldsymbol{x}) - f_0(\boldsymbol{x})$ and $y_{j0} = y_j - y_0$. Note that $y_{j0} = y_j$, since we have $y_0 = 0$. In the situation, the formulation of loss $\tilde{\mathcal{L}}_0$ is similar to Eq.(4) and can be solved efficiently.

5 Partial Multi-Label Learning with CCMN

As discussed above, partial multi-label learning (PML) is a recently proposed framework, and is a typical CCMN task. In this section, we take PML as an example to examine the effectiveness of the proposed framework. Specifically, we propose a new approach for partial multi-label learning with unbiased estimator (uPML for short). In the PML setting, the noise rates satisfy $\forall j \in [q], p(\tilde{y}_j = +1|y_j = -1) = \rho^j, p(\tilde{y}_j = -1|y_j = +1) = 0$. Here, we omit the subscript of ρ^j for notational simplicity.

Based on the results in Section 4, in the independent case, the objective function of uPML $\tilde{\mathcal{L}}_h^{\mathrm{PML}}$ can be defined as follows:

$$\tilde{\mathcal{L}}_h^{\mathrm{PML}}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y}) = \sum_{j=1}^{q} \tilde{\phi}(y_j f_j(\boldsymbol{x})), \tag{8}$$

where,

$$\tilde{\phi}(y_j f_j(\boldsymbol{x})) = \begin{cases} \dfrac{\phi(-f_j(\boldsymbol{x})) - \rho^j \phi(f_j(x))}{1-\rho^j} & \text{if } y_j = -1 \\ \phi(f_j(x)) & \text{otherwise} \end{cases}$$

In the dependent case, the objective function of uPML $\tilde{\mathcal{L}}_r^{\mathrm{PML}}$ can be defined as follows:

$$\tilde{\mathcal{L}}_r^{\mathrm{PML}}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y}) = \sum_{1 \le j < k \le q} \tilde{\phi}(y_{jk} f_{jk}(\boldsymbol{x})), \tag{9}$$

where,

$$\tilde{\phi}(y_{jk} f_{jk}) = \begin{cases} \frac{\phi(f_{jk})}{1-\rho^k} & \text{if } y_{jk} = +1 \\ \frac{\phi(-f_{jk})}{1-\rho^j} & \text{if } y_{jk} = -1 \\ \frac{-\rho^j \phi(f_{jk}) - \rho^k \phi(-f_{jk})}{(1-\rho^j)(1-\rho^k)} & \text{if } y_j = y_k = -1 \\ 0 & \text{if } y_j = y_k = +1 \end{cases}$$

We derive the generalization error bound for the proposed uPML method, which can be regarded as special cases of Theorem 1 and 3.

Corollary 1. *For the least square loss $\phi(t) = (1-t)^2$, with probability at least $1-\delta$, we have*

$$R_{\mathcal{L}_h}(\hat{\boldsymbol{f}}) - \min_{\boldsymbol{f} \in \mathcal{F}} R_{\mathcal{L}_h}(\boldsymbol{f}) \le 4qK_\rho \mathcal{R}_n(\mathcal{F}) + \frac{8q}{1-\rho_{\min}} \sqrt{\frac{\ln \frac{1}{\delta}}{2n}},$$

where $\rho_{\min} = \min_j \rho^j$ and $\hat{\boldsymbol{f}}$ is trained by minimizing $\widehat{R}_{\tilde{\mathcal{L}}_h^{PML}}(f)$, and

$$\begin{aligned} & R_{\mathcal{L}_r}(\hat{\boldsymbol{f}}) - \min_{\boldsymbol{f} \in \mathcal{F}} R_{\mathcal{L}_r}(\boldsymbol{f}) \\ & \le 4q(q-1)K_\rho \mathcal{R}_n(\mathcal{F}) + 8q(q-1)\mu \sqrt{\frac{\ln \frac{1}{\delta}}{2n}}, \end{aligned}$$

where $\mu = \min_j \frac{1+\rho^j}{(1-\rho^j)^2}$ and $\hat{\boldsymbol{f}}$ is trained by minimizing $\widehat{R}_{\tilde{\mathcal{L}}_r^{PML}}(\boldsymbol{f})$.

It is easy to prove the consistency of the proposed uPML method based on the results derived in previous sections. To the best of our knowledge, uPML is the first consistent method for solving PML problems.

6 Noise Rate Estimation

In the aforementioned discussions, the noise rates ρ are assumed to be known. However, in many real-world scenarios, the true noise rates ρ would be unknown and need to be estimated. In this section, we provide an efficient method to estimate ρ, which is a multi-label extension of the recent noise estimator [24], [46].

When learning with class-conditional multiple noisy labels, the information of ground-truth labels is no longer accessible by the learner. Without any extra information, it is impractical to estimate the true noise rates. Based on the two assumptions, we propose to estimate the noise rates for CCMN as following theorem.

Theorem 5. *Assume $p(\boldsymbol{x}, \boldsymbol{y})$ is such that:*

(1) For each label $y_j, \forall j \in [q]$, there exist anchor points in the sense that

$$\exists \bar{\boldsymbol{x}}_l^j \in \mathcal{X}, p(\bar{\boldsymbol{x}}_l^j) > 0 \wedge p(y_j = l|\bar{\boldsymbol{x}}_l^j) = 1, l \in \{+1, -1\}.$$

(2) given sufficiently many corrupted training examples, $\boldsymbol{f}$ [1] is rich enough to model $p(\tilde{y}_j = l|\boldsymbol{x})$ accurately.

It follows that $\forall j \in [q], l \in \{-1, +1\}, \rho_l^j = p(\tilde{y}_j = l|\bar{\boldsymbol{x}}_l^j)$.

Proof. For any $j \in [q]$ and any $\boldsymbol{x} \in \mathcal{X}$, we have:

$$p(\tilde{y}_j = l|\boldsymbol{x}) = \rho_{-l}^j p(y_j = -l|\boldsymbol{x}) + (1-\rho_l^j) p(y_j = l|\boldsymbol{x}).$$

Given condition (1), when $\boldsymbol{x} = \bar{\boldsymbol{x}}_l^j$, we have $p(y_j = l|\bar{\boldsymbol{x}}_l^j) = 1$, which means $\rho_l^j = p(\tilde{y}_j = -l|\bar{\boldsymbol{x}}_l^j)$. □

Theorem 5 tells us that the noise rates can be estimated based on noisy label probability estimates, i.e., the outputs of sigmoid of a network trained on noisy labels. In particular, as shown in [24], it even dose not require these examples to have any clean labels at all. Specifically, $\forall j \in [q], l \in \{+1, -1\}$, the noise rate ρ_l^j can be approximated with two procedures:

$$\begin{aligned} \bar{\boldsymbol{x}}_j^l &= \arg\max_{\boldsymbol{x} \in S_\rho} \hat{p}(\tilde{y}_j = l|\boldsymbol{x}) \\ \hat{\rho}_l^j &= \hat{p}(\tilde{y}_j = -l|\bar{\boldsymbol{x}}_j), \end{aligned}$$

where $\hat{p}(\tilde{y}_j|\boldsymbol{x})$ is the sigmoid output and can be regarded as an estimation of class-conditional probabilities $p(\tilde{y}_j|\boldsymbol{x})$.

In practice, it would satisfy assumption (1) of Theorem 5 when the noisy training set S_ρ is large enough. However, it is more difficult to justify assumption (2) of Theorem 5, since the true class-conditional probability is unknown. In our experiments, it can be observed that the proposed methods obtain promising performance based on the estimated noise rates. This validates the effectiveness of the proposed estimator.

7 Proof

In this section, we provide detailed proofs of the theorems derived in previous sections.

1. Note that $\boldsymbol{f}(\cdot)$ passes through the *sigmoid* function $1/(1+e^{-f_j(\boldsymbol{x})})$, which can be interpreted as a vector approximating the class-conditional probabilities $p(\boldsymbol{y}|\boldsymbol{x})$

7.1 Proof of Lemma 1

It is straightforward to show the assertion of Lemma 1 as follows. Recall that here we use ρ_j and ρ_{-j} to denote ρ_{y_j} and ρ_{-y_j}, respectively, then, we have

$$\begin{aligned}
&\mathbb{E}_{\tilde{\boldsymbol{y}}}\left[\tilde{\mathcal{L}}_h(\boldsymbol{f},\tilde{\boldsymbol{y}})\right] \\
&= \sum_{j=1}^{q}(1-\rho_j)\tilde{\ell}(f_j,y_j)+\rho_j\tilde{\ell}(f_j,-y_j) \\
&= \sum_{j=1}^{q}\kappa_j\{(1-\rho_j)[(1-\rho_{-j})\ell(f_j,y_j)-\rho_j\ell(f_j,-y_j)] \\
&\qquad +\rho_j[(1-\rho_j)\ell(f_j,-y_j)-\rho_{-j}\ell(f_j,y_j)]\} \\
&= \sum_{j=1}^{K}\kappa_j(1-\rho_j-\rho_{-j})\ell(f_j,y_j)=\mathcal{L}_h(\boldsymbol{f},\boldsymbol{y}),
\end{aligned}$$

which completes proof. □

7.2 Proof of Theorem 2

The proof is mainly composed of the following two lemmas.

Lemma 3. *Let $\mathcal{R}_n(\tilde{\mathcal{L}}_h\circ\mathcal{F})$ be the Rademacher complexity of $\tilde{\mathcal{L}}_h$ and $\mathcal{F}$ over S_ρ with n training points drawn from $\mathcal{D}_\rho$, which can be defined as*

$$\mathcal{R}_n(\tilde{\mathcal{L}}_h\circ\mathcal{F})=\mathbb{E}_{S_\rho}\mathbb{E}_\sigma\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\tilde{\mathcal{L}}_h(\boldsymbol{f}(\boldsymbol{x}_i),\tilde{\boldsymbol{y}}_i)\right].$$

Then,

$$\mathcal{R}_n(\tilde{\mathcal{L}}_h\circ\mathcal{F})\le qK_\rho\mathcal{R}_n(\mathcal{F}),$$

where K_ρ is the Lipschitz constant of $\tilde{\mathcal{L}}_h$.

Proof. Recall that $\tilde{\mathcal{L}}_h(\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{y})=\sum_{j=1}^{q}\tilde{\phi}(y_jf_j(\boldsymbol{x}))$, then, we have

$$\begin{aligned}
&\mathcal{R}_n(\tilde{\mathcal{L}}_h\circ\mathcal{F}) \\
&= \mathbb{E}_{S_\rho}\mathbb{E}_\sigma\left[\sup_{f_1,\ldots,f_q\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\sum_{j=1}^{q}\tilde{\phi}(y_j^{(i)}f_j(\boldsymbol{x}_i))\right] \\
&= \mathbb{E}_{\mathcal{X}}\mathbb{E}_\sigma\left[\sup_{f_1,\ldots,f_q\in\mathcal{F}}\frac{1}{n}\sum_{\boldsymbol{x}_i\in\mathcal{X}}\sigma_i\sum_{j=1}^{q}\tilde{\phi}(y_j^{(i)}f_j(\boldsymbol{x}_i))\right] \\
&\le \sum_{j=1}^{q}\mathbb{E}_{\mathcal{X}}\mathbb{E}_\sigma\left[\sup_{f_j\in\mathcal{F}}\frac{1}{n}\sum_{\boldsymbol{x}_i\in\mathcal{X}}\sigma_i\tilde{\phi}(y_j^{(i)}f_j(\boldsymbol{x}_i))\right] \\
&= q\mathcal{R}_n(\tilde{\phi}\circ\mathcal{F}).
\end{aligned}$$

Sequentially, according to Talagrand's contraction lemma [47], we have

$$\begin{aligned}
\mathcal{R}_n(\tilde{\mathcal{L}}_h\circ\mathcal{F}) &\le q\mathcal{R}_n(\tilde{\phi}\circ\mathcal{F}) \\
&\le qK_\rho\mathcal{R}_n(\mathcal{F}),
\end{aligned}$$

which completes the proof. □

Without loss of generality, assume that $\forall j\in[q],\mu=\max_j\frac{1}{1-\rho_{-1}^j-\rho_{+1}^j}$.

Lemma 4. *For any $\delta>0$, with probability at least $1-\delta$,*

$$\max_{\boldsymbol{f}\in\mathcal{F}}\left|\widehat{R}_{\tilde{\mathcal{L}}_h}(\boldsymbol{f})-R_{\tilde{\mathcal{L}}_h}(\boldsymbol{f})\right|\le 2qK_\rho\mathcal{R}_n(\mathcal{F})+q\mu\Theta\sqrt{\frac{\ln\frac{2}{\delta}}{2n}}.$$

Proof. Since both two directions can be proved in the same way, we consider one single direction $\sup_{f_1,\ldots,f_q\in\mathcal{F}}(\widehat{R}_{\tilde{\mathcal{L}}}(\boldsymbol{f})-R_{\tilde{\mathcal{L}}}(\boldsymbol{f}))$. Note that the change in $\boldsymbol{x}_i$ leads to a perturbation of at most $\frac{q\mu\Theta}{n}$ by replacing a single point $(\boldsymbol{x}_i,\tilde{\boldsymbol{y}}_i)$ with $(\boldsymbol{x}_i',\tilde{\boldsymbol{y}}_i')$, since the change in any $\tilde{y}_j^{(i)}$ leads to a perturbation as Eq.(10). By using McDiarmid's inequality [48] to the single-direction uniform deviation $\sup_{f_1,\ldots,f_k\in\mathcal{F}}(\widehat{R}_{\tilde{\mathcal{L}}}(\boldsymbol{f})-R_{\tilde{\mathcal{L}}}(\boldsymbol{f}))$, we have

$$\begin{aligned}
&p\left\{\sup_{\boldsymbol{f}\in\mathcal{F}}(\widehat{R}_{\tilde{\mathcal{L}}}(\boldsymbol{f})-R_{\tilde{\mathcal{L}}}(\boldsymbol{f}))-\mathbb{E}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}(\widehat{R}_{\tilde{\mathcal{L}}}(\boldsymbol{f})-R_{\tilde{\mathcal{L}}}(\boldsymbol{f}))\right]\ge\epsilon\right\} \\
&\le\exp\left(-\frac{2\epsilon^2}{n(\frac{q\mu\Theta}{n})^2}\right),
\end{aligned}$$

or equivalently, with probability at least $1-\delta$,

$$\begin{aligned}
&\sup_{\boldsymbol{f}\in\mathcal{F}}(\widehat{R}_{\tilde{\mathcal{L}}}(\boldsymbol{f})-R_{\tilde{\mathcal{L}}}(\boldsymbol{f})) \\
&\le\mathbb{E}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}(\widehat{R}_{\tilde{\mathcal{L}}}(\boldsymbol{f})-R_{\tilde{\mathcal{L}}}(\boldsymbol{f}))\right]+q\mu\Theta\sqrt{\frac{\ln\frac{1}{\delta}}{2n}}.
\end{aligned}$$

According to [48], it is straightforward to show that

$$\mathbb{E}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}(\widehat{R}_{\tilde{\mathcal{L}}}(\boldsymbol{f})-R_{\tilde{\mathcal{L}}}(\boldsymbol{f}))\right]\le 2\mathcal{R}_n(\tilde{\mathcal{L}}\circ\mathcal{F}).$$

With the lemma 3, we complete the proof. □

Based on the Lemma 4, with $\boldsymbol{f}^*=\operatorname{argmin}_{\boldsymbol{f}\in\mathcal{F}}R_{\mathcal{L},\mathcal{D}}(\boldsymbol{f})$, it is obvious to prove the generalization error bound as follows:

$$\begin{aligned}
&R_{\mathcal{L},\mathcal{D}}(\hat{\boldsymbol{f}})-R_{\mathcal{L},\mathcal{D}}(\boldsymbol{f}^*) \\
&= R_{\tilde{\mathcal{L}},\mathcal{D}_\rho}(\hat{\boldsymbol{f}})-R_{\tilde{\mathcal{L}},\mathcal{D}_\rho}(\boldsymbol{f}^*) \\
&= \left(\widehat{R}_{\tilde{\mathcal{L}}}(\hat{\boldsymbol{f}})-\widehat{R}_{\tilde{\mathcal{L}}}(\boldsymbol{f}^*)\right)+\left(R_{\tilde{\mathcal{L}},\mathcal{D}_\rho}(\hat{\boldsymbol{f}})-\widehat{R}_{\tilde{\mathcal{L}}}(\hat{\boldsymbol{f}})\right) \\
&\quad+\left(\widehat{R}_{\tilde{\mathcal{L}}}(\boldsymbol{f}^*)-R_{\tilde{\mathcal{L}},\mathcal{D}_\rho}(\boldsymbol{f}^*)\right) \\
&\le 0+2\max_{\boldsymbol{f}\in\mathcal{F}}\left|\widehat{R}_{\tilde{\mathcal{L}}}(\boldsymbol{f})-R_{\tilde{\mathcal{L}},\mathcal{D}_\rho}(\boldsymbol{f})\right|.
\end{aligned}$$

The fist equation holds due to the unbiasedness of the estimator and for the last line, we used the fact $\widehat{R}_{\tilde{\mathcal{L}}}(\hat{\boldsymbol{f}})\le\widehat{R}_{\tilde{\mathcal{L}}}(\boldsymbol{f}^*)$ by the definition of $\hat{\boldsymbol{f}}$. □

7.3 Proof for Theorem 3

Before providing the proof, the definition of multi-label consistency can be formulated as follows.

Definition 1. *[49] Given a below-bounded surrogate loss $\mathcal{L}$, where $\mathcal{L}(\cdot,\boldsymbol{y})$ is continuous for every $\boldsymbol{y}\in\mathcal{Y}$, $\mathcal{L}$ is said to be multi-label consistent w.r.t. the loss L if it holds, for every $\boldsymbol{p}$, that*

$$W^*(\boldsymbol{p})<\inf_{\boldsymbol{f}}\{W(\boldsymbol{p},\boldsymbol{f}):\boldsymbol{f}\notin\mathcal{A}(\boldsymbol{p})\},$$

where $\mathcal{A}(\boldsymbol{p})=\{\boldsymbol{f}:l(\boldsymbol{p},\boldsymbol{f})=\inf_{\boldsymbol{f}'}l(\boldsymbol{p},\boldsymbol{f}')\}$ is the set of Bayes decision functions.

Based on the definition, the following theorem can be further established.

Theorem 6. *[49] The surrogate loss $\mathcal{L}$ is multi-label consistent w.r.t. the loss L if and only if it holds for any sequence $\{\boldsymbol{f}_n\}_{n\ge1}$ that*

$$\textit{if}\quad R_{\mathcal{L}}(\boldsymbol{f}_n)\to R_{\mathcal{L}}^*\quad\textit{then}\quad R(\boldsymbol{f}_n)\to R^*.$$

$$\frac{1}{n}\left|\frac{(1-\rho_{-1}^{j})\phi(f_j)-\rho_{+1}^{j}\phi(-f_j)-[(1-\rho_{+1}^{j})\phi(-f_j)-\rho_{-1}^{j}\phi(f_j)]}{1-\rho_{-1}^{j}-\rho_{+1}^{j}}\right|=\frac{1}{n}\left|\frac{\phi(f_j)-\phi(-f_j)}{1-\rho_{-1}^{j}-\rho_{+1}^{j}}\right|\leq\frac{\mu\Theta}{n}. \tag{10}$$

$$\begin{bmatrix}\alpha_{+1}^{j}\alpha_{-1}^{k} & \rho_{+1}^{j}\rho_{-1}^{k} & \rho_{+1}^{j}\alpha_{-1}^{k} & \rho_{-1}^{k}\alpha_{+1}^{j}\\ \rho_{-1}^{j}\rho_{+1}^{k} & \alpha_{-1}^{j}\alpha_{+1}^{k} & \rho_{-1}^{j}\alpha_{+1}^{k} & \rho_{-1}^{j}\alpha_{+1}^{k}\\ \alpha_{+1}^{j}\rho_{+1}^{k} & \rho_{+1}^{j}\alpha_{+1}^{k} & \rho_{+1}^{j}\rho_{+1}^{k} & \alpha_{+1}^{j}\alpha_{+1}^{k}\\ \rho_{-1}^{j}\alpha_{-1}^{k} & \alpha_{-1}^{j}\rho_{-1}^{k} & \alpha_{-1}^{j}\alpha_{-1}^{k} & \rho_{-1}^{j}\rho_{-1}^{k}\end{bmatrix}\begin{bmatrix}\tilde{\phi}((f_j,f_k),(+1,-1))\\ \tilde{\phi}((f_j,f_k),(-1,+1))\\ \tilde{\phi}((f_j,f_k),(+1,+1))\\ \tilde{\phi}((f_j,f_k),(-1,-1))\end{bmatrix}=\begin{bmatrix}\phi(f_j-f_k)\\ \phi(f_k-f_j)\\ \phi(0)\\ \phi(0)\end{bmatrix}. \tag{11}$$

where $\alpha_{+1}^{j}=1-\rho_{+1}^{j}$ and $\alpha_{-1}^{j}=1-\rho_{-1}^{j}$.

$$\begin{aligned}
\tilde{\phi}((f_j,f_k),(+1,-1)) &= \kappa_{jk}\left[(1-\rho_{-1}^{j})(1-\rho_{+1}^{k})\phi(f_{jk})+\rho_{+1}^{j}\rho_{-1}^{k}\phi(-f_{jk})\right]\\
\tilde{\phi}((f_j,f_k),(-1,+1)) &= \kappa_{jk}\left[(1-\rho_{+1}^{j})(1-\rho_{-1}^{k})\phi(-f_{jk})+\rho_{-1}^{j}\rho_{+1}^{k}\phi(f_{jk})\right]\\
\tilde{\phi}((f_j,f_k),(+1,+1)) &= -\kappa_{jk}\left[\rho_{+1}^{j}(1-\rho_{-1}^{k})\phi(-f_{jk})+\rho_{+1}^{k}(1-\rho_{-1}^{j})\phi(f_{jk})\right]\\
\tilde{\phi}((f_j,f_k),(-1,-1)) &= -\kappa_{jk}\left[\rho_{-1}^{j}(1-\rho_{+1}^{k})\phi(f_{jk})+\rho_{-1}^{k}(1-\rho_{+1}^{j})\phi(-f_{jk})\right].
\end{aligned} \tag{12}$$

This theorem tells us that the multi-label consistency is a necessary and sufficient condition for the convergence of $\mathcal{L}$-risk to the Bayes $\mathcal{L}$-risk, implying $R(\boldsymbol{f})\to R^*$. The proof of Theorem 3 is presented as follows.

With respect to $\tilde{\mathcal{L}}_h$, the conditional surrogate loss can be defined as

$$\begin{aligned}
\widetilde{W}(\boldsymbol{p},\boldsymbol{f}) &= \mathbb{E}_{\tilde{\boldsymbol{y}}}[\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}),\tilde{\boldsymbol{y}})]=\sum_{\boldsymbol{y}\in\mathcal{Y}}p_{\boldsymbol{y}}\mathbb{E}_{\tilde{\boldsymbol{y}}|\boldsymbol{y}}[\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}),\tilde{\boldsymbol{y}})]\\
&= \sum_{\boldsymbol{y}\in\mathcal{Y}}p_{\boldsymbol{y}}\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{y})=\sum_{j=1}^{q}\sum_{\boldsymbol{y}\in\mathcal{Y}}p_{\boldsymbol{y}}\phi(y_jf_j(\boldsymbol{x}))\\
&= W(\boldsymbol{p},\boldsymbol{f})=\sum_{j=1}^{q}p_j^{+}\phi(f_j(\boldsymbol{x}))+p_j^{-}\phi(-f_j(\boldsymbol{x})).
\end{aligned}$$

where the second equality is based on the law of total probability. Here, $p_j^{+}=\sum_{\boldsymbol{y}:y_j=+1}p_{\boldsymbol{y}}$ and $p_j^{-}=\sum_{\boldsymbol{y}:y_j=-1}p_{\boldsymbol{y}}$. Accordingly, we have

$$\widetilde{W}^*(\boldsymbol{p})=W^*(\boldsymbol{p})=\inf_{\boldsymbol{f}}W(\boldsymbol{p},\boldsymbol{f})=\sum_{j=1}^{q}\inf_{f_j}W_j(p_j^{+},f_j),$$

where $W_j(p_j^{+},f_j)=p_j^{+}\phi(f_j)+p_j^{-}\phi(-f_j)$. This yields that minimizing $\widetilde{W}(\boldsymbol{p},\boldsymbol{f})$ is equivalent to minimizing $W_j(p_j^{+},f_j)$ for every $1\leq j\leq q$. The consistency for binary classification has been well-studied [50], [51]. Based on their results, it is easy to prove that learning from CCMN data with $\tilde{\mathcal{L}}_h$ is consistent with respect to hamming loss. Therefore, based on Corollary 25 in [52], we have

$$R_{L_h}(\hat{\boldsymbol{f}})-R_{L_h}^{*}\leq\xi(R_{\mathcal{L}_h}(\hat{\boldsymbol{f}})-R_{\mathcal{L}_h}^{*}),$$

where ξ is a non-negative concave function with $\xi(0)=0$. □

7.4 Proof for Lemma 2

For each instance, regarding each pair of noisy labels $(\tilde{y}_j,\tilde{y}_k)$, there may exist four cases as follows: $(\tilde{y}_j=+1,\tilde{y}_k=-1)$, $(\tilde{y}_j=-1,\tilde{y}_k=+1)$, $(\tilde{y}_j=+1,\tilde{y}_k=+1)$ and $(\tilde{y}_j=-1,\tilde{y}_k=-1)$. By considering these four cases separately, we have four linear equations as presented in Eq.(11). Solving these four equations for $\tilde{\phi}(j,k)$ gives the solution as shown in Eq.(12). With simple computation for these four equations, we obtain the unbiased estimator defined in Eq.(6), which completes the proof. □

7.5 Proof for Theorem 4

We first propose the following two lemmas, which are useful for proving Theorem 4.

Lemma 5. *Let $\mathcal{R}_n(\tilde{\mathcal{L}}_r\circ\mathcal{F})$ be the Rademacher complexity of $\tilde{\mathcal{L}}_r$ and $\mathcal{F}$ over S_ρ with n training data drawn from $\mathcal{D}_\rho$, which can be defined as*

$$\mathcal{R}_n(\tilde{\mathcal{L}}_r\circ\mathcal{F})=\mathbb{E}_{S_\rho}\mathbb{E}_\sigma\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\tilde{\mathcal{L}}_r(\boldsymbol{f}(\boldsymbol{x}_i),\tilde{\boldsymbol{y}}_i)\right].$$

Then,

$$\mathcal{R}_n(\tilde{\mathcal{L}}_r\circ\mathcal{F})\leq 2q(q-1)K_\rho\mathcal{R}_n(\mathcal{F}),$$

where K_ρ is the Lipschitz constant of $\tilde{\mathcal{L}}_r$.

Proof. Recalling $\tilde{\mathcal{L}}_r(\boldsymbol{f}(\boldsymbol{x}),\tilde{\boldsymbol{y}})=\sum_{j,k}\tilde{\phi}((f_j,f_k),(y_j,y_k))$, we have

$$\begin{aligned}
&\mathcal{R}_n(\tilde{\mathcal{L}}_r\circ\mathcal{F})\\
&=\mathbb{E}_{S_\rho}\mathbb{E}_\sigma\left[\sup_{f_1,\ldots,f_q\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\sum_{1\leq j<k\leq q}\tilde{\phi}((f_j,f_k),(y_j,y_k))\right]\\
&=\mathbb{E}_{\mathcal{X}}\mathbb{E}_\sigma\left[\sup_{f_1,\ldots,f_q\in\mathcal{F}}\frac{1}{n}\sum_{\boldsymbol{x}_i\in\mathcal{X}}\sigma_i\sum_{1\leq j<k\leq q}\tilde{\phi}((f_j,f_k),(y_j,y_k))\right]\\
&\leq\sum_{1\leq j<k\leq q}\mathbb{E}_{\mathcal{X}}\mathbb{E}_\sigma\left[\sup_{f_1,\ldots,f_q\in\mathcal{F}}\frac{1}{n}\sum_{\boldsymbol{x}_i\in\mathcal{X}}\sigma_i\tilde{\phi}((f_j,f_k),(y_j,y_k))\right].
\end{aligned} \tag{13}$$

Sequentially, let (y, y') be the current label pair to be cumulated, then, according to Talagrand's contraction lemma [47], we have

$$\begin{aligned}
&\mathbb{E}_{\mathcal{X}}\mathbb{E}_{\sigma}\left[\sup_{f_y,f_{y'}\in\mathcal{F}}\frac{1}{n}\sum_{\boldsymbol{x}_i\in\mathcal{X}}\sigma_i\tilde{\phi}\left((f_y,f_{y'}),(y,y')\right)\right]\\
&\leq K_\rho\mathbb{E}_{\mathcal{X}}\mathbb{E}_{\sigma}\left[\sup_{f_y,f_{y'}\in\mathcal{F}}\frac{1}{n}\sum_{\boldsymbol{x}_i\in\mathcal{X}}\sigma_i(f_y(\boldsymbol{x}_i)-f_{y'}(\boldsymbol{x}_i))\right]\\
&\leq K_\rho\mathbb{E}_{\mathcal{X}}\mathbb{E}_{\sigma}\left[\sup_{f_y\in\mathcal{F}}\frac{1}{n}\sum_{\boldsymbol{x}_i\in\mathcal{X}}\sigma_i f_y(\boldsymbol{x}_i)\right]\\
&\quad+K_\rho\mathbb{E}_{\mathcal{X}}\mathbb{E}_{\sigma}\left[\sup_{f_{y'}\in\mathcal{F}}\frac{1}{n}\sum_{\boldsymbol{x}_i\in\mathcal{X}}\sigma_i f_{y'}(\boldsymbol{x}_i)\right]\\
&=2K_\rho\mathcal{R}_n(\mathcal{F}),
\end{aligned}$$

where f_y represent the classifier corresponding class label y. Then, by combining with Eq.(13), it is easy to prove that $\mathcal{R}_n(\tilde{\mathcal{L}}_r\circ\mathcal{F})\leq 2q(q-1)K_\rho\mathcal{R}_n(\mathcal{F})$. □

Without loss of generality, assume that $\forall j\in[q], \mu = \max_j \frac{1-\min(\rho^j_{-1}-\rho^j_{+1},\rho^j_{+1}-\rho^j_{-1})}{(1-\rho^j_{-1}-\rho^j_{+1})^2}$.

Lemma 6. *For any $\delta>0$, with probability at least $1-\delta$,*

$$\begin{aligned}
&\max_{\boldsymbol{f}\in\mathcal{F}}\left|\widehat{R}_{\tilde{\mathcal{L}}}(\boldsymbol{f})-\mathcal{R}_{\tilde{\mathcal{L}}}(\boldsymbol{f})\right|\\
&\leq 2q(q-1)K_\rho\mathcal{R}_n(\mathcal{F})+q(q-1)\mu\Theta\sqrt{\frac{\ln\frac{2}{\delta}}{2n}}.
\end{aligned}$$

We omit the proof, since it can be proved similarly to Lemma 4. □

Finally, similar to theorem 2, based on lemma 5 and 6, it is easy to obtain theorem 4. □

7.6 Proof for Theorem 5

For notational simplicity, we introduce the following notation:

$$\Delta_{j,k}=\sum_{\boldsymbol{y}:y_j=-1,y_k=+1}p_{\boldsymbol{y}}.$$

With respect to $\tilde{\mathcal{L}}$, the conditional surrogate loss can be defined as

$$\begin{aligned}
\widetilde{W}(\boldsymbol{p},\boldsymbol{f})&=\mathbb{E}_{\tilde{\boldsymbol{y}}}[\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}),\tilde{\boldsymbol{y}})]=\sum_{\boldsymbol{y}\in\mathcal{Y}}p_{\boldsymbol{y}}\mathbb{E}_{\tilde{\boldsymbol{y}}|\boldsymbol{y}}[\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}),\tilde{\boldsymbol{y}})] \quad (14)\\
&=\sum_{\boldsymbol{y}\in\mathcal{Y}}p_{\boldsymbol{y}}\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{y})=W(\boldsymbol{p},\boldsymbol{f})\\
&=\sum_{1\leq j<k\leq q}\Delta_{j,k}\phi(f_k-f_j)+\Delta_{k,j}\phi(f_j-f_k),
\end{aligned}$$

where the second equality is based on the law of total probability.

Following Theorem 10 in [49], if ϕ is a differential and non-increasing function with $\phi'(0)<0$ and $\phi(t)+\phi(-t)=2\phi(0)$, it suffices to prove that $f_j>f_k$ if $\Delta_{j,k}<\Delta_{k,j}$ for every $\boldsymbol{f}$ such that $W^*(\boldsymbol{p})=W(\boldsymbol{p},\boldsymbol{f})=\widetilde{W}(\boldsymbol{p},\boldsymbol{f})$, where $W^*(\boldsymbol{p})=\inf_{\boldsymbol{f}}W(\boldsymbol{p},\boldsymbol{f})$ is the conditional Bayes $\mathcal{L}$-risk. Therefore, by minimizing $\widetilde{W}(\boldsymbol{p},\boldsymbol{f})$, we obtain the Bayes decision function $\boldsymbol{f}^*$, which proves $\tilde{\mathcal{L}}_r$ is consistent to ranking loss. Accordingly, based on Corollary 25 in [52], we have

$$R_{L_r}(\hat{\boldsymbol{f}})-R^*_{L_r}\leq\xi(R_{\mathcal{L}_r}(\hat{\boldsymbol{f}})-R^*_{\mathcal{L}_r}),$$

where ξ is a non-negative concave function with $\xi(0)=0$. □

8 Experiment

The experiments for CCMN data are first reported, followed by the experiments for PML data.

8.1 Experimental Settings

Datasets We evaluate our method on eight benchmark multi-label datasets: music_emotion, music_style, mirflickr, tmc2007 [2], Multi-MNIST [3] [53], Multi-Kuzushiji-MNIST (Multi-KMNIST for short), Multi-Fashion-MNIST [4] (Multi-FMNIST for short), and VOC2007 [5] [54]. It is noteworthy that the first three datasets, i.e., music_emotion, music_style and mirflickr are real-world PML datasets [32]. For these datasets, candidate are first collected from web users and then these candidate labels are further examined by human labelers to specify the ground-truth labels. For tmc2007, we use its shorter version which contains 28,596 instances and 500 features for each instance. For three Multi-MNIST-style datasets, each dataset contains 10,000 images. VOC2007 (VOC for short) contains 9,963 images for 20 object categories, which are divided into train, val and test sets. Following [55], [56], we use the *trainval* set to train the models, and evaluate the performance on the test set. For each dataset except for VOC, we randomly sample 60% examples for training and 40% examples for testing.

Metrics We evaluate the performance of the proposed method based on multiple standard multi-label criterion: hamming loss, ranking loss, coverage and average precision. For hamming loss, ranking loss, coverage, the smaller value, the better performance; for average precision, the larger value, the better performance. The detail of these criterion can be found in [40].

Implementation For experiments on all datasets except for VOC, we train a linear model by using Adam [57] optimizer with learning rate chosen from $\{0.01, 0.001\}$. We added an ℓ_2-regularization term, with the regularization parameter of 0.0001. For experiments on VOC, we use an Alexnet [58] pre-trained with the ILSVRC2012 dataset on Pytorch platform [59]. The Alexnet is trained by using stochastic gradient descent (SGD) with learning rate chosen from $\{0.01, 0.001\}$. An ℓ_2-regularization term is added with the regularization parameter of 0.0001. The batch size for all datasets is set as 200 except for tmc and mirflickr, where the batch size is set as 400. All the experiments are conducted on GeForce RTX 2080 GPUs

2. See http://mulan.sourceforge.net/datasets-mlc.html for tmc2007.

3. See https://github.com/shaohua0116/MultiDigitMNIST for Multi-MNIST.

4. Similar to Multi-MNIST, we construct Multi-Kuzushiji-MNIST and Multi-Fashion-MNIST for two commonly used datasets Kuzushiji-MNIST and Fashion-MNIST, repsectively.

5. See http://host.robots.ox.ac.uk/pascal/VOC/voc2007/ for VOC2007.

TABLE 1
Comparison results between Ub-HL, Ub-RL and their baselines with diverse loss functions using linear models, where •/◦ indicates whether the proposed method is significantly superior/inferior to the comparing methods via paired t-test (at 0.05 significance level).

Data	music_emotion	music_style	mirflickr	tmc2007	Multi-MNIST	Multi-KMNIST	Multi-FMNIST	VOC
Hamming loss (the smaller, the better)								
Ub-HL/Hinge	.207 ± .003•	.125 ± .003•	**.103 ± .007**	**.059 ± .002**	.258 ± .008	.256 ± .007•	.199 ± .005	**.063 ± .002**
Ub-HL/Square	.201 ± .002	**.122 ± .002**	.104 ± .009	.060 ± .001•	.258 ± .006	.256 ± .007•	.205 ± .005•	.092 ± .006•
Ub-RL/Sigmoid	**.199 ± .004**	.123 ± .001	.111 ± .017	.062 ± .001•	**.256 ± .007**	**.249 ± .006**	**.198 ± .007**	.077 ± .000•
Ub-RL/Hinge	.203 ± .001	.123 ± .002•	.107 ± .010•	.061 ± .002•	.257 ± .007	.251 ± .006•	.203 ± .005	.068 ± .001•
Ub-RL/Square	.203 ± .002	.124 ± .003•	.115 ± .013•	.063 ± .002•	.259 ± .005	.259 ± .002•	.218 ± .005•	.092 ± .015•
B-HL/Hinge	.277 ± .028•	.189 ± .043•	.151 ± .021•	.124 ± .020•	.278 ± .011•	.282 ± .009•	.235 ± .009•	.172 ± .017•
B-HL/Square	.261 ± .022•	.171 ± .030•	.130 ± .015•	.112 ± .017•	.274 ± .008•	.280 ± .010•	.232 ± .010•	.235 ± .021•
B-RL/Sigmoid	.280 ± .008•	.188 ± .045•	.167 ± .025•	.130 ± .021•	.275 ± .010•	.281 ± .012•	.229 ± .006•	.098 ± .007•
B-RL/Hinge	.263 ± .023•	.166 ± .026•	.133 ± .015•	.112 ± .017•	.275 ± .009•	.280 ± .011•	.231 ± .009•	.183 ± .019•
B-RL/Square	.255 ± .018•	.164 ± .028•	.131 ± .014•	.112 ± .015•	.276 ± .008•	.280 ± .010•	.236 ± .007•	.215 ± .019•
Ranking loss (the smaller, the better)								
Ub-HL/Hinge	.245 ± .003•	.168 ± .006•	**.100 ± .012**	**.062 ± .004**	.340 ± .010•	.330 ± .014•	.234 ± .008	.130 ± .016•
Ub-HL/Square	**.237 ± .002**	.160 ± .005	.100 ± .013	.067 ± .003•	.343 ± .009•	.333 ± .017•	.245 ± .009•	.180 ± .015•
Ub-RL/Sigmoid	.239 ± .003	**.153 ± .004**	.104 ± .019	.068 ± .003•	**.333 ± .012**	**.313 ± .014**	**.223 ± .012**	**.114 ± .013**
Ub-RL/Hinge	.238 ± .003	.160 ± .004	.105 ± .016	.064 ± .004•	.338 ± .011•	.323 ± .016•	.241 ± .012•	.133 ± .015•
Ub-RL/Square	.240 ± .005	.163 ± .006	.111 ± .020	.070 ± .003•	.348 ± .014•	.336 ± .012•	.270 ± .011•	.227 ± .018•
B-HL/Hinge	.350 ± .036•	.309 ± .114	.153 ± .031•	.193 ± .040•	.388 ± .018•	.389 ± .028•	.294 ± .021•	.248 ± .038•
B-HL/Square	.332 ± .034•	.284 ± .080•	.139 ± .044	.185 ± .035•	.380 ± .014•	.386 ± .029•	.290 ± .021•	.270 ± .035•
B-RL/Sigmoid	.392 ± .024•	.298 ± .053•	.181 ± .031•	.240 ± .037•	.390 ± .014•	.384 ± .038•	.279 ± .020•	.241 ± .034•
B-RL/Hinge	.342 ± .035•	.288 ± .086•	.139 ± .039	.185 ± .033•	.380 ± .014•	.384 ± .033•	.286 ± .019•	.250 ± .034•
B-RL/Square	.330 ± .030•	.259 ± .053•	.143 ± .040	.186 ± .031•	.381 ± .013•	.385 ± .029•	.296 ± .017•	.285 ± .029•
Coverage (the smaller, the better)								
Ub-HL/Hinge	.415 ± .007•	.230 ± .008•	.107 ± .010	**.154 ± .005**	.491 ± .009•	.480 ± .014•	.393 ± .012	.189 ± .019•
Ub-HL/Square	**.403 ± .004**	.222 ± .006	**.107 ± .011**	.164 ± .004•	.493 ± .010•	.481 ± .017•	.401 ± .009•	.250 ± .015•
Ub-RL/Sigmoid	.403 ± .004	**.211 ± .004**	.109 ± .016	.161 ± .005•	**.483 ± .010**	**.466 ± .016**	**.375 ± .011**	**.171 ± .016**
Ub-RL/Hinge	.407 ± .006	.221 ± .006	.111 ± .014	.156 ± .005	.488 ± .012•	.477 ± .019•	.400 ± .018	.191 ± .017•
Ub-RL/Square	.410 ± .009	.224 ± .008	.116 ± .017•	.168 ± .004•	.499 ± .013•	.488 ± .018•	.429 ± .014•	.293 ± .018•
B-HL/Hinge	.502 ± .029•	.354 ± .098•	.151 ± .026•	.324 ± .040•	.539 ± .016•	.532 ± .029•	.449 ± .023•	.317 ± .037•
B-HL/Square	.497 ± .031•	.336 ± .069•	.141 ± .037	.323 ± .036•	.531 ± .013•	.530 ± .030•	.446 ± .022•	.339 ± .032•
B-RL/Sigmoid	.578 ± .016•	.349 ± .040•	.174 ± .027•	.391 ± .033•	.546 ± .010•	.527 ± .039•	.436 ± .027•	.310 ± .032•
B-RL/Hinge	.506 ± .032•	.343 ± .072•	.140 ± .033	.325 ± .033•	.530 ± .013•	.527 ± .033•	.440 ± .024•	.319 ± .031•
B-RL/Square	.497 ± .028•	.317 ± .046•	.144 ± .034	.327 ± .033•	.530 ± .013•	.527 ± .028•	.452 ± .021•	.353 ± .028•
Average precision (the greater, the better)								
Ub-HL/Hinge	.632 ± .004•	.700 ± .007•	**.834 ± .014**	**.790 ± .009**	.510 ± .016	.517 ± .014•	.644 ± .012	.650 ± .034•
Ub-HL/Square	**.640 ± .002**	**.709 ± .008**	.832 ± .017	.786 ± .007•	.511 ± .011	.514 ± .016•	.629 ± .014•	.573 ± .030•
Ub-RL/Sigmoid	.636 ± .008	.703 ± .002	.820 ± .029	.779 ± .008•	**.516 ± .015**	**.533 ± .014**	**.650 ± .017**	**.668 ± .028**
Ub-RL/Hinge	.638 ± .002	.706 ± .007	.825 ± .020•	.782 ± .008•	.515 ± .014	.526 ± .016•	.636 ± .013•	.628 ± .038•
Ub-RL/Square	.634 ± .003•	.702 ± .009•	.813 ± .026•	.771 ± .009•	.506 ± .013•	.508 ± .007•	.597 ± .016•	.458 ± .055•
B-HL/Hinge	.469 ± .047•	.498 ± .140•	.750 ± .036•	.505 ± .091•	.463 ± .022•	.458 ± .024•	.566 ± .021•	.444 ± .070•
B-HL/Square	.510 ± .048•	.554 ± .102•	.784 ± .035	.560 ± .080•	.474 ± .018•	.463 ± .024•	.572 ± .023•	.423 ± .061•
B-RL/Sigmoid	.482 ± .043•	.506 ± .114•	.716 ± .045•	.472 ± .092•	.468 ± .019•	.461 ± .030•	.583 ± .016•	.484 ± .074•
B-RL/Hinge	.502 ± .051•	.564 ± .102•	.780 ± .032•	.559 ± .076•	.472 ± .018•	.463 ± .027•	.577 ± .020•	.457 ± .064•
B-RL/Square	.520 ± .042•	.584 ± .083•	.781 ± .031•	.556 ± .072•	.469 ± .015•	.464 ± .024•	.564 ± .018•	.393 ± .045•

8.2 Study on CCMM data

To validate the effectiveness of the proposed unbiased estimators, we perform experiments on multi-label data with class-conditional multiple noisy labels. To inject label noise into training examples, each class label is flipped according to noise rate ρ_{-1}^j and ρ_{+1}^j randomly sampled from $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. We repeat experiments 5 times with different noise rates and report their averaging results.

We compare the performance of the proposed method with their baselines. Specifically, two variants of the proposed method, i.e., **Unbiased-HL** ($\tilde{\mathcal{L}}_h$ defined in Eq.(4), Ub-HL for short) and **Unbiased-RL** ($\tilde{\mathcal{L}}_r$ defined in Eq.(6), Ub-RL for short) are compared with their baselines, i.e., **Biased-HL** ($\mathcal{L}_h$ defined in Eq.(3), B-HL for short) and **Biased-RL** ($\mathcal{L}_r$ defined in Eq.(5), B-RL for short). For $\tilde{\mathcal{L}}_h$ and $\mathcal{L}_h$, the surrogate loss function ϕ is defined as the hinge loss (Hinge for short) and least square loss (Square for short), respectively; For $\tilde{\mathcal{L}}_r$ and $\mathcal{L}_r$, besides the hinge loss and least square loss, we also adopt sigmoid loss (Sigmoid for short), since it is a consistent surrogate loss, which satisfies the conditions in Theorem 4.

Table 1 reports comparison results of the proposed methods against baseline methods by using linear models. For each data set, paired t-test based on 5 repeats (at 0.05 significance level) is conducted to show weather the proposed unbiased estimator is significantly different from the comparing methods. From the results, it is obvious that the proposed method outperforms their corresponding baselines with significant superiority in almost all cases, which validates the effectiveness of the proposed method. The performance of Ub-RL is generally superior to Ub-HL, which validates the label correlation is beneficial for learning with class-conditional multi-label noise. From the results, regarding Ub-RL, it can be observed that Sigmoid surrogate loss generally achieves better performance than the other losses, which provides an empirical validation of Theorem 4, since Sigmoid loss satisfies the conditions in Theorem 4 and is thus consistent for Ub-RL with respect to ranking loss while others are not [49].

8.3 Study on PML data

To validate the practical usefulness of the proposed unbiased estimators, we perform experiments on PML tasks. Regarding the multi-label datasets, for training instances, each negative label y_j can be flipped into a candidate label according to noise rate ρ^j randomly sampled from $\{0.2, 0.3, 0.4, 0.5, 0.6\}$. The noise rates are unknown in experiments and to be

TABLE 2
Comparison results between the proposed method (using linear model) and PML methods, where •/◦ indicates whether the proposed method is significantly superior/inferior to the comparing method via paired t-test (at 0.05 significance level).

Data	music_emotion	music_style	mirflickr	tmc2007	Multi-MNIST	Multi-KMNIST	Multi-FMNIST	VOC
Hamming loss (the smaller, the better)								
uPML-HL	.200 ± .002	**.114 ± .002**	**.156 ± .008**	**.053 ± .000**	**.234 ± .000**	.231 ± .001	.171 ± .000•	**.052 ± .000**
uPML-RL	**.199 ± .002**	.115 ± .001	.157 ± .003	.054 ± .000•	.234 ± .000	**.230 ± .000**	**.166 ± .000**	.066 ± .011
PMLNI	.228 ± .001•	.115 ± .002	.171 ± .002•	.081 ± .003•	.258 ± .004•	.263 ± .004•	.198 ± .005•	.096 ± .001•
PMLLRS	.231 ± .000•	.121 ± .001•	.175 ± .008	.078 ± .003•	.260 ± .004•	.261 ± .005•	.197 ± .007•	.094 ± .001•
PARVLS	.256 ± .001•	.126 ± .001•	.174 ± .002•	.095 ± .001•	.320 ± .005•	.325 ± .007•	.289 ± .004•	.128 ± .000•
PARMAP	.251 ± .004•	.125 ± .004•	.167 ± .003	.097 ± .002•	.283 ± .004•	.288 ± .004•	.251 ± .003•	.120 ± .002•
fPML	.232 ± .001•	.122 ± .002•	.176 ± .009•	.080 ± .003•	.258 ± .003•	.261 ± .007•	.198 ± .006•	.104 ± .003•
Ranking loss (the smaller, the better)								
uPML-HL	.236 ± .005•	.144 ± .005	.124 ± .007	**.051 ± .000**	.303 ± .000•	.287 ± .001•	.180 ± .000•	.089 ± .004•
uPML-RL	**.226 ± .003**	**.142 ± .002**	**.118 ± .002**	.055 ± .001•	**.298 ± .000**	**.272 ± .000**	**.172 ± .000**	**.083 ± .003**
PMLNI	.250 ± .003•	.145 ± .005	.124 ± .002•	.109 ± .010•	.334 ± .007•	.339 ± .013•	.223 ± .009•	.160 ± .006•
PMLLRS	.260 ± .001•	.151 ± .000•	.127 ± .005•	.106 ± .010•	.340 ± .007•	.340 ± .016•	.221 ± .011•	.155 ± .004•
PARVLS	.356 ± .003•	.243 ± .008•	.170 ± .004•	.186 ± .004•	.462 ± .004•	.467 ± .010•	.401 ± .008•	.291 ± .002•
PARMAP	.317 ± .006•	.197 ± .006•	.139 ± .003•	.197 ± .013•	.405 ± .010•	.410 ± .017•	.339 ± .010•	.283 ± .003•
fPML	.264 ± .003•	.157 ± .004•	.127 ± .007•	.118 ± .011•	.339 ± .009•	.338 ± .014•	.223 ± .012•	.200 ± .011•
Coverage (the smaller, the better)								
uPML-HL	.411 ± .002•	.205 ± .005	.125 ± .006	**.138 ± .001**	.456 ± .000•	.435 ± .000•	.327 ± .000•	.140 ± .004•
uPML-RL	**.392 ± .003**	**.199 ± .003**	**.120 ± .002**	.142 ± .001•	**.450 ± .000**	**.419 ± .000**	**.319 ± .000**	**.133 ± .003**
PMLNI	.413 ± .002•	.204 ± .006	.124 ± .002•	.231 ± .014•	.483 ± .006•	.486 ± .016•	.373 ± .010•	.230 ± .008•
PMLLRS	.419 ± .002•	.209 ± .001•	.128 ± .004	.229 ± .015•	.488 ± .007•	.485 ± .019•	.371 ± .012•	.224 ± .006•
PARVLS	.492 ± .002•	.276 ± .008•	.160 ± .003•	.332 ± .007•	.551 ± .002•	.546 ± .004•	.490 ± .008•	.355 ± .003•
PARMAP	.479 ± .003•	.257 ± .006•	.140 ± .003•	.356 ± .015•	.551 ± .006•	.558 ± .020•	.498 ± .013•	.362 ± .001•
fPML	.423 ± .002•	.215 ± .004•	.127 ± .006•	.245 ± .012•	.491 ± .007•	.487 ± .017•	.376 ± .016•	.272 ± .013•
Average precision (the greater, the better)								
uPML-HL	.646 ± .005	**.730 ± .005**	.753 ± .014	**.818 ± .001**	.561 ± .000•	.570 ± .001•	.707 ± .000•	**.746 ± .004**
uPML-RL	**.649 ± .005**	.728 ± .005	**.755 ± .004**	.810 ± .002•	**.562 ± .000**	**.575 ± .000**	**.717 ± .000**	.744 ± .007
PMLNI	.601 ± .003•	.728 ± .006	.752 ± .004	.723 ± .017•	.516 ± .010•	.505 ± .010•	.651 ± .011•	.621 ± .011•
PMLLRS	.586 ± .003•	.708 ± .002•	.746 ± .013	.732 ± .018•	.513 ± .009•	.507 ± .012•	.654 ± .012•	.632 ± .007•
PARVLS	.524 ± .004•	.658 ± .004•	.731 ± .004•	.631 ± .005•	.458 ± .008•	.451 ± .008•	.530 ± .006•	.470 ± .002•
PARMAP	.533 ± .007•	.671 ± .008•	.752 ± .004	.612 ± .015•	.454 ± .004•	.445 ± .012•	.533 ± .006•	.456 ± .010•
fPML	.583 ± .005•	.703 ± .008•	.744 ± .014	.721 ± .018•	.516 ± .007•	.508 ± .014•	.653 ± .010•	.561 ± .014•

TABLE 3
Friedman statistics F_F in terms of each evaluation metric and the critical value at 0.05 significance level (# comparing algorithms $k = 7$, # data sets $N = 8$).

Evaluation metric	F_F	critical value
Hamming Loss	29.6784	
Ranking loss	55.7200	
Coverage	53.8932	2.3240
Average Precision	34.5364	

estimated by using the estimator proposed in Section 6. We repeat experiments 5 times with different noise rate and report the averaging results.

We compare with five state-of-the-art PML algorithms: PML-NI [37], PARTICLE (including two implementations: PAR-VLS and PAR-MAP) [32], PML-LRS [34] and fPML [60]. It is noteworthy that to make a fair comparison, we use a linear classifier as the base model for uPML in all datasets except for VOC, where we only fine-tune parameters of the last layer while the other parameters are frozen. For other comparing methods, parameters are determined by the performance on validation set if no default value given in their literature.

Table 2 reports the comparison result of two variants of the proposed uPML method, i.e., **uPML-HL** ($\tilde{\mathcal{L}}_h$ defined in Eq.(8)) with hinge loss and **uPML-RL** ($\tilde{\mathcal{L}}_r$ defined in Eq.(9)) with Sigmoid loss, against comparing PML methods. For each data set, paired t-test based on five repeats (at 0.05 significance level) is conducted to show weather the proposed unbiased estimator is significantly different from the comparing methods. The proposed uPML method significantly outperforms the comparing PML methods in almost all cases. In particular, on three real-world PML datasets, i.e., music_emotion, music_style and mirflickr, uPML achieves the best performances on all cases, which demonstrates the practical usefulness of the proposed method. In general, uPML-RL is better than uPML-HL due to the label correlation is considered for the former one.

Furthermore, the commonly used *Friedman test* [61] is employed as the statistical test to analyze the relative performance among the comparing approaches. Table 3 reports the Friedman statistics F_F and the corresponding critical value with respect to each evaluation metric (# comparing algorithms $k = 7$, # data sets $N = 8$). For each evaluation metric, the null hypothesis of indistinguishable performance among the comparing algorithm is rejected at 0.05 significance level.

Finally, the post-hoc *Bonferroni-Dunn test* [61] is utilized to illustrate the relative performance among comparing approaches. Here, uPML is regarded as the control method whose average rank difference against the comparing algorithm is calibrated with the *critical difference* (CD). Accordingly, uPML is deemed to have significantly different performance to one comparing algorithm if their average ranks differ by at least one CD (CD = 2.8494 in our experiment: # comparing algorithms $k = 7$, # data sets $N = 8$). Figure 1 shows the CD diagrams ([61]) on each evaluation metric, where the average rank of each comparing algorithm is marked along the axis (lower ranks to the right). In each subfigure, any comparing algorithms whose average rank is within one CD to that of uPML is interconnected to each other with a thick line. From the figure, it can be observed that: 1)

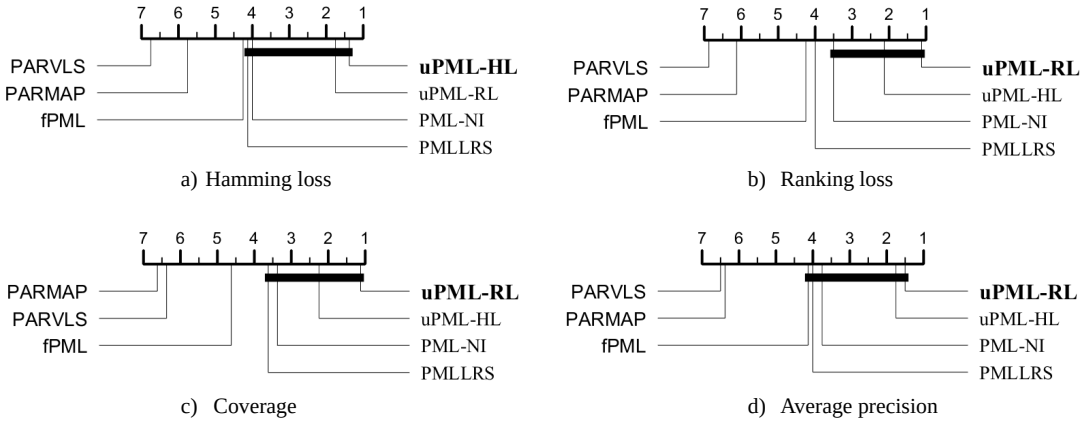

Fig. 1. Comparison of uPML (control algorithm) against five comparing algorithms with the Bonferroni-Dunn test. Algorithms not connected with uPML in the CD diagram are considered to have a significantly different performance from the control algorithm (CD = 2.8494 at 0.05 significance level).

uPML achieves the best (lowest) average rank in terms of all evaluation metrics; 2) uPML is significantly better than the PARVLS and PARMAP in terms of all evaluation metrics; 3) uPML significantly outperforms comparing methods other than PML-NI and PMLLRS in terms of *hamming loss*, *ranking loss* and *coverage*. These experimental results convincingly demonstrate the significance of the superiority for our uPML approach.

9 Conclusion

In this paper, we study the problem of multi-label classification with class-conditional multiple noisy labels, where multiple class labels assigned to each instance may be corrupted simultaneously with class-conditional probabilities. From the perspective of the unbiased estimator, we derive efficient methods for solving CCMN problems with theoretical guarantee. Generally, we prove that learning from class-conditional multiple noisy labels with the proposed unbiased estimators is consistent with respect to hamming loss and ranking loss. Furthermore, we propose a novel method called uPML for solving PML problems, which can be regarded as a special case of CCMN framework. Empirical studies on multiple data sets validate the effectiveness of the proposed method. In the future, we will study CCMN with more loss functions.

References

[1] B. Frénay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 845–869, 2014.

[2] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2691–2699.

[3] I. Jindal, D. Pressel, B. Lester, and M. Nokleby, "An effective label noise model for dnn text classification," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 3246–3256.

[4] B. Biggio, B. Nelson, and P. Laskov, "Support vector machines under adversarial label noise," in *Asian conference on machine learning*, 2011, pp. 97–112.

[5] D. Angluin and P. Laird, "Learning from noisy examples," *Machine Learning*, vol. 2, no. 4, pp. 343–370, 1988.

[6] G. Stempfel and L. Ralaivola, "Learning svms from sloppily labeled data," in *International conference on artificial neural networks*. Springer, 2009, pp. 884–893.

[7] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Advances in neural information processing systems*, 2013, pp. 1196–1204.

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

[9] T. Cour, B. Sapp, and B. Taskar, "Learning from partial labels," *Journal of Machine Learning Research*, vol. 12, pp. 1501–1536, 2011.

[10] M. Xie and S. Huang, "Partial multi-label learning," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018, pp. 4302–4309.

[11] C. Gong, T. Liu, Y. Tang, J. Yang, J. Yang, and D. Tao, "A regularization approach for instance-based superset label learning," *IEEE transactions on cybernetics*, vol. 48, no. 3, pp. 967–978, 2017.

[12] Y. Sun, Y. Zhang, and Z. Zhou, "Multi-label learning with weak label," in *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, 2010.

[13] Y. Freund, "A more robust boosting algorithm," *stat*, vol. 1050, p. 13, 2009.

[14] P. M. Long and R. A. Servedio, "Random classification noise defeats all convex potential boosters," *Machine learning*, vol. 78, no. 3, pp. 287–304, 2010.

[15] M. Dredze, K. Crammer, and F. Pereira, "Confidence-weighted linear classification," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 264–271.

[16] K. Crammer, A. Kulesza, and M. Dredze, "Adaptive regularization of weight vectors," in *Advances in neural information processing systems*, 2009, pp. 414–422.

[17] K. Crammer and D. D. Lee, "Learning via gaussian herding," in *Advances in neural information processing systems*, 2010, pp. 451–459.

[18] N. Manwani and P. Sastry, "Noise tolerance under risk minimization," *IEEE transactions on cybernetics*, vol. 43, no. 3, pp. 1146–1151, 2013.

[19] C. Scott, G. Blanchard, and G. Handy, "Classification with asymmetric label noise: Consistency and maximal denoising," in *Conference On Learning Theory*, 2013, pp. 489–511.

[20] G. Blanchard and C. Scott, "Decontamination of mutually contaminated models," in *Artificial Intelligence and Statistics*, 2014, pp. 1–9.

[21] A. Vahdat, "Toward robustness against label noise in training deep discriminative neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 5596–5605.

[22] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, "Learning from noisy labels with distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1910–1918.

[23] X. Xia, T. Liu, N. Wang, B. Han, C. Gong, G. Niu, and M. Sugiyama, "Are anchor points really indispensable in label-noise learning?"

Advances in Neural Information Processing Systems, vol. 32, pp. 6838–6849, 2019.

[24] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1944–1952.

[25] B. Han, J. Yao, G. Niu, M. Zhou, I. Tsang, Y. Zhang, and M. Sugiyama, "Masking: A new perspective of noisy supervision," in *Advances in Neural Information Processing Systems*, 2018, pp. 5836–5846.

[26] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *International Conference on Machine Learning*, 2018, pp. 4334–4343.

[27] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," *arXiv preprint arXiv:1804.06872*, 2018.

[28] J. Li, R. Socher, and S. C. Hoi, "Dividemix: Learning with noisy labels as semi-supervised learning," *arXiv preprint arXiv:2002.07394*, 2020.

[29] A. Ghosh, H. Kumar, and P. Sastry, "Robust loss functions under label noise for deep neural networks," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[30] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Advances in neural information processing systems*, 2018, pp. 8778–8788.

[31] H. Wang, W. Liu, Y. Zhao, C. Zhang, T. Hu, and G. Chen, "Discriminative and correlative partial multi-label learning," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 2019, pp. 3691–3697.

[32] M.-L. Zhang and J.-P. Fang, "Partial multi-label learning via credible label elicitation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[33] W. Liu, X. Shen, H. Wang, and I. W. Tsang, "The emerging trends of multi-label learning," *arXiv preprint arXiv:2011.11197*, 2020.

[34] T. W. C. L. Lijuan Sun, Songhe Feng and Y. Jin, "Partial multi-label learning by low-rank and sparse decomposition," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, (AAAI-19)*, 2019.

[35] Y. Yan and Y. Guo, "Adversarial partial multi-label learning," *arXiv preprint arXiv:1909.06717*, 2019.

[36] T. Yu, G. Yu, J. Wang, C. Domeniconi, and X. Zhang, "Partial multi-label learning using label compression," in *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2020, pp. 761–770.

[37] M.-K. Xie and S.-J. Huang, "Partial multi-label learning with noisy label identification." in *AAAI*, 2020, pp. 6454–6461.

[38] ——, "Semi-supervised partial multi-label learning," in *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2020, pp. 691–700.

[39] Z.-S. Chen, X. Wu, Q.-G. Chen, Y. Hu, and M.-L. Zhang, "Multi-view partial multi-label learning with graph-based disambiguation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 3553–3560.

[40] M. Zhang and Z. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[41] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[42] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Advances in neural information processing systems*, 2002, pp. 681–687.

[43] N. Lu, G. Niu, A. K. Menon, and M. Sugiyama, "On the minimal supervision for training any binary classifier from only unlabeled data," in *International Conference on Learning Representations*, 2018.

[44] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine learning*, vol. 85, no. 3, p. 333, 2011.

[45] H. Yu, P. Jain, P. Kar, and I. S. Dhillon, "Large-scale multi-label learning with missing labels," in *Proceedings of the 31th International Conference on Machine Learning*, 2014, pp. 593–601.

[46] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 38, no. 3, pp. 447–461, 2015.

[47] M. Ledoux and M. Talagrand, *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

[48] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[49] W. Gao and Z.-H. Zhou, "On the consistency of multi-label learning," *Artificial Intelligence*, vol. 199, no. 1, pp. 22–44, 2013.

[50] T. Zhang, "Statistical behavior and consistency of classification methods based on convex risk minimization," *Annals of Statistics*, pp. 56–85, 2004.

[51] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006.

[52] T. Zhang, "Statistical analysis of some multi-category large margin classification methods," *Journal of Machine Learning Research*, vol. 5, no. Oct, pp. 1225–1251, 2004.

[53] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," PMLR, pp. 1126–1135, 2017.

[54] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[55] T. Chen, Z. Wang, G. Li, and L. Lin, "Recurrent attentional reinforcement learning for multi-label image recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[56] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5177–5186.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[58] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[59] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.

[60] G. Yu, X. Chen, C. Domeniconi, J. Wang, Z. Li, Z. Zhang, and X. Wu, "Feature-induced partial multi-label learning," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 1398–1403.

[61] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.