Unbiased Multi-Label Learning from Crowdsourced Annotations

Mingxuan Xia [1 2] Zenan Huang [2] Runze Wu [3] Gengyu Lyu [4] Junbo Zhao [2] Gang Chen [2] Haobo Wang [1 2]

Abstract

This work studies the novel Crowdsourced Multi-Label Learning (CMLL) problem, where each instance is related to multiple true labels but the model only receives unreliable labels from different annotators. Although a few Crowdsourced Multi-Label Inference (CMLI) methods have been developed, they require both the training and testing sets to be assigned crowdsourced labels and focus on true label inferring rather than prediction, making them less practical. In this paper, by excavating the generation process of crowdsourced labels, we establish the first **unbiased risk estimator** for CMLL based on the crowdsourced transition matrices. To facilitate transition matrix estimation, we upgrade our unbiased risk estimator by aggregating crowdsourced labels and transition matrices from all annotators while guaranteeing its theoretical characteristics. Integrating with the unbiased risk estimator, we further propose a decoupled autoencoder framework to exploit label correlations and boost performance. We also provide a generalization error bound to ensure the convergence of the empirical risk estimator. Experiments on various CMLL scenarios demonstrate the effectiveness of our proposed method. The source code is available at https://github.com/MingxuanXia/CLEAR.

1. Introduction

Multi-label learning (MLL) deals with scenarios where each instance belongs to multiple categories concurrently (Zhang & Zhou, 2007; 2014), which is widely adopted in real-world applications such as image recognition (Zha et al., 2008; Chen et al., 2019b), document classification (Rubin et al., 2012; Xiao et al., 2019), protein function prediction (Wu et al., 2014), and so on. However, the success of MLL relies on large amounts of precisely labeled data, making data annotation labor-intensive and time-consuming. On the other hand, crowdsourcing (Snow et al., 2008; Albarqouni et al., 2016; Rodrigues & Pereira, 2018) has recently established itself as an efficient and cost-effective solution for large-scale data annotation, where labels are collected from low-cost crowds. This gives rise to the potential significance of implementing crowdsourcing in the context of MLL.

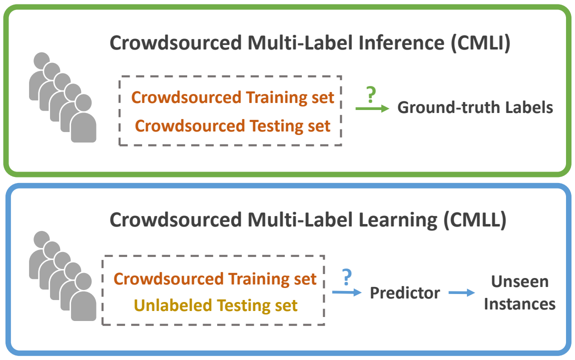
Crowdsourced Multi-Label Inference (CMLI)
Crowdsourced Training set
Crowdsourced Testing set
?
Ground-truth Labels
Crowdsourced Multi-Label Learning (CMLL)
Crowdsourced Training set
Unlabeled Testing set
?
Predictor
Unseen Instances

Figure 1. CMLI approaches focus on directly uncovering the ground-truth labels given the crowdsourced ones on both training and testing sets, while CMLL takes a further step by learning a robust predictor based on crowdsourced labels that can generalize well on unseen instances.

Nonetheless, the study of crowdsourcing MLL has been overlooked, since most existing crowdsourcing methods emphasize multi-class classification problems, where each instance is associated with a single label (Guan et al., 2018; Rodrigues & Pereira, 2018; Wei et al., 2022; Gao et al., 2022). Recently, a few *Crowdsourced Multi-Label Inference* (CMLI) (Zhang & Wu, 2018; Li et al., 2019) approaches have been proposed to address the crowdsourcing scenario when learning with multiple labels. As shown in the upper part of Figure 1, CMLI approaches focus on directly uncovering the ground-truth labels given the crowdsourced ones on both training and testing sets. However, CMLI

[1] School of Software Technology, Zhejiang University, Ningbo, China [2] State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou, China [3] Fuxi AI Lab, NetEase Inc., Hangzhou, China [4] Faculty of Information Technology, Beijing University of Technology, Beijing, China. Correspondence to: Haobo Wang <wanghaobo@zju.edu.cn>, Runze Wu <wurunze1@corp.netease.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

appears to be not only less practical but also lacks solid theoretical grounding. On the one hand, CMLI requires accessing crowdsourced labels on testing sets, which is typically intractable. On the other hand, no existing CMLI methods could provide theoretical guarantees as to how the model trained based on crowdsourced labels generalizes on unseen instances. This gives rise to an emergent question: *How to infer a theoretically robust MLL classifier from crowdsourced labels?*

To bridge the gaps, we deal with the urgent but under-explored CMLL problem which aims to train a multi-label predictor given crowdsourced labels directly and proposed a theoretically grounded method named **CLEAR**, i.e., **C**rowdsourced mu**L**ti-label learning with d**E**coupled **A**utoencode**R**. Specifically, we first excavate the generation process of crowdsourced data in the setting of multiple labels and establish the first **unbiased risk estimator** for CMLL based on the crowdsourced transition matrices. Subsequently, to avoid high time costs and accumulated errors when estimating transition matrices, we upgrade the unbiased risk estimator by aggregating labels from multiple annotators and present the existence and formulation of aggregated transition matrices. We also design feasible solutions for approximating noisy posterior and estimating the aggregated transition matrices which practicably realize our unbiased objective. Equipping with the unbiased risk estimator, we further devise a benchmark solution by a decoupled autoencoder framework with latent space distillation to exploit label correlations and boost performance. Besides, we derive a generalization error bound for our statistically consistent algorithm to guarantee the performance of our method on new instances. Empirically, we evaluate CLEAR on five multi-label datasets under three crowdsourcing scenarios, where CLEAR demonstrates superior results among all baselines including crowdsourcing-based, MLL, and weakly-supervised MLL approaches.

2. Related Work

2.1. Multi-Label Learning

Multi-label learning (MLL) (Liu et al., 2022) is a classical learning paradigm where each data example simultaneously relates to multiple binary labels. The most intuitive strategy to resolve MLL is the one-versus-all algorithm (OVA) (Zhang & Zhou, 2014) that decomposes MLL into several binary prediction problems, which is followed by recent deep approaches (Ridnik et al., 2021; Li et al., 2022; Gao et al., 2023). Despite its simplicity, OVA neglects the rich semantic dependencies among labels and thus suffers from limited performance. To remedy this problem, there has been a plethora of approaches developed, such as chain-based algorithms (Read et al., 2011; Wang et al., 2016), graph-based methods (Chen et al., 2019b; Zhu et al., 2023), attention-based method (Huynh & Elhamifar, 2020a; Zhu & Wu, 2021) and vision-language models (Hu et al., 2023; Ding et al., 2023). Amongst them, the label embedding algorithm (Chen & Lin, 2012; Yeh et al., 2017; Chen et al., 2019a; Wang et al., 2020; Xiong et al., 2022) is a popular solution that assumes the label vectors can be projected into a lower dimensional space due to semantic relations. Following this line of work, we also devise a label embedding framework, which only manipulates the feature space, to be compatible with our unbiased loss.

2.2. Weakly-Supervised Multi-Label Learning

Classical MLL approaches mostly assume the training data are fully-supervised. However, due to the complicated structure of the label space, it can be too expensive and time-consuming to collect precise labels. To mitigate this problem, researchers have proposed a variety of weakly-supervised settings of MLL, including semi-supervised MLL (Wei et al., 2018; Shi et al., 2020; Wang et al., 2021), multi-label with missing labels (Durand et al., 2019; Huynh & Elhamifar, 2020b; Schultheis et al., 2022), MLL with single positive label (Cole et al., 2021; Cho et al., 2022; Xu et al., 2022), and partial MLL (Xie & Huang, 2018; Wang et al., 2019; Lyu et al., 2020; Xu et al., 2020). In this work, we study the crowdsourced MLL that collects labels from multiple weak annotators for reduced cost.

2.3. Crowdsourcing

Crowdsourcing is a popular paradigm that collects low-cost but unreliable labels, which release the burden of large-scale data annotations (Liu et al., 2023; Wang et al., 2024). Traditional crowdsourcing methods model crowdsourced labels by expectation-maximization (EM) algorithm (Dawid & Skene, 1979) that identify the accurate labels (Whitehill et al., 2009; Raykar et al., 2009; Raykar & Yu, 2012; Dalvi et al., 2013; Zhang et al., 2016). Subsequently, deep learning-based methods (Albarqouni et al., 2016; Guan et al., 2018) are proposed and demonstrate superiority, where they deal with crowdsourced label noise by learning label transition matrices (Rodrigues & Pereira, 2018; Li et al., 2023; Chen et al., 2020; Wei et al., 2022; Gao et al., 2022). However, these methods study the problem where each instance is associated with a single label. Instead, we explore the crowdsourcing problem in the multi-label learning scenario where samples are related to multiple labels. There have also been some works (Zhang & Wu, 2018; Li et al., 2019) studying the crowdsourcing problem in the context of learning with multiple labels. Nevertheless, they mostly concentrate on inferring the ground-truth labels behind the crowdsourced labels and need further training to obtain a predictor. In contrast, our work aims to learn a classifier end-to-end that can be *generalized on unseen instances* and *provides theoretical insights*.

3. Problem Setting

3.1. Multi-Label Learning

Multi-label learning (MLL) aims at assigning each instance multiple binary labels simultaneously. Let $\mathcal{X}$ denotes the d-dimensional feature space and $\mathcal{Y} = \{0,1\}^K$ denotes the label space with K class labels. The training dataset $\mathcal{D} = \{(\boldsymbol{x}^i, \boldsymbol{y}^i) | 1 \le i \le n\}$ contains n examples, where $\boldsymbol{x}^i \in \mathcal{X}$ is the instance vector and $\boldsymbol{y}^i \in \mathcal{Y}$ is the label vector. In this setting, $y^i_k = 1$ indicates that the k-th label is associated with instance $\boldsymbol{x}^i$ and $y^i_k = 0$, otherwise. MLL aims to learn a multi-label predictor $\boldsymbol{f} : \mathcal{X} \to \mathcal{Y}$ by minimizing the following risk:

$$R(\boldsymbol{f}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}) \sim P(\boldsymbol{X},\boldsymbol{Y})} \left[\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y}) \right], \tag{1}$$

where $\mathcal{L} : \mathbb{R}^K \times \mathcal{Y} \to \mathbb{R}$ is the multi-label loss function. $\boldsymbol{X}, \boldsymbol{Y}$ denotes the random variables of $\boldsymbol{x}, \boldsymbol{y}$, and $P(\boldsymbol{X}, \boldsymbol{Y})$ is the data distribution from where the dataset is sampled. Note that we say a method guarantees ***risk consistency*** if an unbiased risk estimator is implemented, i.e., the risk estimator that is equivalent to $R(\boldsymbol{f})$ given the same classifier $\boldsymbol{f}$ (Mohri et al., 2018; Feng et al., 2020).

3.2. Crowdsourced Multi-Label Learning

In this paper, we study a novel scenario called Crowdsourced Multi-Label Learning (CMLL), where the fully supervised data is not accessible, and a crowdsourced dataset $\tilde{\mathcal{D}} = \{(\boldsymbol{x}^i, \{\tilde{\boldsymbol{y}}^i_m\}_{m=1}^M) | 1 \le i \le n\}$ is given. Specifically, each instance is labeled by M annotators independently, and $\tilde{\boldsymbol{y}}^i_m \in \mathcal{Y}$ denotes the label vector tagged by the m-th annotator. Let $\tilde{y}^i_{mk}$ denote the label on k-th class given by the m-th annotator. The goal of CMLL is to learn a multi-label predictor $\boldsymbol{f} : \mathcal{X} \to \mathcal{Y}$ from $\tilde{\mathcal{D}}$ to assign a relevant label set for each unseen instance. It is worth noting that the ground-truth label $\boldsymbol{y}^i$ corresponding to each instance is related to the crowdsourced labels $\{\tilde{\boldsymbol{y}}^i_m\}_{m=1}^M$, but is inaccessible during training. In the following section, we omit the sample index i when the context is CLEAR.

Data Generation Process of CMLL. We consider that crowdsourced labels $\tilde{y}_{mk}$ are corrupted from their ground-truth label y_k through $M \times K$ class-dependent instance-independent transition matrices $\{\boldsymbol{T}^{mk}\}_{m=1,k=1}^{M,K} \in [0,1]^{2\times 2}$. Denoting $\tilde{Y}_{mk}$ and Y_k as the random variables of $\tilde{y}_{mk}$ and y_k, the transition matrix is defined by $T^{mk}_{ij} = P(\tilde{Y}_{mk} = j | Y_k = i), \forall i,j \in \{0,1\}$. With the instance-independent assumption (Xie & Huang, 2023; Li et al., 2022), i.e. $P(\tilde{Y}_{mk} = j | Y_k = i, \boldsymbol{X} = \boldsymbol{x}) = P(\tilde{Y}_{mk} = j | Y_k = i)$, the transition matrix bridges the class posterior probabilities for noisy and clean data following:

$$P(\tilde{Y}_{mk} = j | \boldsymbol{X} = \boldsymbol{x}) = \sum_{i \in \{0,1\}} T^{mk}_{ij} P(Y_k = i | \boldsymbol{X} = \boldsymbol{x}), \\ \forall j \in \{0,1\}, T^{mk}_{01} + T^{mk}_{10} < 1. \tag{2}$$

Note that we assume the annotators will not make profound mistakes (Xie & Huang, 2023; Gao et al., 2022), which gives rise to the constraint on $\boldsymbol{T}^{mk}$ in Eq. (2).

4. The Proposed Method

4.1. Unbiased Risk Estimator

In this subsection, we establish the first unbiased risk estimator for the CMLL problem. The theorem proposed below guarantees risk consistency when solving CMLL.

Theorem 1. *By decomposing the MLL problem into K independent binary classification problem, i.e., $\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y}) = \sum_{k=1}^K \ell(f_k(\boldsymbol{x}), y_k)$, where f_k refers to prediction of the model on k-th class and ℓ is the base loss function. With $\tilde{R}(\boldsymbol{f}) = \mathbb{E}_{P(\boldsymbol{X},\tilde{\boldsymbol{Y}})} \left[\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}), \{\tilde{\boldsymbol{y}}_m\}_{m=1}^M) \right]$, and define*

$$\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}), \{\tilde{\boldsymbol{y}}_m\}_{m=1}^M) = \frac{1}{2^M} \sum_{k=1}^K \sum_{j=0}^1 \\ \frac{P(Y_k = j | \boldsymbol{X} = \boldsymbol{x})}{\prod_{m=1}^M \sum_{i=0}^1 T^{mk}_{i\tilde{y}_{mk}} P(Y_k = i | \boldsymbol{X} = \boldsymbol{x})} \ell(f_k(\boldsymbol{x}), j), \tag{3}$$

where $\tilde{\boldsymbol{Y}}$ denotes the random variable of the crowdsourced labels for each $\boldsymbol{x}$. Then, $\tilde{R}(\boldsymbol{f})$ is the unbiased risk estimator with respect to $R(\boldsymbol{f})$.

The proof is provided in Appendix A. Note that decomposing MLL loss into multiple binary classification loss is commonly used for deep MLL (Ridnik et al., 2021; Li et al., 2022; Gao et al., 2023).

Remark. The unbiased risk estimator provided by theorem 1 directly models the impact on each individual annotator. However, this objective requires estimating $M \times K$ individual transition matrices, which is not only time-consuming but also troublesome since the transition matrix estimation error can accumulate. In what follows, we show that there exists an alternative solution that aggregates M crowdsourced label vectors $\{\tilde{\boldsymbol{y}}_m\}_{m=1}^M \in \{0,1\}^{M\times K}$ into one label vector $\tilde{\boldsymbol{y}} \in \{0,1\}^K$. In this way, we only need to estimate K transition matrices if there do exist transition matrices for those aggregated labels.

Theorem 2. *Let $\tilde{\boldsymbol{y}} = [\tilde{y}_1, \ldots, \tilde{y}_K]$ be the aggregated label vector for each $\boldsymbol{x}$, and $\tilde{Y}_k$ is the random variable of $\tilde{y}_k$. We have the following consequences:*

<u>(Existence)</u> There exist a set of class-dependent instance-independent transition matrices $\{\bar{\boldsymbol{T}}^k\}_{k=1}^K \in [0,1]^{2\times 2}$ such

that $\bar{T}^k_{ij} = P(\tilde{Y}_k = j|Y_k = i), \forall i, j \in \{0, 1\}$*, the unbiased risk estimator for CMLL with respect to* $R(\boldsymbol{f})$ *is* $\tilde{R}(\boldsymbol{f}) = \mathbb{E}_{P(\boldsymbol{X},\tilde{\boldsymbol{Y}})}\left[\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}), \tilde{\boldsymbol{y}})\right]$*, and*

$$\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}), \tilde{\boldsymbol{y}}) = \sum_{k=1}^{K}\left(\frac{P(\tilde{Y}_k = 1|\boldsymbol{X} = \boldsymbol{x}) - \bar{T}^k_{01}}{1 - \bar{T}^k_{01} - \bar{T}^k_{10}}\ell(f_k(\boldsymbol{x}), 1) + \frac{P(\tilde{Y}_k = 0|\boldsymbol{X} = \boldsymbol{x}) - \bar{T}^k_{10}}{1 - \bar{T}^k_{01} - \bar{T}^k_{10}}\ell(f_k(\boldsymbol{x}), 0)\right) \tag{4}$$

<u>(Formulation of Transition Matrices)</u> Let A be the random variable of the index of the annotator. By denoting $\omega^m = P(A = m|\boldsymbol{X} = \boldsymbol{x})$ *as the contribution of m-th annotator on tagging $\boldsymbol{x}$, and with $\boldsymbol{T}^{mk}$ defined in subsection 3.2, the transition matrices $\{\bar{\boldsymbol{T}}^k\}_{k=1}^K$ for aggregated labels are formalized as linear combinations of $\boldsymbol{T}^{mk}$:*

$$\bar{\boldsymbol{T}}^k = \sum_{m=1}^{M}\omega^m \cdot \boldsymbol{T}^{mk}, \tag{5}$$

which are class-dependent and instance-independent.

The proof of Theorem 2 is provided in the Appendix B. Theorem 2 enables us to adopt an aggregated version of the unbiased risk estimator, reducing the cost of estimating transition matrices.

Practical Implementation. Despite the efficiency brought by objective 4, the aggregated label $\tilde{\boldsymbol{y}}$ is unfortunately inaccessible and so does its noisy posterior probability $P(\tilde{Y}_k = 1|\boldsymbol{X} = \boldsymbol{x})$. To deal with this problem, by assuming that each annotator tags each instance with uniform contribution, we approximate $P(\tilde{Y}_k = 1|\boldsymbol{X} = \boldsymbol{x})$ by averaging the crowdsourced labels of M annotators, i.e., $s_k = \frac{1}{M}\tilde{y}_{mk}$. Thus, our unbiased objective function is finally formalized as:

$$\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{s}) = \sum_{k=1}^{K}\left(\frac{\left[s_k - \bar{T}^k_{01}\right]_+}{1 - \bar{T}^k_{01} - \bar{T}^k_{10}}\ell(f_k(\boldsymbol{x}), 1) + \frac{\left[(1 - s_k) - \bar{T}^k_{10}\right]_+}{1 - \bar{T}^k_{01} - \bar{T}^k_{10}}\ell(f_k(\boldsymbol{x}), 0)\right), \tag{6}$$

where $\boldsymbol{s} = [s_1, \dots, s_K]$ denotes the averaged crowdsourced label vector for sample $\boldsymbol{x}$, and $[\cdot]_+$ is abbreviated for $\max(\cdot, 0)$ which ensures the loss non-negative.

Transition Matrix Estimation. With the existence of the aggregated transition matrices proved in Theorem 2, we further introduce how we estimate them in practice. Our implementation is motivated by the anchor point assumption, which is widely adopted in noisy label learning (Liu & Tao, 2016; Patrini et al., 2017; Xia et al., 2019). Here, we present the transition matrix estimator following the anchor point assumption in the setting of MLL.

Algorithm 1 Pseudo-code of CLEAR.

```
Input: Crowdsourced multi-label dataset $\tilde{\mathcal{D}}$
Aggregating the crowdsourced labels by $s_k = \frac{1}{M}\tilde{y}_{mk}$
Fitting $\boldsymbol{s}$ by a neural network to estimate $\{\bar{\boldsymbol{T}}^k\}_{k=1}^K$ by averaging $\boldsymbol{s}$ of the top-$C$ anchor points
Initialize the input of the label VAE by $\boldsymbol{s}' = \boldsymbol{s}$
for $epoch = 1, 2, \dots$ do
    for $step = 1, 2, \dots$ do
        Calculate the unbiased loss $\mathcal{L}_{unbiased}$ by Eq. (7)
        Calculate the distillation loss $\mathcal{L}_{distill}$ by Eq. (10)
        Train the decoupled autoencoders $\boldsymbol{f}$ and $\boldsymbol{f}'$ by minimizing $\mathcal{L}_{final} = \mathcal{L}_{unbiased} + \mathcal{L}_{distill}$
        Update $\boldsymbol{s}'$ by Eq. (9)
    end for
end for
Output: Multi-label predictor $\boldsymbol{f}$
```

Proposition 1. *Given a sample $\boldsymbol{x}$, if $\boldsymbol{x}$ satisfies* $P(Y_k = a|\boldsymbol{X} = \boldsymbol{x}) = 1, a \in \{0, 1\}$*, we say that $\boldsymbol{x}$ is the anchor point for label value a of class k, and we have* $\bar{T}^k_{aj} = P(\tilde{Y}_k = j|\boldsymbol{X} = \boldsymbol{x})$.

The proof is given by $P(\tilde{Y}_k = j|\boldsymbol{X} = \boldsymbol{x}) = \sum_{i\in\{0,1\}}\bar{T}^k_{ij}P(Y_k = i|\boldsymbol{X} = \boldsymbol{x}) = \bar{T}^k_{aj}$. In other words, proposition 1 enables us to estimate the transition matrices based on the noisy class probabilities, which are approximated by the aggregated crowdsourced label s_k as mentioned above. Following (Liu & Tao, 2016; Patrini et al., 2017; Xia et al., 2019), we select samples that are far from the classification boundary as anchor points, namely, $\bar{\boldsymbol{x}}^{ka} = \arg\max_{\boldsymbol{x}\in\mathcal{D}} a \cdot \hat{f}_k(\boldsymbol{x}) + (1 - a) \cdot (1 - \hat{f}_k(\boldsymbol{x}))$, where $\bar{\boldsymbol{x}}^{ka}$ is the anchor point for label a of class k, and $\hat{f}$ is the multi-label predictor after sigmoid, which is trained by s_k. Moreover, instead of selecting the most confident sample, we select top-C confident samples as anchor points and take the average of their noisy class probabilities to approximate the aggregated transition matrices, which turned out to be more robust. The detailed pseudo-code of transition matrix estimation is summarized in Algorithm 2.

4.2. Training with Decoupled Autoencoder

Despite the risk consistency provided by the above objective functions, all labels are treated independently. To capture the rich semantic correlation among labels, we propose a benchmark solution for CMLL, i.e., **C**rowdsourced mu**L**ti-label learning with d**E**coupled **A**utoencode**R** (**CLEAR**), which integrates the unbiased risk estimator into a decoupled autoencoder framework.

As shown in Figure 2, CLEAR contains two variational au-

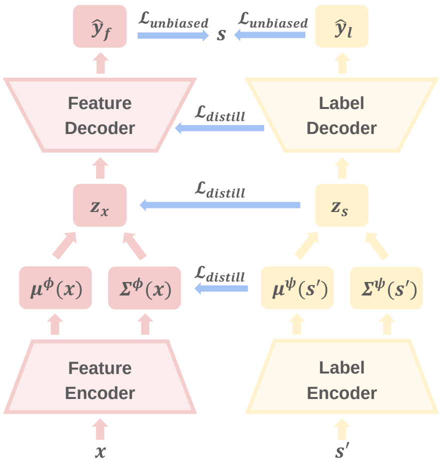

Figure 2. Model architecture of our proposed CLEAR framework.

toencoders (VAE), namely a feature VAE $\boldsymbol{f}$ and a label VAE $\boldsymbol{f}'$, where the feature $\boldsymbol{x}$ and the denoised label $\boldsymbol{s}'$ are encoded and decoded respectively to reconstruct the aggregated crowdsourced label $\boldsymbol{s}$. On the one hand, the unbiased risk estimator is implemented as the reconstruction loss of the two VAEs, which encourages training robust classifiers and building clean latent spaces. On the other hand, we leverage the label VAE whose latent embedding contains implicit label correlations, to distill its latent space to the feature VAE. Guided by the unbiased objective and label correlation distillation, the feature VAE $\boldsymbol{f}$ can be optimized and serves as the predictor at test time.

Specifically, each encoder first maps each input to a Gaussian subspace, namely $\mathcal{N}(\mu^{\phi}(\boldsymbol{x}), \Sigma^{\phi}(\boldsymbol{x}))$ and $\mathcal{N}(\mu^{\psi}(\boldsymbol{s}'), \Sigma^{\psi}(\boldsymbol{s}'))$, where ϕ and ψ are trainable parameters for the two encoders. Let $\boldsymbol{z}_x$ and $\boldsymbol{z}_s$ denote samples from these two distributions respectively. Then, we map $\boldsymbol{z}_x$ and $\boldsymbol{z}_s$ through two decoders $\boldsymbol{g}_x$ and $\boldsymbol{g}_y$ respectively, and input them into two Multivariate Probit Models (Chen et al., 2018) to output final prediction $\hat{\boldsymbol{y}}_f$ and $\hat{\boldsymbol{y}}_l$, which is proved to be effective on building label dependencies (Bai et al., 2020). With the unbiased loss $\tilde{\mathcal{L}}$ defined in Eq. (6), our reconstruction loss can be formalized as:

$$\mathcal{L}_{unbiased} = \tilde{\mathcal{L}}(\hat{\boldsymbol{y}}_f, \boldsymbol{s}) + \tilde{\mathcal{L}}(\hat{\boldsymbol{y}}_l, \boldsymbol{s}). \tag{7}$$

Without loss of generality, we implement the popular binary cross-entropy loss (BCE) as the base loss function, i.e.:

$$\ell(\hat{y}_k, s_k) = -\left(s_k \log \sigma(\hat{y}_k) + (1 - s_k) \log \sigma(1 - \hat{y}_k)\right), \tag{8}$$

where $\hat{y}_k$ denotes the k-th value of the output vector $\hat{\boldsymbol{y}}_f$ or $\hat{\boldsymbol{y}}_l$. Note that instead of directly reconstructing $\boldsymbol{s}$ for the label VAE, which might be problematic since the crowdsourced labels are not reliable, we adopt a stable solution that uses a denoised label $\boldsymbol{s}'$ as the input of the label VAE which is initialized by $\boldsymbol{s}$ and progressively refined by the more and more reliable output $\hat{\boldsymbol{y}}_f$, i.e.,

$$\boldsymbol{s}' = \eta \cdot \boldsymbol{s}' + (1 - \eta) \cdot \text{MultiHot}(\hat{\boldsymbol{y}}_f), \tag{9}$$

where η is the momentum parameter and MultiHot$(\hat{\boldsymbol{y}}_f)$ is the Multi-Hot version of $\hat{\boldsymbol{y}}_f$ with threshold 0.5.

Then we define the latent space distillation loss by the distance measures on multiple layers between the two autoencoders, i.e.,

$$\mathcal{L}_{distill} = \alpha \mathcal{L}_{kl} + \beta \mathcal{L}_{mse}, \tag{10}$$

where $\mathcal{L}_{kl}$ is the KL divergence between the two multivariate Gaussian distributions, and $\mathcal{L}_{mse}$ is the mean square error of the latent embedding samples and decoder output between the two autoencoders. α, β are hyper-parameters that trade off the weights of different losses. The two losses are formalized as follows:

$$\begin{aligned}\mathcal{L}_{kl} = \sum_{i=1}^{d} \log \frac{\Sigma_{i,i}^{\phi}(\boldsymbol{x})}{\Sigma_{i,i}^{\psi}(\boldsymbol{s}')} - d + \sum_{i=1}^{d} \frac{\Sigma_{i,i}^{\psi}(\boldsymbol{s}')}{\Sigma_{i,i}^{\phi}(\boldsymbol{x})} \\ + \sum_{i=1}^{d} \frac{(\mu_i^{\phi}(\boldsymbol{x}) - \mu_i^{\psi}(\boldsymbol{s}'))^2}{\Sigma_{i,i}^{\phi}(\boldsymbol{x})},\end{aligned} \tag{11}$$

$$\mathcal{L}_{mse} = (\boldsymbol{z}_x - \boldsymbol{z}_s)^2 + (\boldsymbol{g}_x(\boldsymbol{z}_x) - \boldsymbol{g}_y(\boldsymbol{z}_s))^2. \tag{12}$$

where d is the dimension of the Gaussian subspace. Overall, the final objective of CLEAR is defined by $\mathcal{L}_{final} = \mathcal{L}_{unbiased} + \mathcal{L}_{distill}$. The pseudo-code of CLEAR is summarized in Algorithm 1.

4.3. Generalization Error Bound

In this subsection, we establish a generalization error bound for our proposed method. With the unbiased risk estimator defined in Eq. (6), we can obtain an learned classifier $\hat{\boldsymbol{f}}$ by minimizing the empirical risk $\bar{R}(\boldsymbol{f}) = \frac{1}{n}\sum_{i=1}^{n} \tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{s}_i)$. We then define $\mathcal{F}$ as the hypothesis class and $\mathcal{H}_k = \{h : \boldsymbol{x} \mapsto f_k(\boldsymbol{x}) | \boldsymbol{f} \in \mathcal{F}\}$ as the functional space for the k-th class. Further, by denoting $\mathfrak{R}_n(\mathcal{H}_k)$ as the expected Rademacher complexity (Bartlett & Mendelson, 2002) of $\mathcal{H}_k$ with sample size n, the generalization error bound for our proposed unbiased risk estimator can be derived as the following theorem.

Theorem 3. *Assume that the true aggregated transition matrices $\{\bar{\boldsymbol{T}}^k\}_{k=1}^{K}$ are given, and the loss function $\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{s})$ is L_T-Lipschitz continuous with respect to $\boldsymbol{f}(\boldsymbol{x})$, and the base loss function l is upper-bounded by λ. Let $\mu = \max_k \frac{1}{1 - \bar{T}_{01}^k - \bar{T}_{10}^k}$. Then, for any $\delta > 0$, with probability*

Table 1. Comparison of CLEAR with baselines when $\bar{T}_{01}^k = 0.2$ and $\bar{T}_{10}^k = 0.2$. The best results are shown in boldface.

Metric	Dataset	BCE	MV	DoctorNet	ML-KNN	MPVAE	PML-NI	CLEAR
Example-F1	Image	0.6360	0.6433	0.6427	0.5060	0.6269	0.5845	**0.6730**
	Scene	0.7241	0.7277	0.7288	0.6701	0.7106	0.5949	**0.7596**
	Corel5K	0.0194	0.0210	0.0234	0.0172	**0.1240**	0.1084	0.1237
	Mirflickr	0.7792	0.7838	0.7841	0.7098	0.7875	0.6830	**0.7997**
	NUS-WIDE	0.2677	0.2931	0.2893	0.1014	0.2934	0.0677	**0.3198**
Micro-F1	Image	0.6579	0.6677	0.6721	0.5707	0.6491	0.6072	**0.6809**
	Scene	0.7569	0.7579	0.7593	0.7219	0.7364	0.5706	**0.7743**
	Corel5K	0.0195	0.0210	0.0249	0.0251	**0.1437**	0.1375	0.1432
	Mirflickr	0.8162	0.8185	0.8160	0.7567	0.8179	0.6717	**0.8254**
	NUS-WIDE	0.3358	0.3524	0.3420	0.1628	0.3533	0.0630	**0.3827**
Macro-F1	Image	0.6621	0.6692	0.6754	0.5666	0.6508	0.6079	**0.6851**
	Scene	0.7639	0.7634	0.7651	0.7249	0.7433	0.5750	**0.7836**
	Corel5K	0.0103	0.0049	0.0013	0.0067	0.0237	0.0216	**0.0263**
	Mirflickr	0.7133	0.7228	0.7175	0.6330	0.7240	0.5591	**0.7405**
	NUS-WIDE	0.0314	0.0423	0.0407	0.0194	0.0714	0.0548	**0.0740**

Table 2. Comparison of CLEAR with baselines when $\bar{T}_{01}^k = 0.2$ and $\bar{T}_{10}^k = 0.5$. The best results are shown in boldface.

Metric	Dataset	BCE	MV	DoctorNet	ML-KNN	MPVAE	PML-NI	CLEAR
Example-F1	Image	0.3098	0.3183	0.3347	0.0305	0.3292	0.5617	**0.5738**
	Scene	0.3348	0.3337	0.3519	0.0955	0.3656	0.4310	**0.6055**
	Corel5K	0.0182	0.0176	0.0112	0.0005	0.0390	0.0241	**0.0601**
	Mirflickr	0.4272	0.4344	0.4403	0.0535	0.4242	0.4242	**0.7267**
	NUS-WIDE	0.0101	0.1028	0.0876	0.0001	0.0781	0.0832	**0.1968**
Micro-F1	Image	0.3905	0.4092	0.4189	0.0497	0.4181	0.5761	**0.6036**
	Scene	0.4325	0.4468	0.4528	0.1628	0.4662	0.4199	**0.6609**
	Corel5K	0.0182	0.0176	0.0121	0.0007	0.0406	0.0341	**0.0677**
	Mirflickr	0.5109	0.5278	0.5170	0.0773	0.5039	0.4278	**0.7818**
	NUS-WIDE	0.0219	0.1474	0.1290	0.0001	0.1122	0.0781	**0.2565**
Macro-F1	Image	0.3892	0.4074	0.4192	0.0492	0.4165	0.5783	**0.6050**
	Scene	0.4338	0.4473	0.4552	0.1563	0.4652	0.4186	**0.6606**
	Corel5K	0.0099	0.0045	0.0008	0.0003	0.0145	0.0060	**0.0159**
	Mirflickr	0.3949	0.4159	0.3993	0.0799	0.4015	0.3651	**0.6328**
	NUS-WIDE	0.0019	0.0142	0.0121	0.0005	0.0163	**0.0584**	0.0491

at least $1-\delta$*, we have*

$$\mathbb{E}[\bar{R}(\hat{\boldsymbol{f}})] - \bar{R}(\hat{\boldsymbol{f}}) \leq 2\sqrt{2}L_T \sum_{k=1}^{K} \mathfrak{R}_n(\mathcal{H}_k) + \lambda K(\mu+1)\sqrt{\frac{\log(1/\delta)}{2n}}. \tag{13}$$

Theorem 3 shows that minimizing the empirical risk can bound population level error, which ensures the generalization ability of our proposed unbiased loss. The proof is given in Appendix C.

5. Experiments

In this section, we report our empirical results to show the superiority of CLEAR. We refer the readers to the Appendix for more experimental results.

5.1. Setup

Datasets. We conduct our experiments on five benchmark multi-label image datasets[1], including *Image*, *Scene*, *Corel5K*, *Mirflickr*, *NUS-WIDE*. For these datasets, we corrupt the training sets according to true transition matrices $\{\boldsymbol{T}^{mk}\}_{m=1,k=1}^{M,K}$. We consider the following three CMLL scenarios: $T_{01}^{mk} = T_{10}^{mk}$, $T_{01}^{mk} < T_{10}^{mk}$, and $T_{01}^{mk} > T_{10}^{mk}$. Specifically, the inverse diagonal elements of the aggregated transition matrices $(\bar{T}_{01}^k, \bar{T}_{10}^k)$ are set as (0.2,0.2), (0.2,0.5), and (0.5,0.2) for the above three scenarios. For convenience, for each annotator, we adopt the same true transition matrices for all classes but do not leak this information to the algorithm. Moreover, we set the number of annotators as $M = 5$ for all the experiments unless otherwise specified.

Baselines. For a comprehensive comparison, we exploit the following four types of baselines: 1) A naive baseline **BCE**, which trains the classifier by BCE loss on the aggre-

[1] http://mulan.sourceforge.net/datasets-mlc.html

Table 3. Comparison of CLEAR with baselines when $\bar{T}_{01}^{k} = 0.5$ and $\bar{T}_{10}^{k} = 0.2$. The best results are shown in boldface.

Metric	Dataset	BCE	MV	DoctorNet	ML-KNN	MPVAE	PML-NI	CLEAR
Example-F1	Image	0.4885	0.4889	0.4912	0.3938	0.4891	0.5824	**0.6224**
	Scene	0.4429	0.4433	0.4482	0.3539	0.4484	0.3122	**0.7043**
	Corel5K	0.0259	0.0261	**0.0264**	0.0255	0.0256	0.0075	0.0258
	Mirflickr	0.5432	0.5395	0.5453	0.4274	0.5388	0.3969	**0.7106**
	NUS-WIDE	0.0782	**0.0834**	0.0821	0.0697	0.0817	0.0571	0.0763
Micro-F1	Image	0.4821	0.4836	0.4856	0.4006	0.4819	0.5753	**0.6120**
	Scene	0.4254	0.4258	0.4306	0.3499	0.4272	0.3126	**0.6865**
	Corel5K	0.0259	0.0261	**0.0264**	0.0255	0.0256	0.0103	0.0258
	Mirflickr	0.5402	0.5382	0.5423	0.4329	0.5366	0.4045	**0.7166**
	NUS-WIDE	0.0794	**0.0856**	0.0844	0.0706	0.0830	0.0578	0.0786
Macro-F1	Image	0.4796	0.4816	0.4837	0.3986	0.4796	0.5778	**0.6235**
	Scene	0.4236	0.4252	0.4288	0.3575	0.4268	0.3114	**0.7051**
	Corel5K	0.0180	0.0184	0.0187	0.0176	0.0180	0.0017	**0.0192**
	Mirflickr	0.4275	0.4264	0.4302	0.3610	0.4265	0.3520	**0.6378**
	NUS-WIDE	0.0542	0.0622	0.0628	0.0523	0.0612	0.0523	**0.0683**

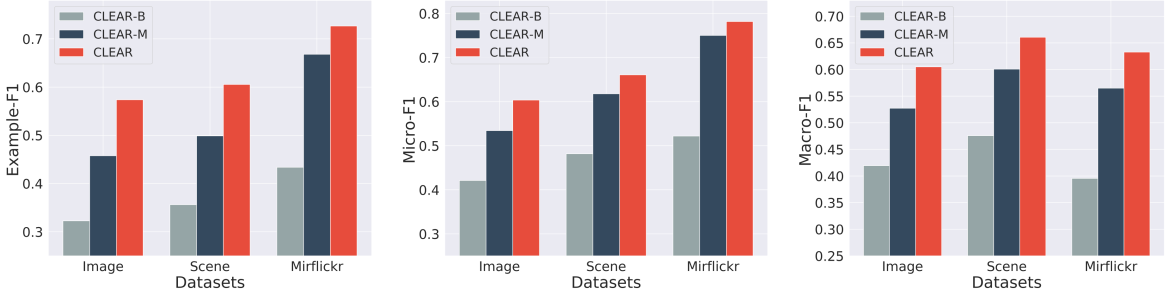

Figure 3. Ablation analysis when $\bar{T}_{01}^{k} = 0.2$ and $\bar{T}_{10}^{k} = 0.5$ where CLEAR is compared with its variant CLEAR-B and CLEAR-M.

gated crowdsourced label s; 2) Two crowdsourcing-based methods, namely, **MV** (Majority Voting) (Zhou, 2012), which trains the classifier with majority voting labels, and **DoctorNet** (Guan et al., 2018), which models multiple annotators individually and averages their outputs at test time. Note that we use sigmoid layers and BCE loss to replace the softmax layers and cross-entropy loss; 3) two MLL methods, namely, **ML-KNN** (Zhang & Zhou, 2007) a nearest-neighbor based MLL approach where we input the majority voting labels as the training targets, and **MPVAE** (Bai et al., 2020), a Multivariate Probit Variational Autoencoder designed for learning latent embedding spaces with label correlations; 4) A partial multi-label learning (PML) methods **PML-NI** (Xie & Huang, 2022), which simultaneously models the ground-truth labels and noisy labels.

Implementation Details. The encoder and decoder for CLEAR are parameterized as three fully connected layer neural networks with hidden sizes 512 and 256. To facilitate fair comparison, the compared methods are equipped with the same network structure in cases where neural networks are used. Note that for the baseline BCE, MV, and DoctorNet, we also implement the three-layer fully connected networks of hidden sizes 512 and 256. Following (Bai et al., 2020), we train the models with Adam optimizer (Kingma & Ba, 2015) with a learning rate of 7.5×10^{-4} and a weight decay of $1e^{-5}$. For hyper-parameters in CLEAR, the confident-sample number C and the momentum parameter η are fixed as 20 and 0.9 for all settings. The Gaussian subspace dimensionality d is set as 100 for Corel5K and NUS-WIDE, and 50 otherwise. The trade-off parameters α and β are set by 1.0 and 1.1 by default. Other parameters in the baselines are set to their default values. Besides, the training targets of all baselines are replaced by the aggregated crowdsourced label $\boldsymbol{s}$ except ML-KNN, MV, and DoctorNet.

For performance evaluations, we adopt three widely used multi-label metrics, namely example-based F1 (example-F1), micro-averaged F1 (micro-F1), and macro-averaged F1 (macro-F1) (Zhang & Zhou, 2014; Chen et al., 2019a; Bai et al., 2020). Note that for all these metrics, the higher the better. For all the experiments, we perform ten-fold cross-validation and report the mean as well as the standard deviation for metric values.

5.2. Main Results

The comparison results of CLEAR on three CMLL scenarios are shown in Table 1, 2, and 3, where the best results are

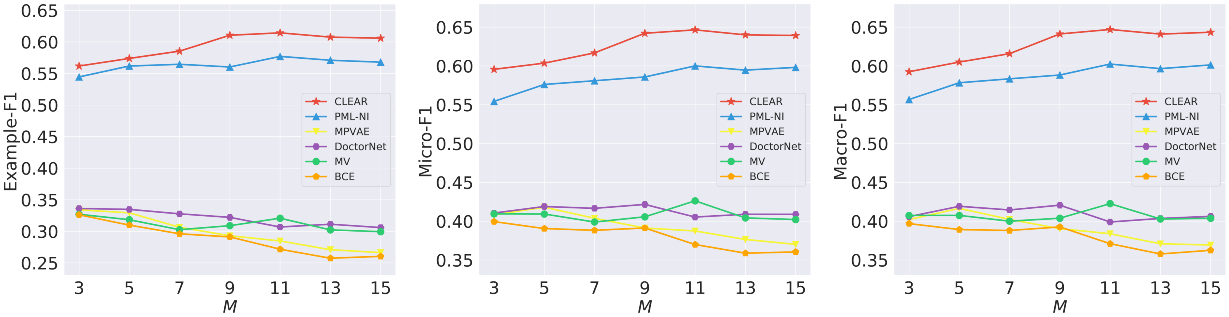

Figure 4. Comparison results of different numbers of annotators on *Image* in the setting of $\bar{T}_{01}^k = 0.2$ and $\bar{T}_{10}^k = 0.5$.

shown in boldface. Overall, our proposed method outperforms all baselines on three metrics on most CMLL datasets and scenarios. For example, in the setting of $\bar{T}_{01}^k = 0.2$ and $\bar{T}_{10}^k = 0.5$ on the Mirflickr dataset, CLEAR improves the best baseline by a notable margin of **28.64%**, **25.40%**, and **21.69%** on Example-F1, Micro-F1, and Macro-F1 respectively. The superior results against various types of baselines imply that CLEAR can effectively tackle the CMLL task.

Specifically, CLEAR improves BCE on average by 12.49%, 11.89%, and 10.03% on Example-F1, Micro-F1, and Macro-F1 respectively which significantly proves the effectiveness of our proposed method. Moreover, CLEAR outperforms the crowdsourcing-based approaches, i.e. MV and DoctorNet, especially when the corruption probability is large. This is because MV naively trusts the false majority voting labels and DoctorNet overfits a large number of error labels when most labels are incorrectly tagged. Besides, these two methods lack the consideration of label dependencies. Furthermore, CLEAR achieves better results compared to the MLL algorithm, i.e., ML-KNN and MPVAE, which shows the robustness of CLEAR when handling unreliable crowdsourcing information. Our method also outperforms PML-NI, which is designed for solving redundant labels.

5.3. Additional Experiments

Ablation Analysis. To show how the proposed unbiased risk estimator and the decoupled autoencoder influence CLEAR respectively, we conduct comparison on two variants of CLEAR: 1) *CLEAR-B*, which implements BCE loss as the reconstruction loss instead of unbiased loss; 2) *CLEAR-M*, which individually trains each class predictor by the unbiased loss, without considering label relationships. For CLEAR-M, we implement a three-layer fully connected network with the same hidden sizes as CLEAR. Figure 3 demonstrates the comparison results on the three datasets. In general, CLEAR consistently achieves the best performance compared to the two variants. These results clearly verify the superiority of our proposed unbiased loss and the decoupled autoencoder framework.

Table 4. Results of estimating transition matrices of CLEAR with different strategies. Note that the smaller the mean absolute error is the better, and the best results are shown in boldface.

Dataset	$\bar{T}_{01}^k/\bar{T}_{10}^k$	T-max	S-1	S-20
Image	0.2/0.2	0.55	0.34	**0.21**
	0.2/0.5	0.31	0.66	**0.22**
	0.5/0.2	0.33	0.74	**0.27**
Scene	0.2/0.2	0.61	0.47	**0.30**
	0.2/0.5	0.30	0.63	**0.20**
	0.5/0.2	0.34	0.57	**0.25**
Corel5K	0.2/0.2	0.73	0.73	**0.57**
	0.2/0.5	0.33	0.35	**0.16**
	0.5/0.2	0.60	0.63	**0.59**
Mirflickr	0.2/0.2	0.52	0.39	**0.15**
	0.2/0.5	0.25	0.57	**0.12**
	0.5/0.2	0.38	0.39	**0.26**
NUS-WIDE	0.2/0.2	0.63	0.69	**0.54**
	0.2/0.5	0.25	0.34	**0.17**
	0.5/0.2	0.58	0.60	**0.57**

Effect of Annotator Number. To investigate how the number of annotators affects CLEAR, we further explore the performance of our method as well as the competitive baselines on a wide range of the annotator number M on the Image dataset when $\bar{T}_{01}^k = 0.2$ and $\bar{T}_{10}^k = 0.5$, where $M \in \{3, 5, 7, 9, 11, 13, 15\}$. As shown in Figure 4, CLEAR achieves the best results by beating all baselines on all settings on Example-F1, Micro-F1, and Macro-F1. In addition, as the number of annotators increases, the result of CLEAR improves in general while most other baselines decrease. This indicates that CLEAR can benefit from the growth of the annotation number, even when heavy noise exists on crowdsourced labels.

Results of Estimating Transition Matrices. Moreover, we evaluate the transition matrix estimation results of CLEAR under different estimation strategies. Specifically, *T-max* uses the model-predicted probability of the most confident sample to estimate the transition matrices following (Patrini et al., 2017; Xia et al., 2019; Li et al., 2022).

S-C conducts approximation by averaging the top-C confident crowdsourced labels, as mentioned in section 4.1. We use the mean absolute error to measure the estimation results, i.e., $\frac{1}{K}\sum_{k=1}^{K}\|\hat{\bar{\boldsymbol{T}}}^k - \bar{\boldsymbol{T}}^k\|_1$, where $\hat{\bar{\boldsymbol{T}}}^k$ and $\bar{\boldsymbol{T}}^k$ denote the estimated and ground-truth transition matrices respectively. As shown in Table 4, our proposed estimation strategy achieves the best results.

6. Conclusion

In this work, we study the CMLL problem which aims to learn a robust multi-label predictor given crowdsourcing labels. We establish the first unbiased risk estimator under the CMLL and upgrade it by integrating the annotations while ensuring theoretical characteristics. We then exploit label correlations by proposing a decoupled autoencoder framework. Experiments on various CMLL settings verify the effectiveness of our algorithm. Note that our study resolves the data annotation issue in MLL by greatly reducing the labeling cost while ensuring the robustness of the learner.

Acknowledgements

This work is majorly supported by the Pioneer R&D Program of Zhejiang (No. 2024C01035) and CCF-NetEase ThunderFire Innovation Research Funding (NO. 202305). This work is also supported by Netease Youling Crowdsourcing Platform. Gengyu Lyu would like to thank the National Natural Science Foundation of China (No. 62306020).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Albarqouni, S., Baur, C., Achilles, F., Belagiannis, V., Demirci, S., and Navab, N. Aggnet: Deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE Trans. Medical Imaging*, 35(5):1313–1321, 2016.

Bai, J., Kong, S., and Gomes, C. P. Disentangled variational autoencoder based multi-label classification with covariance-aware multivariate probit model. In *IJCAI*, pp. 4313–4321. ijcai.org, 2020.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, 2002.

Boucheron, S., Lugosi, G., and Massart, P. *Concentration Inequalities - A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Chen, C., Wang, H., Liu, W., Zhao, X., Hu, T., and Chen, G. Two-stage label embedding via neural factorization machine for multi-label classification. In *AAAI*, pp. 3304–3311. AAAI Press, 2019a.

Chen, D., Xue, Y., and Gomes, C. P. End-to-end learning for the deep multivariate probit model. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 931–940. PMLR, 2018.

Chen, Y. and Lin, H. Feature-aware label space dimension reduction for multi-label classification. In *NeurIPS*, pp. 1538–1546, 2012.

Chen, Z., Wei, X., Wang, P., and Guo, Y. Multi-label image recognition with graph convolutional networks. In *CVPR*, pp. 5177–5186. Computer Vision Foundation / IEEE, 2019b.

Chen, Z., Wang, H., Sun, H., Chen, P., Han, T., Liu, X., and Yang, J. Structured probabilistic end-to-end learning from crowds. In *IJCAI*, pp. 1512–1518. ijcai.org, 2020.

Cho, Y., Kim, D., Khan, M. A., and Choo, J. Mining multi-label samples from single positive labels. In *NeurIPS*, 2022.

Cole, E., Aodha, O. M., Lorieul, T., Perona, P., Morris, D., and Jojic, N. Multi-label learning from single positive labels. In *CVPR*, pp. 933–942. Computer Vision Foundation / IEEE, 2021.

Dalvi, N. N., Dasgupta, A., Kumar, R., and Rastogi, V. Aggregating crowdsourced binary ratings. In *WWW*, pp. 285–294. International World Wide Web Conferences Steering Committee / ACM, 2013.

Dawid, A. P. and Skene, A. M. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979.

Ding, Z., Wang, A., Chen, H., Zhang, Q., Liu, P., Bao, Y., Yan, W., and Han, J. Exploring structured semantic prior for multi label recognition with incomplete labels. In *CVPR*, pp. 3398–3407. IEEE, 2023.

Durand, T., Mehrasa, N., and Mori, G. Learning a deep convnet for multi-label classification with partial labels. In *CVPR*, pp. 647–657. IEEE, 2019.

Feng, L., Lv, J., Han, B., Xu, M., Niu, G., Geng, X., An, B., and Sugiyama, M. Provably consistent partial-label learning. In *NeurIPS*, 2020.

Gao, Y., Xu, M., and Zhang, M. Unbiased risk estimator to multi-labeled complementary label learning. In *IJCAI*, pp. 3732–3740. ijcai.org, 2023.

Gao, Z., Sun, F., Yang, M., Ren, S., Xiong, Z., Engeler, M., Burazer, A., Wildling, L., Daniel, L., and Boning, D. S. Learning from multiple annotator noisy labels via sample-wise label fusion. In *ECCV*, volume 13684 of *Lecture Notes in Computer Science*, pp. 407–422. Springer, 2022.

Guan, M. Y., Gulshan, V., Dai, A. M., and Hinton, G. E. Who said what: Modeling individual labelers improves classification. In *AAAI*, pp. 3109–3118. AAAI Press, 2018.

Hu, P., Sun, X., Sclaroff, S., and Saenko, K. Dualcoop++: Fast and effective adaptation to multi-label recognition with limited annotations. *CoRR*, abs/2308.01890, 2023.

Huynh, D. and Elhamifar, E. A shared multi-attention framework for multi-label zero-shot learning. In *CVPR*, pp. 8773–8783. Computer Vision Foundation / IEEE, 2020a.

Huynh, D. and Elhamifar, E. Interactive multi-label CNN learning with partial labels. In *CVPR*, pp. 9420–9429. IEEE, 2020b.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Li, S., Jiang, Y., Chawla, N. V., and Zhou, Z. Multi-label learning from crowds. *IEEE Trans. Knowl. Data Eng.*, 31 (7):1369–1382, 2019.

Li, S., Xia, X., Zhang, H., Zhan, Y., Ge, S., and Liu, T. Estimating noise transition matrix with label correlations for noisy multi-label learning. In *NeurIPS*, 2022.

Li, S., Xia, X., Deng, J., Ge, S., and Liu, T. Transferring annotator- and instance-dependent transition matrix for learning from crowds. *CoRR*, abs/2306.03116, 2023.

Liu, H., Wang, F., Lin, M., Wu, R., Zhu, R., Zhao, S., Wang, K., Lv, T., and Fan, C. Towards long-term annotators: A supervised label aggregation baseline. *CoRR*, abs/2311.14709, 2023.

Liu, T. and Tao, D. Classification with noisy labels by importance reweighting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(3):447–461, 2016.

Liu, W., Wang, H., Shen, X., and Tsang, I. W. The emerging trends of multi-label learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(11):7955–7974, 2022.

Lyu, G., Feng, S., and Li, Y. Partial multi-label learning via probabilistic graph matching mechanism. In *KDD*, pp. 105–113. ACM, 2020.

Maurer, A. A vector-contraction inequality for rademacher complexities. In *ALT*, volume 9925 of *Lecture Notes in Computer Science*, pp. 3–17, 2016.

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

Patrini, G., Rozza, A., Menon, A. K., Nock, R., and Qu, L. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pp. 2233–2241. IEEE Computer Society, 2017.

Raykar, V. C. and Yu, S. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *J. Mach. Learn. Res.*, 13:491–518, 2012.

Raykar, V. C., Yu, S., Zhao, L. H., Jerebko, A. K., Florin, C., Valadez, G. H., Bogoni, L., and Moy, L. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *ICML*, volume 382 of *ACM International Conference Proceeding Series*, pp. 889–896. ACM, 2009.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. Classifier chains for multi-label classification. *Mach. Learn.*, 85(3): 333–359, 2011.

Ridnik, T., Baruch, E. B., Zamir, N., Noy, A., Friedman, I., Protter, M., and Zelnik-Manor, L. Asymmetric loss for multi-label classification. In *ICCV*, pp. 82–91. IEEE, 2021.

Rodrigues, F. and Pereira, F. C. Deep learning from crowds. In *AAAI*, pp. 1611–1618. AAAI Press, 2018.

Rubin, T. N., Chambers, A., Smyth, P., and Steyvers, M. Statistical topic models for multi-label document classification. *Mach. Learn.*, 88(1-2):157–208, 2012.

Schultheis, E., Wydmuch, M., Babbar, R., and Dembczynski, K. On missing labels, long-tails and propensities in extreme multi-label classification. In *SIGKDD*, pp. 1547–1557. ACM, 2022.

Shi, W., Sheng, V. S., Li, X., and Gu, B. Semi-supervised multi-label learning from crowds via deep sequential generative model. In *SIGKDD*, pp. 1141–1149. ACM, 2020.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. Cheap and fast - but is it good? evaluating non-expert annotations for natural language tasks. In *ACL*, pp. 254–263. ACL, 2008.

Wang, F., Liu, H., Bi, H., Shen, X., Zhu, R., Wu, R., Lin, M., Lv, T., Fan, C., Liu, Q., Huang, Z., and Chen, E. A dataset for the validation of truth inference algorithms suitable for online deployment. *CoRR*, abs/2403.08826, 2024.

Wang, H., Liu, W., Zhao, Y., Zhang, C., Hu, T., and Chen, G. Discriminative and correlative partial multi-label learning. In *IJCAI*, pp. 3691–3697. ijcai.org, 2019.

Wang, H., Chen, C., Liu, W., Chen, K., Hu, T., and Chen, G. Incorporating label embedding and feature augmentation for multi-dimensional classification. In *AAAI*, pp. 6178–6185. AAAI Press, 2020.

Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., and Xu, W. CNN-RNN: A unified framework for multi-label image classification. In *CVPR*, pp. 2285–2294. IEEE Computer Society, 2016.

Wang, L., Liu, Y., Di, H., Qin, C., Sun, G., and Fu, Y. Semi-supervised dual relation learning for multi-label classification. *IEEE Trans. Image Process.*, 30:9125–9135, 2021.

Wei, H., Xie, R., Feng, L., Han, B., and An, B. Deep learning from multiple noisy annotators as a union. *IEEE transactions on neural networks and learning systems*, 2022.

Wei, T., Guo, L., Li, Y., and Gao, W. Learning safe multi-label prediction for weakly labeled data. *Mach. Learn.*, 107(4):703–725, 2018.

Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., and Movellan, J. R. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NeurIPS*, pp. 2035–2043. Curran Associates, Inc., 2009.

Wu, J., Huang, S., and Zhou, Z. Genome-wide protein function prediction through multi-instance multi-label learning. *IEEE ACM Trans. Comput. Biol. Bioinform.*, 11 (5):891–902, 2014.

Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., and Sugiyama, M. Are anchor points really indispensable in label-noise learning? In *NeurIPS*, pp. 6835–6846, 2019.

Xiao, L., Huang, X., Chen, B., and Jing, L. Label-specific document representation for multi-label text classification. In *EMNLP-IJCNLP*, pp. 466–475. Association for Computational Linguistics, 2019.

Xie, M. and Huang, S. Partial multi-label learning. In *AAAI*, pp. 4302–4309. AAAI Press, 2018.

Xie, M. and Huang, S. Partial multi-label learning with noisy label identification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(7):3676–3687, 2022.

Xie, M. and Huang, S. CCMN: A general framework for learning with class-conditional multi-label noise. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(1):154–166, 2023.

Xiong, B., Cochez, M., Nayyeri, M., and Staab, S. Hyperbolic embedding inference for structured multi-label prediction. In *NeurIPS*, 2022.

Xu, N., Liu, Y., and Geng, X. Partial multi-label learning with label distribution. In *AAAI*, pp. 6510–6517. AAAI Press, 2020.

Xu, N., Qiao, C., Lv, J., Geng, X., and Zhang, M. One positive label is sufficient: Single-positive multi-label learning with label enhancement. In *NeurIPS*, 2022.

Yeh, C., Wu, W., Ko, W., and Wang, Y. F. Learning deep latent space for multi-label classification. In *AAAI*, pp. 2838–2844. AAAI Press, 2017.

Zha, Z., Hua, X., Mei, T., Wang, J., Qi, G., and Wang, Z. Joint multi-label multi-instance learning for image classification. In *CVPR*. IEEE Computer Society, 2008.

Zhang, J. and Wu, X. Multi-label inference for crowdsourcing. In *SIGKDD*, pp. 2738–2747. ACM, 2018.

Zhang, M. and Zhou, Z. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognit.*, 40(7): 2038–2048, 2007.

Zhang, M. and Zhou, Z. A review on multi-label learning algorithms. *IEEE Trans. Knowl. Data Eng.*, 26(8):1819–1837, 2014.

Zhang, Y., Chen, X., Zhou, D., and Jordan, M. I. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. *JMLR*, 17:102:1–102:44, 2016.

Zhou, Z.-H. *Ensemble methods: foundations and algorithms*. CRC press, 2012.

Zhu, K. and Wu, J. Residual attention: A simple but effective method for multi-label recognition. In *ICCV*, pp. 184–193. IEEE, 2021.

Zhu, X., Liu, J., Liu, W., Ge, J., Liu, B., and Cao, J. Scene-aware label graph learning for multi-label image classification. In *ICCV*, pp. 1473–1482, 2023.

A. Proof of Theorem 1

Theorem 1. *By decomposing the MLL problem into K independent binary classification problem, i.e., $\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y}) = \sum_{k=1}^{K} \ell(f_k(\boldsymbol{x}), y_k)$, where f_k refers to prediction of the model on k-th class and ℓ is the base loss function. With $\tilde{R}(\boldsymbol{f}) = \mathbb{E}_{P(\boldsymbol{X}, \tilde{\boldsymbol{Y}})} \left[\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}), \{\tilde{\boldsymbol{y}}_m\}_{m=1}^{M})\right]$, and define*

$$\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}), \{\tilde{\boldsymbol{y}}_m\}_{m=1}^{M}) = \frac{1}{2^M} \sum_{k=1}^{K} \sum_{j=0}^{1} \frac{P(Y_k = j | \boldsymbol{X} = \boldsymbol{x})}{\prod_{m=1}^{M} \sum_{i=0}^{1} T_{i\tilde{y}_{mk}}^{mk} P(Y_k = i | \boldsymbol{X} = \boldsymbol{x})} \ell(f_k(\boldsymbol{x}), j), \tag{14}$$

where $\tilde{\boldsymbol{Y}}$ denotes the random variable of the crowdsourced labels for each $\boldsymbol{x}$. Then, $\tilde{R}(\boldsymbol{f})$ is the unbiased risk estimator with respect to $R(\boldsymbol{f})$.

Proof. The multi-label learning risk $R(\boldsymbol{f})$ could be rewritten as

$$\begin{aligned}
R(\boldsymbol{f}) &= \mathbb{E}_{P(\boldsymbol{X}, \boldsymbol{Y})} \left[\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y})\right] \\
&= \sum_{k=1}^{K} \mathbb{E}_{P(\boldsymbol{X}, \boldsymbol{Y}_k)} \left[\ell(f_k(\boldsymbol{x}), y_k)\right] \\
&= \sum_{k=1}^{K} \int_{\boldsymbol{x}} \sum_{j=0}^{1} P(\boldsymbol{X} = \boldsymbol{x}, Y_k = j) \ell(f_k(\boldsymbol{x}), j) \mathrm{d}\boldsymbol{x} \\
&= \sum_{k=1}^{K} \int_{\boldsymbol{x}} \sum_{\{\tilde{y}_{mk}\}_{m=1}^{M} \in \{0,1\}^M} \frac{1}{2^M} \cdot \\
&\qquad \sum_{j=0}^{1} \frac{P(Y_k = j | \boldsymbol{X} = \boldsymbol{x})}{P(\{\tilde{Y}_{mk}\}_{m=1}^{M} = \{\tilde{y}_{mk}\}_{m=1}^{M} | \boldsymbol{X} = \boldsymbol{x})} P(\boldsymbol{X} = \boldsymbol{x}, \{\tilde{Y}_{mk}\}_{m=1}^{M} = \{\tilde{y}_{mk}\}_{m=1}^{M}) \ell(f_k(\boldsymbol{x}), j) \mathrm{d}\boldsymbol{x} \\
&= \sum_{k=1}^{K} \mathbb{E}_{P(\boldsymbol{X}, \{\tilde{Y}_{mk}\}_{m=1}^{M})} \left[\frac{1}{2^M} \sum_{j=0}^{1} \frac{P(Y_k = j | \boldsymbol{X} = \boldsymbol{x})}{P(\{\tilde{Y}_{mk}\}_{m=1}^{M} = \{\tilde{y}_{mk}\}_{m=1}^{M} | \boldsymbol{X} = \boldsymbol{x})} \ell(f_k(\boldsymbol{x}), j)\right] \qquad (15) \\
&= \sum_{k=1}^{K} \mathbb{E}_{P(\boldsymbol{X}, \{\tilde{Y}_{mk}\}_{m=1}^{M})} \left[\frac{1}{2^M} \sum_{j=0}^{1} \frac{P(Y_k = j | \boldsymbol{X} = \boldsymbol{x})}{\prod_{m=1}^{M} P(\tilde{Y}_{mk} = \tilde{y}_{mk} | \boldsymbol{X} = \boldsymbol{x})} \ell(f_k(\boldsymbol{x}), j)\right] \\
&= \sum_{k=1}^{K} \mathbb{E}_{P(\boldsymbol{X}, \{\tilde{Y}_{mk}\}_{m=1}^{M})} \left[\frac{1}{2^M} \sum_{j=0}^{1} \frac{P(Y_k = j | \boldsymbol{X} = \boldsymbol{x})}{\prod_{m=1}^{M} \sum_{i=0}^{1} T_{i\tilde{y}_{mk}}^{mk} P(Y_k = i | \boldsymbol{X} = \boldsymbol{x})} \ell(f_k(\boldsymbol{x}), j)\right] \\
&= \mathbb{E}_{P(\boldsymbol{X}, \tilde{\boldsymbol{Y}})} \left[\frac{1}{2^M} \sum_{k=1}^{K} \sum_{j=0}^{1} \frac{P(Y_k = j | \boldsymbol{X} = \boldsymbol{x})}{\prod_{m=1}^{M} \sum_{i=0}^{1} T_{i\tilde{y}_{mk}}^{mk} P(Y_k = i | \boldsymbol{X} = \boldsymbol{x})} \ell(f_k(\boldsymbol{x}), j)\right] \\
&= \mathbb{E}_{P(\boldsymbol{X}, \tilde{\boldsymbol{Y}})} \left[\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}), \{\tilde{\boldsymbol{y}}_m\}_{m=1}^{M})\right] \\
&= \tilde{R}(\boldsymbol{f}).
\end{aligned}$$

By proving that $\tilde{R}(\boldsymbol{f})$ is equivalent to $R(\boldsymbol{f})$ given the same classifier $\boldsymbol{f}$, we demonstrate our proposed risk estimator $\tilde{R}(\boldsymbol{f})$ guarantees risk consistency.

B. Proof of Theorem 2

Theorem 2. *Let $\tilde{\boldsymbol{y}} = [\tilde{y}_1, \ldots, \tilde{y}_K]$ be the aggregated label vector for each $\boldsymbol{x}$, and $\tilde{Y}_k$ is the random variable of $\tilde{y}_k$. We have the following consequences:*

(Existence) There exist a set of class-dependent instance-independent transition matrices $\{\bar{\boldsymbol{T}}^k\}_{k=1}^{K} \in [0, 1]^{2 \times 2}$ such that $\bar{T}_{ij}^{k} = P(\tilde{Y}_k = j | Y_k = i), \forall i, j \in \{0, 1\}$, the unbiased risk estimator for CMLL with respect to $R(\boldsymbol{f})$ is $\tilde{R}(\boldsymbol{f}) =$

$\mathbb{E}_{P(\boldsymbol{X},\tilde{\boldsymbol{Y}})}\left[\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}),\tilde{\boldsymbol{y}})\right]$, *and*

$$\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}),\tilde{\boldsymbol{y}})=\sum_{k=1}^{K}\left(\frac{P(\tilde{Y}_k=1|\boldsymbol{X}=\boldsymbol{x})-\bar{T}_{01}^{k}}{1-\bar{T}_{01}^{k}-\bar{T}_{10}^{k}}\ell(f_k(\boldsymbol{x}),1)+\frac{P(\tilde{Y}_k=0|\boldsymbol{X}=\boldsymbol{x})-\bar{T}_{10}^{k}}{1-\bar{T}_{01}^{k}-\bar{T}_{10}^{k}}\ell(f_k(\boldsymbol{x}),0)\right) \tag{16}$$

(Formulation of Transition Matrices) Let A be the random variable of the index of the annotator. By denoting $\omega^m=P(A=m|\boldsymbol{X}=\boldsymbol{x})$ as the contribution of m-th annotator on tagging $\boldsymbol{x}$, and with $\boldsymbol{T}^{mk}$ defined in subsection 3.2, the transition matrices $\{\bar{\boldsymbol{T}}^k\}_{k=1}^K$ for aggregated labels are formalized as linear combinations of $\boldsymbol{T}^{mk}$:

$$\bar{\boldsymbol{T}}^k=\sum_{m=1}^{M}\omega^m\cdot\boldsymbol{T}^{mk}, \tag{17}$$

which are class-dependent and instance-independent.

Proof. We first detail the proof of the existence and the formulation of the aggregated transition matrices $\{\bar{\boldsymbol{T}}^k\}_{k=1}^K$. Assuming there exists a set of class-dependent instance-independent transition matrices $\{\bar{\boldsymbol{T}}^k\}_{k=1}^K$, we have $P(\tilde{Y}_k=j|\boldsymbol{X}=\boldsymbol{x})=\sum_{i=0}^{1}\bar{T}_{ij}^{k}P(Y_k=i|\boldsymbol{X}=\boldsymbol{x})$, which is similar to the data generation process discussed in subsection 3.2. Also,

$$\begin{aligned}P(\tilde{Y}_k=j|\boldsymbol{X}=\boldsymbol{x})&=\sum_{m=1}^{M}P(\tilde{Y}_k=j,A=m|\boldsymbol{X}=\boldsymbol{x})\\&=\sum_{m=1}^{M}P(A=m|\boldsymbol{X}=\boldsymbol{x})P(\tilde{Y}_k=j|A=m,\boldsymbol{X}=\boldsymbol{x})\\&=\sum_{m=1}^{M}\omega^m P(\tilde{Y}_{mk}=j|\boldsymbol{X}=\boldsymbol{x})\\&=\sum_{m=1}^{M}\omega^m\sum_{i=0}^{1}T_{ij}^{mk}P(Y_k=i|\boldsymbol{X}=\boldsymbol{x})\\&=\sum_{i=0}^{1}(\sum_{m=1}^{M}\omega^m\cdot T_{ij}^{mk})P(Y_k=i|\boldsymbol{X}=\boldsymbol{x}).\end{aligned} \tag{18}$$

Thus, $\{\bar{\boldsymbol{T}}^k\}_{k=1}^K$ exist and $\bar{\boldsymbol{T}}^k=\sum_{m=1}^{M}\omega^m\cdot\boldsymbol{T}^{mk}$. Then, the unbiased risk estimator is derived by:

$$\begin{aligned}R(\boldsymbol{f})&=\mathbb{E}_{P(\boldsymbol{X},\boldsymbol{Y})}\left[\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{y})\right]\\&=\sum_{k=1}^{K}\mathbb{E}_{P(\boldsymbol{X},\boldsymbol{Y}_k)}\left[\ell(f_k(\boldsymbol{x}),y_k)\right]\\&=\sum_{k=1}^{K}\int_{\boldsymbol{x}}\sum_{j=0}^{1}P(\boldsymbol{X}=\boldsymbol{x})P(Y_k=j|\boldsymbol{X}=\boldsymbol{x})\ell(f_k(\boldsymbol{x}),j)\mathrm{d}\boldsymbol{x}\\&=\sum_{k=1}^{K}\mathbb{E}_{P(\boldsymbol{X})}\left[(P(Y_k=1|\boldsymbol{X}=\boldsymbol{x})\ell(f_k(\boldsymbol{x}),1)+P(Y_k=0|\boldsymbol{X}=\boldsymbol{x})\ell(f_k(\boldsymbol{x}),0))\right]\\&=\mathbb{E}_{P(\boldsymbol{X},\tilde{\boldsymbol{Y}})}\left[\sum_{k=1}^{K}\left(\frac{P(\tilde{Y}_k=1|\boldsymbol{X}=\boldsymbol{x})-\bar{T}_{01}^{k}}{1-\bar{T}_{01}^{k}-\bar{T}_{10}^{k}}\ell(f_k(\boldsymbol{x}),1)+\frac{P(\tilde{Y}_k=0|\boldsymbol{X}=\boldsymbol{x})-\bar{T}_{10}^{k}}{1-\bar{T}_{01}^{k}-\bar{T}_{10}^{k}}\ell(f_k(\boldsymbol{x}),0)\right)\right]\\&=\mathbb{E}_{P(\boldsymbol{X},\tilde{\boldsymbol{Y}})}\left[\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}),\tilde{\boldsymbol{y}})\right]\\&=\tilde{R}(\boldsymbol{f}).\end{aligned} \tag{19}$$

Note that the third last equation holds because, with $P(\tilde{Y}_k = j|\boldsymbol{X} = \boldsymbol{x}) = \sum_{i=0}^{1} \bar{T}_{ij}^k P(Y_k = i|\boldsymbol{X} = \boldsymbol{x})$, we have:

$$\begin{aligned} P(\tilde{Y}_k = 1|\boldsymbol{X} = \boldsymbol{x}) &= \bar{T}_{01}^k P(Y_k = 0|\boldsymbol{X} = \boldsymbol{x}) + \bar{T}_{11}^k P(Y_k = 1|\boldsymbol{X} = \boldsymbol{x}) \\ &= \bar{T}_{01}^k (1 - P(Y_k = 1|\boldsymbol{X} = \boldsymbol{x})) + (1 - \bar{T}_{10}^k) P(Y_k = 1|\boldsymbol{X} = \boldsymbol{x}) \\ &= \bar{T}_{01}^k + (1 - \bar{T}_{01}^k - \bar{T}_{10}^k) P(Y_k = 1|\boldsymbol{X} = \boldsymbol{x}), \end{aligned} \tag{20}$$

thus $P(Y_k = 1|\boldsymbol{X} = \boldsymbol{x}) = \frac{P(\tilde{Y}_k=1|\boldsymbol{X}=\boldsymbol{x})-\bar{T}_{01}^k}{1-\bar{T}_{01}^k-\bar{T}_{10}^k}$, and similarly $P(Y_k = 0|\boldsymbol{X} = \boldsymbol{x}) = \frac{P(\tilde{Y}_k=0|\boldsymbol{X}=\boldsymbol{x})-\bar{T}_{10}^k}{1-\bar{T}_{01}^k-\bar{T}_{10}^k}$.

C. Proof of Theorem 3

Theorem 3. *Assume that the true aggregated transition matrices $\{\bar{\boldsymbol{T}}^k\}_{k=1}^K$ are given, and the loss function $\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{s})$ is L_T-Lipschitz continuous with respect to $\boldsymbol{f}(\boldsymbol{x})$, and the base loss function l is upper-bounded by λ. Let $\mu = \max_k \frac{1}{1-T_{01}^k-T_{10}^k}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, we have*

$$\mathbb{E}[\bar{R}(\hat{\boldsymbol{f}})] - \bar{R}(\hat{\boldsymbol{f}}) \leq 2\sqrt{2} L_T \sum_{k=1}^{K} \mathfrak{R}_n(\mathcal{H}_k) + \lambda K(\mu + 1)\sqrt{\frac{\log(1/\delta)}{2n}}. \tag{21}$$

Recall that the empirical risk is defined by $\bar{R}(\boldsymbol{f}) = \frac{1}{n}\sum_{i=1}^{n} \tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{s}_i)$, and

$$\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{s}) = \sum_{k=1}^{K} \left(\frac{\left[s_k - \bar{T}_{01}^k\right]_+}{1 - \bar{T}_{01}^k - \bar{T}_{10}^k} \ell(f_k(\boldsymbol{x}), 1) + \frac{\left[(1 - s_k) - \bar{T}_{10}^k\right]_+}{1 - \bar{T}_{01}^k - \bar{T}_{10}^k} \ell(f_k(\boldsymbol{x}), 0) \right). \tag{22}$$

Let S and S' be two crowdsourced datasets that exactly differ by the i-th example, i.e.,

$$\begin{aligned} S &= \{(\boldsymbol{x}_1, \boldsymbol{s}_1), \ldots, (\boldsymbol{x}_i, \boldsymbol{s}_i), \ldots, (\boldsymbol{x}_n, \boldsymbol{s}_n)\}, \\ S' &= \{(\boldsymbol{x}_1, \boldsymbol{s}_1), \ldots, (\boldsymbol{x}_i', \boldsymbol{s}_i'), \ldots, (\boldsymbol{x}_n, \boldsymbol{s}_n)\}, \end{aligned} \tag{23}$$

and denote the function Φ as:

$$\Phi(S) = \sup_{\boldsymbol{f} \in \mathcal{F}} \left(\mathbb{E}\left[\bar{R}(\boldsymbol{f})\right] - \bar{R}(\boldsymbol{f}) \right) \tag{24}$$

where the generalization risk $\mathbb{E}\left[\bar{R}(\boldsymbol{f})\right]$ is equivalent to $\mathbb{E}_{P(\boldsymbol{X}, \tilde{\boldsymbol{Y}})}\left[\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{s}_i)\right] = \tilde{R}(\boldsymbol{f})$ and the empirical risk $\bar{R}(\boldsymbol{f})$ is equivalent to $\hat{\mathbb{E}}_S\left[\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{s}_i)\right]$. The proof of Theorem 3 is mainly composed of the following two lemmas.

Lemma 1. *Let $\hat{\boldsymbol{f}}$ be the minimizer of the empirical risk $\bar{R}(\boldsymbol{f})$, and $\mathop{\mathbb{E}}_{S}[\Phi(S)]$ is the expectation of $\Phi(S)$ over all S drawn from the data distribution. With the base loss function l upper-bounded by λ and $\mu = \max_k \frac{1}{1-T_{01}^k-T_{10}^k}$, for any $\delta > 0$, with probability at least $1 - \delta$, we have*

$$\mathbb{E}[\bar{R}(\hat{\boldsymbol{f}})] - \bar{R}(\hat{\boldsymbol{f}}) \leq \mathop{\mathbb{E}}_{S}[\Phi(S)] + \lambda K(\mu + 1)\sqrt{\frac{\log(1/\delta)}{2n}}. \tag{25}$$

Proof. To apply McDiarmid's inequality (Boucheron et al., 2013) to prove the lemma, we first check the bounded difference property of $\Phi(S)$ by

$$\begin{aligned}
\Phi(S)-\Phi(S') &\leq \sup_{\boldsymbol{f}\in\mathcal{F}} \frac{1}{n}\left(\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}_i),\boldsymbol{s}_i)-\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}'_i),\boldsymbol{s}'_i)\right)\\
&= \sup_{\boldsymbol{f}\in\mathcal{F}} \frac{1}{n}\sum_{k=1}^{K}\left(\frac{\left[s_k-\bar{T}^k_{01}\right]_+}{1-\bar{T}^k_{01}-\bar{T}^k_{10}}\ell(f_k(\boldsymbol{x}_i),1)-\frac{\left[s'_k-\bar{T}^k_{01}\right]_+}{1-\bar{T}^k_{01}-\bar{T}^k_{10}}\ell(f_k(\boldsymbol{x}'_i),1)\right.\\
&\qquad\left.+\frac{\left[(1-s_k)-\bar{T}^k_{10}\right]_+}{1-\bar{T}^k_{01}-\bar{T}^k_{10}}\ell(f_k(\boldsymbol{x}_i),0)-\frac{\left[(1-s'_k)-\bar{T}^k_{10}\right]_+}{1-\bar{T}^k_{01}-\bar{T}^k_{10}}\ell(f_k(\boldsymbol{x}'_i),0)\right)\\
&\leq \sup_{\boldsymbol{f}\in\mathcal{F}} \frac{1}{n}\sum_{k=1}^{K}\left(\frac{1-\bar{T}^k_{01}}{1-\bar{T}^k_{01}-\bar{T}^k_{10}}\ell(f_k(\boldsymbol{x}_i),1)-\frac{1-\bar{T}^k_{01}}{1-\bar{T}^k_{01}-\bar{T}^k_{10}}\ell(f_k(\boldsymbol{x}'_i),1)\right.\\
&\qquad\left.+\frac{1-\bar{T}^k_{10}}{1-\bar{T}^k_{01}-\bar{T}^k_{10}}\ell(f_k(\boldsymbol{x}_i),0)-\frac{1-\bar{T}^k_{10}}{1-\bar{T}^k_{01}-\bar{T}^k_{10}}\ell(f_k(\boldsymbol{x}'_i),0)\right)\\
&\leq \frac{1}{n}\sum_{k=1}^{K}\left(\frac{1-\bar{T}^k_{01}}{1-\bar{T}^k_{01}-\bar{T}^k_{10}}\cdot\lambda+\frac{1-\bar{T}^k_{10}}{1-\bar{T}^k_{01}-\bar{T}^k_{10}}\cdot\lambda\right)\\
&= \frac{\lambda}{n}\sum_{k=1}^{K}\left(\frac{1}{1-\bar{T}^k_{01}-\bar{T}^k_{10}}+1\right)\\
&\leq \frac{\lambda K(\mu+1)}{n}.
\end{aligned} \tag{26}$$

Similarly, we can obtain $\Phi(S')-\Phi(S)\leq\frac{\lambda K(\mu+1)}{n}$ and thus $|\Phi(S)-\Phi(S')|\leq\frac{\lambda K(\mu+1)}{n}$. Then, by McDiarmid's inequality, for any $\delta>0$, with probability at least $1-\delta$, we have

$$\Phi(S)\leq\mathop{\mathbb{E}}_{S}[\Phi(S)]+\lambda K(\mu+1)\sqrt{\frac{\log(1/\delta)}{2n}}. \tag{27}$$

Besides, we have $\mathbb{E}[\bar{R}(\hat{\boldsymbol{f}})]-\bar{R}(\hat{\boldsymbol{f}})\leq\sup_{\boldsymbol{f}\in\mathcal{F}}\left(\mathbb{E}\left[\bar{R}(\boldsymbol{f})\right]-\bar{R}(\boldsymbol{f})\right)=\Phi(S)$, which complete the proof. Then, we give an upper bound of $\mathop{\mathbb{E}}_{S}[\Phi(S)]$ in the following lemma.

Lemma 2. *Denote $\mathcal{F}$ as the hypothesis class and $\mathcal{H}_k=\{h:\boldsymbol{x}\mapsto f_k(\boldsymbol{x})|\boldsymbol{f}\in\mathcal{F}\}$ as the functional space for the k-th class and let $\mathfrak{R}_n(\mathcal{H}_k)$ be the expected Rademacher complexity of $\mathcal{H}_k$ with sample size n. Assuming the loss function $\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{s})$ is L_T-Lipschitz continuous with respect to $\boldsymbol{f}(\boldsymbol{x})$, then we have*

$$\mathop{\mathbb{E}}_{S}[\Phi(S)]\leq 2\sqrt{2}L_T\sum_{k=1}^{K}\mathfrak{R}_n(\mathcal{H}_k) \tag{28}$$

Proof. Note that the Rademacher complexity of $\mathcal{H}_k$ is formalized as $\mathfrak{R}_n(\mathcal{H}_k)=\mathop{\mathbb{E}}_{\boldsymbol{\sigma},S}\left[\sup_{h\in\mathcal{H}_k}\frac{1}{n}\sum_{i=1}^{n}\sigma_i h(\boldsymbol{x}_i)\right]$, where $\boldsymbol{\sigma}=[\sigma_1,\ldots,\sigma_n]$ are i.i.d. Rademacher random variables. By abbreviating $\mathbb{E}_{P(\boldsymbol{X},\tilde{\boldsymbol{Y}})}\left[\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}_i),\boldsymbol{s}_i)\right]$ as $\mathbb{E}[\tilde{\mathcal{L}}\circ\boldsymbol{f}]$, and denoting the base loss function of the unbiased risk as $\tilde{\ell}$, we bound $\mathop{\mathbb{E}}_{S}[\Phi(S)]$ by the following derivations:

$$\begin{aligned}
\mathop{\mathbb{E}}_{S}[\Phi(S)] &= \mathop{\mathbb{E}}_{S}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\Big(\mathbb{E}[\tilde{\mathcal{L}}\circ\boldsymbol{f}] - \hat{\mathbb{E}}_S[\tilde{\mathcal{L}}\circ\boldsymbol{f}]\Big)\right] \\
&= \mathop{\mathbb{E}}_{S}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\mathop{\mathbb{E}}_{S'}\Big[\hat{\mathbb{E}}_{S'}[\tilde{\mathcal{L}}\circ\boldsymbol{f}] - \hat{\mathbb{E}}_S[\tilde{\mathcal{L}}\circ\boldsymbol{f}]\Big]\right] \\
&\leq \mathop{\mathbb{E}}_{S,S'}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\Big(\hat{\mathbb{E}}_{S'}[\tilde{\mathcal{L}}\circ\boldsymbol{f}] - \hat{\mathbb{E}}_S[\tilde{\mathcal{L}}\circ\boldsymbol{f}]\Big)\right] \\
&= \mathop{\mathbb{E}}_{S,S'}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\Big(\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}_i'),\boldsymbol{s}_i') - \tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}_i),\boldsymbol{s}_i)\Big)\right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{\sigma},S,S'}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\Big(\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}_i'),\boldsymbol{s}_i') - \tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}_i),\boldsymbol{s}_i)\Big)\right] \\
&\leq \mathop{\mathbb{E}}_{\boldsymbol{\sigma},S'}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}_i'),\boldsymbol{s}_i')\right] + \mathop{\mathbb{E}}_{\boldsymbol{\sigma},S}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}-\sigma_i\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}_i),\boldsymbol{s}_i)\right] \\
&= 2\mathop{\mathbb{E}}_{\boldsymbol{\sigma},S}\left[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\tilde{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}_i),\boldsymbol{s}_i)\right] \\
&\leq 2\sqrt{2}L_T\sum_{k=1}^{K}\mathop{\mathbb{E}}_{\boldsymbol{\sigma},S}\left[\sup_{h\in\mathcal{H}_k}\frac{1}{n}\sum_{i=1}^{n}\sigma_i h(\boldsymbol{x}_i)\right] \\
&= 2\sqrt{2}L_T\sum_{k=1}^{K}\mathfrak{R}_n(\mathcal{H}_k).
\end{aligned} \tag{29}$$

The second inequality holds due to the sub-additivity of the supremum function, and the last inequality holds because of the Rademacher vector contraction inequality (Maurer, 2016). Theorem 3 follows by combining Lemma 1 and Lemma 2.

Extension of Theorem 3. The proposal of Theorem 3 ensures the empirical risk minimizer approximately approaches its population minimizer counterpart. Further, with the uniform convergence of $\mathbb{E}\left[\bar{R}(\boldsymbol{f})\right] - \bar{R}(\boldsymbol{f})$, we extend our theoretical guarantees by proposing Theorem 4, which demonstrates that the empirical risk minimizer converges to the true risk minimizer as $n \to \infty$.

Theorem 4. *Let $\hat{\boldsymbol{f}}$ and $\boldsymbol{f}^*$ be the minimizer of $\bar{R}(\boldsymbol{f})$ and of $R(\boldsymbol{f})$ respectively. With the conditions in Theorem 3, for any $\delta > 0$, with probability at least $1-\delta$, we have*

$$R(\hat{\boldsymbol{f}}) - R(\boldsymbol{f}^*) \leq 4\sqrt{2}L_T\sum_{k=1}^{K}\mathfrak{R}_n(\mathcal{H}_k) + 2\lambda K(\mu+1)\sqrt{\frac{\log(1/\delta)}{2n}}. \tag{30}$$

Proof. We first bound the left-hand side by the following derivation:

$$\begin{aligned}
R(\hat{\boldsymbol{f}}) - R(\boldsymbol{f}^*) &= \tilde{R}(\hat{\boldsymbol{f}}) - \tilde{R}(\boldsymbol{f}^*) \\
&= [\tilde{R}(\hat{\boldsymbol{f}}) - \bar{R}(\hat{\boldsymbol{f}})] + [\bar{R}(\hat{\boldsymbol{f}}) - \bar{R}(\boldsymbol{f}^*)] + [\bar{R}(\boldsymbol{f}^*) - \tilde{R}(\boldsymbol{f}^*)] \\
&\leq [\tilde{R}(\hat{\boldsymbol{f}}) - \bar{R}(\hat{\boldsymbol{f}})] + 0 + [\bar{R}(\boldsymbol{f}^*) - \tilde{R}(\boldsymbol{f}^*)] \\
&\leq 2\sup_{\boldsymbol{f}\in\mathcal{F}}|\tilde{R}(\boldsymbol{f}) - \bar{R}(\boldsymbol{f})|,
\end{aligned} \tag{31}$$

where the first equation holds due to the risk consistency of the estimator, and the first inequality holds since $\hat{\boldsymbol{f}}$ is defined by the minimizer of $\bar{R}(\boldsymbol{f})$. Using the same trick in Lemma 1, we can derive the same bound for $\sup_{\boldsymbol{f}\in\mathcal{F}}\left(\bar{R}(\boldsymbol{f}) - \mathbb{E}\left[\bar{R}(\boldsymbol{f})\right]\right)$ with

$\sup_{\boldsymbol{f}\in\mathcal{F}} \left(\mathbb{E}\left[\bar{R}(\boldsymbol{f})\right] - \bar{R}(\boldsymbol{f})\right)$. Then, combining with Lemma 2, we have

$$\sup_{\boldsymbol{f}\in\mathcal{F}} \left|\mathbb{E}\left[\bar{R}(\boldsymbol{f})\right] - \bar{R}(\boldsymbol{f})\right| \leq 2\sqrt{2}L_T \sum_{k=1}^{K} \mathfrak{R}_n(\mathcal{H}_k) + \lambda K(\mu+1)\sqrt{\frac{\log(1/\delta)}{2n}}. \tag{32}$$

Note that $\mathbb{E}\left[\bar{R}(\boldsymbol{f})\right]$ is equivalent to $\tilde{R}(\boldsymbol{f})$, thus

$$R(\hat{\boldsymbol{f}}) - R(\boldsymbol{f}^*) \leq 2\sup_{\boldsymbol{f}\in\mathcal{F}} \left|\tilde{R}(\boldsymbol{f}) - \bar{R}(\boldsymbol{f})\right| \leq 4\sqrt{2}L_T \sum_{k=1}^{K} \mathfrak{R}_n(\mathcal{H}_k) + 2\lambda K(\mu+1)\sqrt{\frac{\log(1/\delta)}{2n}}. \tag{33}$$

D. Pseudo-Code of Transition Matrix Estimation

Algorithm 2 Pseudo-code of Transition Matrix Estimation.

```
Input: Crowdsourced multi-label dataset $\tilde{\mathcal{D}}$, a randomly initialized fully connected network $\hat{\boldsymbol{f}}$
Aggregating the crowdsourced labels by $s_k = \frac{1}{M}\tilde{y}_{mk}$
Train $\hat{\boldsymbol{f}}$ by minimizing $\mathcal{L}_{est} = -\sum_{k=1}^{K} s_k \log \hat{f}_k(\boldsymbol{x}) + (1-s_k)\log(1-\hat{f}_k(\boldsymbol{x}))$ until convergence
for $k$ in $\{1,\dots,K\}$ do
    for $a$ in $\{0,1\}$ do
        Select top-$C$ samples in $\tilde{\mathcal{D}}$ with largest value of $a\cdot\hat{f}_k(\boldsymbol{x}) + (1-a)\cdot(1-\hat{f}_k(\boldsymbol{x}))$, denoting them as $\mathcal{A}_{ka}$
        Set $\hat{\bar{T}}^k_{a(1-a)} = \frac{1}{C}\sum_{\boldsymbol{x}\in\mathcal{A}_{ka}}(1-\hat{f}_k(\boldsymbol{x}))$ and $\hat{\bar{T}}^k_{aa} = \frac{1}{C}\sum_{\boldsymbol{x}\in\mathcal{A}_{ka}}\hat{f}_k(\boldsymbol{x})$
    end for
end for
Output: Estimated transition matrices $\{\hat{\bar{\boldsymbol{T}}}^k\}_{k=1}^{K}$
```

E. Complexity Analyses

Let B, D_x, and K denote the batch size, the dimensionality of feature $\boldsymbol{x}$ and the number of classes, and let D_h denote the proxy of the hidden dimensionalities of the encoders and decoders in CLEAR. On the one hand, the time complexity of the feature VAE and label VAE correspond to $\mathcal{O}(BD_xD_h)$ and $\mathcal{O}(BKD_h)$ respectively. On the other hand, the time complexity of the sampling process in the multivariate probit models corresponds to $\mathcal{O}(BSK)$ where S is the sampling number. Thus, the total time complexity of CLEAR is $\mathcal{O}(B(D_xD_h + KD_h + SK))$. Table 5 shows the empirical running time (in seconds) of CLEAR and the deep-model-based baselines, regarding one training epoch, which shows that CLEAR is in the same magnitude as baseline methods.

Table 5. Running time (in seconds) of one training epoch of deep-model-based approach.

Dataset	BCE	MV	DoctorNet	MPVAE	CLEAR
Image	1.04	1.04	1.15	1.85	2.06
Scene	1.09	1.07	1.22	2.06	2.47
Corel5K	1.41	1.40	1.94	6.96	7.82
Mirflickr	1.64	1.60	2.12	6.08	7.63
NUS-WIDE	24.24	24.55	26.30	31.33	32.94

F. Standard Deviation

In this paper, we conduct ten-fold cross-validation for all the experiments, and only the mean metric values are reported in the main paper since the page is limited. In this section, we further report the standard deviations of CLEAR and baselines in Table 6, 7, and 8, which demonstrates the robustness of CLEAR.

Table 6. Standard deviations of CLEAR and baselines when $\bar{T}_{01}^{k} = 0.2$ and $\bar{T}_{10}^{k} = 0.2$.

Metric	Dataset	BCE	MV	DoctorNet	ML-KNN	MPVAE	PML-NI	CLEAR
Example-F1	Image	0.0389	0.0358	0.0232	0.0416	0.0360	0.0287	0.0370
	Scene	0.0279	0.0260	0.0272	0.0431	0.0312	0.0146	0.0245
	Corel5K	0.0032	0.0067	0.0142	0.0051	0.0066	0.0130	0.0110
	Mirflickr	0.0097	0.0094	0.0107	0.0123	0.0087	0.0324	0.0094
	NUS-WIDE	0.0013	0.0044	0.0017	0.0075	0.0012	0.0035	0.0082
Micro-F1	Image	0.0350	0.0334	0.0277	0.0355	0.0285	0.0256	0.0343
	Scene	0.0214	0.0223	0.0261	0.0402	0.0314	0.0130	0.0212
	Corel5K	0.0032	0.0067	0.0149	0.0070	0.0066	0.0185	0.0136
	Mirflickr	0.0082	0.0063	0.0068	0.0096	0.0057	0.0368	0.0075
	NUS-WIDE	0.0028	0.0065	0.0041	0.0126	0.0055	0.0014	0.0100
Macro-F1	Image	0.0328	0.0323	0.0271	0.0341	0.0281	0.0293	0.0342
	Scene	0.0200	0.0226	0.0259	0.0377	0.0305	0.0139	0.0202
	Corel5K	0.0018	0.0014	0.0006	0.0014	0.0030	0.0031	0.0023
	Mirflickr	0.0165	0.0110	0.0150	0.0187	0.0108	0.0404	0.0154
	NUS-WIDE	0.0004	0.0016	0.0025	0.0011	0.0055	0.0004	0.0067

Table 7. Standard deviations of CLEAR and baselines when $\bar{T}_{01}^{k} = 0.2$ and $\bar{T}_{10}^{k} = 0.5$.

Metric	Dataset	BCE	MV	DoctorNet	ML-KNN	MPVAE	PML-NI	CLEAR
Example-F1	Image	0.0343	0.0475	0.0282	0.0164	0.0344	0.0376	0.0467
	Scene	0.0309	0.0312	0.0253	0.0178	0.0295	0.0236	0.0251
	Corel5K	0.0309	0.0312	0.0253	0.0178	0.0295	0.0236	0.0251
	Mirflickr	0.0119	0.0084	0.0134	0.0341	0.0166	0.0141	0.0174
	NUS-WIDE	0.0007	0.0079	0.0054	0.0000	0.0035	0.0260	0.0029
Micro-F1	Image	0.0369	0.0477	0.0282	0.0259	0.0367	0.0328	0.0459
	Scene	0.0334	0.0332	0.0248	0.0279	0.0344	0.0213	0.0242
	Corel5K	0.0182	0.0048	0.0070	0.0008	0.0038	0.0090	0.0056
	Mirflickr	0.0185	0.0153	0.0162	0.0519	0.0207	0.0121	0.0104
	NUS-WIDE	0.0017	0.0106	0.0085	0.0000	0.0065	0.0197	0.0116
Macro-F1	Image	0.0355	0.0469	0.0243	0.0243	0.0370	0.0336	0.0430
	Scene	0.0307	0.0321	0.0262	0.0201	0.0355	0.0212	0.0204
	Corel5K	0.0099	0.0015	0.0005	0.0004	0.0023	0.0020	0.0037
	Mirflickr	0.0230	0.0272	0.0126	0.0246	0.0252	0.0069	0.0256
	NUS-WIDE	0.0001	0.0010	0.0005	0.0006	0.0012	0.0048	0.0013

Table 8. Standard deviations of CLEAR and baselines when $\bar{T}_{01}^{k} = 0.5$ and $\bar{T}_{10}^{k} = 0.2$.

Metric	Dataset	BCE	MV	DoctorNet	ML-KNN	MPVAE	PML-NI	CLEAR
Example-F1	Image	0.0133	0.0139	0.0092	0.0039	0.0091	0.0247	0.0269
	Scene	0.0111	0.0086	0.0148	0.0140	0.0119	0.0039	0.0204
	Corel5K	0.0259	0.0004	0.0003	0.0005	0.0005	0.0026	0.0006
	Mirflickr	0.0106	0.0116	0.0103	0.0091	0.0113	0.0025	0.0374
	NUS-WIDE	0.0008	0.0003	0.0004	0.0008	0.0006	0.0003	0.0027
Micro-F1	Image	0.0121	0.0126	0.0103	0.0041	0.0071	0.0220	0.0236
	Scene	0.0096	0.0082	0.0123	0.0132	0.0102	0.0041	0.0243
	Corel5K	0.0259	0.0004	0.0003	0.0005	0.0005	0.0044	0.0006
	Mirflickr	0.0101	0.0116	0.0100	0.0086	0.0110	0.0025	0.0314
	NUS-WIDE	0.0007	0.0004	0.0005	0.0009	0.0006	0.0003	0.0028
Macro-F1	Image	0.0128	0.0129	0.0120	0.0046	0.0081	0.0219	0.0215
	Scene	0.0102	0.0086	0.0137	0.0167	0.0104	0.0043	0.0208
	Corel5K	0.0180	0.0003	0.0003	0.0004	0.0005	0.0007	0.0004
	Mirflickr	0.0066	0.0078	0.0050	0.0056	0.0067	0.0023	0.0304
	NUS-WIDE	0.0020	0.0007	0.0005	0.0003	0.0006	0.0002	0.0013