Estimating Noise Transition Matrix with Label Correlations for Noisy Multi-Label Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

In label-noise learning, the noise *transition matrix*, bridging the class posterior for noisy and clean data, has been widely exploited to learn *statistically consistent* classifiers. The effectiveness of these algorithms relies heavily on estimating the transition matrix. Recently, the problem of label-noise learning in multi-label classification has received increasing attention, and these consistent algorithms can be applied in multi-label cases. However, the estimation of transition matrices in noisy multi-label learning has not been studied and remains challenging, since most of the existing estimators in noisy multi-class learning depend on the existence of anchor points and the accurate fitting of noisy class posterior. To address this problem, in this paper, we first study the *identifiability problem* of the class-dependent transition matrix in noisy multi-label learning, and then inspired by the identifiability results, we propose a new estimator by exploiting label correlations without both anchor points and accurate fitting of noisy class posterior. Specifically, we estimate the occurrence probability of two noisy labels to get noisy label correlations. Then, we perform *sample selection* to extract side information about clean label correlations, which is used to estimate the occurrence probability of one noisy label when a certain clean label appears. By utilizing the mismatch of label correlations implied in these occurrence probabilities, the transition matrix is *identifiable*, and can then be inferred by solving a simple bilinear decomposition problem. Empirical results illustrate the effectiveness of our estimator to estimate the transition matrix with label correlations, leading to better classification performance.

1 Introduction

In real-world scenarios, an instance is naturally associated with multiple labels, and these labels have complex entangled correlations [5]. Recently, the problem of label-noise learning in multi-label classification has received more and more attention [21, 27, 40], since it is time-consuming and expensive to collect large-scale accurate labels and the noisy labels are much cheap and easy to acquire. In the noisy multi-label setting, the multiple labels assigned to one example may be corrupted simultaneously. That is, any label for each class can be flipped with its respective *transition matrix*, which denotes the transition relationship from clean labels to noisy labels.

Transition matrix has been utilized to build a series of *statistically consistent* algorithms for *noisy multi-class learning* [25, 38]. The main advantage of these consistent algorithms is that they can guarantee to vanish the differences between the classifiers learned from noisy data and the optimal ones from clean data by increasing the size of noisy examples [20, 26, 38, 31].

Fortunately, these statistically consistent algorithms for noisy multi-class learning can also be applied in such noisy multi-label learning with a little modification [40] (more details can be found in Appendix A). However, the effectiveness of these algorithms relies heavily on estimating the transition

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

matrix. Although the estimation of the transition matrix has been investigated in noisy multi-class learning, the estimation of the transition matrix in noisy multi-label learning has not been studied and remains challenging. Specifically, a series of methods [20, 26, 44, 38, 18] has been proposed to estimate the transition matrix for noisy multi-class learning. Nevertheless, most of them assume the existence of anchor points [20, 26, 44], which are defined as the training examples belonging to a particular class surely. However, the assumption is strong and hard to check when we only have noisy data [38]. Also, the methods need to accurately fit the noisy or intermediate class posterior of anchor points, which are rather difficult in multi-label cases, due to severe class imbalance [28].

In this paper, to address the problem of estimating the transition matrix in noisy multi-label learning, we consider utilizing label correlations among noisy multiple labels. Specifically, some label correlations that should *not exist* in practice are included in noisy multi-label learning. For example, "fish" and "water" always co-occur, while "bird" and "sky" always co-occur. But, due to label errors, there is a *slight correlation* between "fish" and "sky", which is impractical. In a high level, we can utilize *the mismatch of label correlations* to identify the transition matrix without both anchor points and accurate fitting of noisy class posterior.

In more detail, we first focus on the identifiability problem of the class-dependent transition matrix in noisy multi-label learning. Accordingly, a new method that estimates the transition matrices by exploiting label correlations is proposed. That is, motivated by the identifiability result that two noisy labels will not suffice to identify the transition matrix in noisy multi-label learning, we utilize *sample selection* to extract useful side information about clean label correlation from noisy data, to achieve the identifiability. Afterward, we not only estimate the *occurrence probability* of two noisy labels on the noisy data, but also of one noisy label when a certain clean label appears on the selected data. By utilizing the mismatch of label correlations implied in these occurrence probabilities, we can prove the identifiability, and transform the problem of estimating the transition matrix using label correlations into the problem of *bilinear decomposition*. Finally, with easy frequency counting, we can get a good estimation of the noise transition matrix.

Empirical results illustrate the effectiveness of the proposed estimator for estimating the transition matrix in noisy multi-label learning, and the consistent algorithms with our estimator can achieve better classification performance.

The rest of the paper is organized as follows. In Section 2, we briefly present the problem setting of label-noise learning in multi-label classification. In Section 3, we discuss the identifiability of transition matrices under such noisy multi-label setting, and introduce our estimation method. Experimental results are provided in Section 4. Finally, we conclude the paper in Section 5.

2 Problem Setting

In this section, we present the problem setting of label-noise learning in multi-label classification.

Notations. In what follows, scalars are in lowercase letters, vectors are in lowercase boldface letters, and matrices/variables are in uppercase letters.

Preliminaries. Let D be the distribution of a pair of random variables $(\boldsymbol{X}, \boldsymbol{Y})$, where $\boldsymbol{X} \in \mathcal{X} \subseteq \mathbb{R}^d$ denotes the variable of instances, $\boldsymbol{Y} = \{Y^1, Y^2, ..., Y^q\} \in \{0,1\}^q$ the variable of target with q possible class labels. $Y^j = 1$ indicates instance X is associated with the class k; $Y^j = 0$, otherwise. In multi-label learning, the goal is to learn a function from D which maps each unseen example $\boldsymbol{x} \in \mathcal{X}$ to proper labels $\boldsymbol{y}$. However, as discussed, $\boldsymbol{Y}$ is hard to be annotated precisely. Before being observed, their true labels are independently flipped and what we can obtain are noisy training samples $\{(\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i)\}_{i=1}^n$, where $\bar{\boldsymbol{y}}_i$ denotes the noisy label. Let $\bar{D}$ be the distribution of the noisy random variables $(\boldsymbol{X}, \bar{\boldsymbol{Y}}) \in \mathcal{X} \times \{0,1\}^q$. Let $[z] = \{1, \ldots, z\}$

Transition matrices. The random variables $\bar{Y}^j$ and Y^j for the class j are related through a noise transition matrix $\boldsymbol{T}^j \in [0,1]^{2\times 2}$, $j \in [q]$. Generally, the transition matrix depends on instances, i.e., $T^j_{ik}(\boldsymbol{x}) = P(\bar{Y}^j = k \mid Y^j = i, \boldsymbol{X} = \boldsymbol{x})$. Given only noisy examples, the instance-dependent transition matrix is hard to learn without any additional assumption [38, 43]. In this paper, we study a special case by assuming that the transition matrix is class-dependent and instance-independent, i.e., $P(\bar{Y}^j = k \mid Y^j = i, \boldsymbol{X} = \boldsymbol{x}) = P(\bar{Y}^j = k \mid Y^j = i) = T^j_{ik}$, which is focused by the vast majority of current state-of-the-art label-noise learning algorithms in the single-label case [26, 46, 42, 23, 31].

Consistent algorithms. The transition matrix bridges the class posterior probabilities for noisy and clean data, i.e., $P(\bar{Y} = k \mid \boldsymbol{X} = \boldsymbol{x}) = \sum_{i=0}^{1} T_{ik} P(Y = i \mid \boldsymbol{X} = \boldsymbol{x})$. Thus, it has been exploited to achieve many statistically consistent algorithms for noisy multi-class learning. Specifically, it has been utilized to build risk-consistent estimators via correcting loss functions [20, 26, 38], and to design classifier-consistent estimators via limiting hypotheses, e.g., [26, 6, 48]. Since the multi-label task can be decomposed into multiple conditionally independent binary classification problems, we also can apply these consistent methods in noisy multi-label learning [40]. In this paper, without loss of generality, we adopt a risk-consistent algorithm, i.e. Reweight [20, 38], to learn statistically consistent classifiers with estimated transition matrices. More details can be found in Appendix A.

Transition matrix estimation. As inaccurate transition matrices will degenerate the performances of these consistent algorithms, a series of estimation methods [20, 37, 44, 49, 18] have been proposed for noisy multi-class learning to efficiently identify the transition matrix. However, most of them require anchor points [26, 44, 18], which is a strong assumption and hard to check in multi-label cases when only noisy data are provided [38]. Besides, severe class imbalance in multi-label learning [28] will make it difficult to accurately approximate the noisy or intermediate class posterior of anchor points, which is crucial for these methods. This motivates us to seek for a better estimator that can do without anchor points and avoid estimating noisy class posterior in noisy multi-label learning.

3 Estimating Transition Matrices using Label Correlations

In this section, we first study the identifiability problem [22] of class-dependent transition matrices under multi-label cases. Furthermore, inspired by these results, we propose a new estimator to approximate the transition matrix by utilizing the wealth of label correlations under the noisy multi-label setting. It is worth pointing out that our estimator demands neither the existence of the anchor points nor accurate fitting of noisy class posterior.

3.1 Identifiability of transition matrix

Recently, Liu et al. [22] built identifiability of the noise transition matrix on the Kruskal's identifiability results. Inspired by them, with the complex correlations among class labels, we can get some identifiability results of the class-dependent and instance-independent transition matrix in the noisy multi-label setting. First of all, we define the identifiability of the noise transition matrix $\boldsymbol{T}$ and present some reasonable assumptions.

Definition 1 (Identifiability of $\boldsymbol{T}$)**.** *For a class-dependent transition matrix $\boldsymbol{T}$, denote the distribution induced by $\boldsymbol{T}$ for on the observation space Ω as P_T. Identifiability requires that $P_{\boldsymbol{T}} \neq P_{\boldsymbol{T}'}$ for $\boldsymbol{T} \neq \boldsymbol{T}'$, up to label permutation.*

Assumption 1. $P(\bar{Y}^j = 0 \mid Y^j = 1) + P(\bar{Y}^j = 1 \mid Y^j = 0) < 1$, $j \in [q]$

Assumption 1 means that the noisy label is agreed with the clean label on average, which is a standard condition for analysis under the class-dependent transition matrix [25, 24].

Assumption 2. $P(Y^i = 0 \mid Y^j = 0) \neq P(Y^i = 0 \mid Y^j = 1)$, $i, j \in [q]$ *and* $i \neq j$.

Assumption 2 means that the multiple labels have correlations between each other, which is satisfied by the most of (i, j) pairs in the real-world dataset (See Appendix B).

When considering multi-label learning, the simplest case is having two class labels. In this case, the following result can be obtained:

Theorem 1. *Noisy labels $\{\bar{Y}^j, \bar{Y}^i\}$ will not suffice to identify $\boldsymbol{T}^j$.*

This result tells us that we should utilize more information about Y^j to achieve identifiability except for the correlations of two noisy labels. A natural idea is increasing the number of noisy labels, and actually, we can prove Theorem 2 based on the Kruskal's identifiability result [16, 32]

Theorem 2. *If $\bar{Y}^i$ and $\bar{Y}^k$ are independent given Y^j, noisy labels $\{\bar{Y}^j, \bar{Y}^i, \bar{Y}^k\}$ are sufficient to identify $\boldsymbol{T}^j$.*

The assumption that $\bar{Y}^i$ and $\bar{Y}^k$ are independent given Y^j can be satisfied in certain cases, e.g. the occurrences of "blue" and "dolphin" may be independent given "sea" appearing or not. Nevertheless, due to the complex correlations among labels, this assumption is hard to hold in most cases. When

the assumption can not hold, these label correlations are no more sufficient to determine $\boldsymbol{T}^j$, as shown in Theorem 3 in the following.

Theorem 3. *If $\bar{Y}^i$ and $\bar{Y}^k$ are not independent given Y^j, noisy labels $\{\bar{Y}^j, \bar{Y}^i, \bar{Y}^k\}$ will not suffice to identify $\boldsymbol{T}^j$.*

The inspiration from Theorem 3 is that, although increasing the number of noisy labels can provide more information, it may also increase model complexity due to the entangled correlations, making the identifiability decrease. Hence, we should know that how to improve the identifiability without increasing model complexity.

A direct way is to reduce the number of unknown model parameters with side information. Inspired by this, we prove Theorem 4 by providing the transition relationship between noisy label for class i and clean label for class j via side information.

Theorem 4. *Noisy labels $\{\bar{Y}^j, \bar{Y}^i\}$ and $P(\bar{Y}^i \mid Y^j)$ are sufficient to identify $\boldsymbol{T}^j$.*

In a high level, Theorem 4 theoretically guarantees that the identifiability of class-dependent transition matrix can be achieved by utilizing *the mismatch of label correlations* implied in the occurrence probabilities $P(\bar{Y}^i, \bar{Y}^j)$ and $P(\bar{Y}^i \mid Y^j)$.

The detailed proof of Theorem 1-4 is provided in Appendix C-F.

3.2 Our estimator

Our estimator is based on Theorem 4, where the side information is utilized to estimate $P(\bar{Y}^i \mid Y^j)$. Recently, the *memorization effect* [2] of deep networks have received much attention in learning with noisy labels, which shows that deep networks will first memorize the training data with clean labels and then those with noisy labels. Prior works utilize this characteristic to develop the *sample selection* methods [12, 9, 17, 13] for learning with noisy labels, where we select some samples $\mathcal{D}_s^j$ with more likely clean labels for each class j respectively in the early learning phase. The samples we selected can serve as the side information about clean label correlations, through which we can achieve estimation via counting. However, when implementing sample-selection-based methods, a major concern is whether the sampling bias will lead to large estimation errors.

Generally speaking, according to the *memorization effect*, the sampling bias is selecting easy samples for class j, which usually means these samples has the easy-to-discriminate features for class j, and we can reasonably assume that given Y^j, the features about class j is biased, while the features about another class i is unbiased, i.e.

$$P_{D_s^j}(\bar{Y}^i|Y^j) = \int P_{D_s^j}(\bar{Y}^i|\boldsymbol{X}^i)P_{D_s^j}(\boldsymbol{X}^i|Y^j)d\boldsymbol{x} = \int P_{\bar{D}_s^j}(\bar{Y}^i|\boldsymbol{X}^i)P_{\bar{D}_s^j}(\boldsymbol{X}^i|Y^j)d\boldsymbol{x} = P_{\bar{D}_s^j}(\bar{Y}^i|Y^j) \tag{1}$$

where D_s^j is the distribution of $(\boldsymbol{X}, \boldsymbol{Y_s})$, $\boldsymbol{Y_s} = \{\bar{Y}^1, \bar{Y}^2, ..., Y^j, ..., \bar{Y}^q\}$, $\bar{D}_s^j$ is the biased distribution of $(\boldsymbol{X}, \boldsymbol{Y_s})$, and $\boldsymbol{X}^i$ is the part of $\boldsymbol{X}$, which represents all information about class i appearing or not. When the assumption is satisfied, the sample selection will not lead to large estimation error on $\hat{P}(\bar{Y}^i \mid Y^j)$, and it can converge to zero exponentially fast by counting [4]. In real-world scenarios, due to the complex label correlations, this assumption will not strictly hold, but it may be roughly met as the classifying of simplest samples is not easily affected by the presence or absence of other labels. In Section 4.1, our empirical results justify this by showing a little gap between the estimation error of our estimator with the biased sample selection and an unbiased one.

Based on the above discussions, we can approximate $P(\bar{Y}^i, \bar{Y}^j)$ and $P(\bar{Y}^i \mid Y^j)$ with a little error by frequency counting, and utilize the mismatch of label correlations implied in these occurrence probabilities to estimate the transition matrix. Hence, we propose to estimate transition matrices $\{\hat{\boldsymbol{T}}^j\}_{j=1}^q$ by following two stages:

In the first stage, we utilize sample selection to obtain side information about clean label correlations. We train a classifier with the standard multi-label classification loss on noisy training samples $\mathcal{D}_t$ for a few epochs, and then perform sample selection to get selected "clean" set $\mathcal{D}_s^j$ for each class label j. Specially, we use a commonly used sample selection way [1, 17] in learning with noisy labels, which extracts the subset of instances with small losses by modeling the distribution of per-sample loss for class j with a Gaussian Mixture Model (GMM).

In the second stage, we perform co-occurrence estimation by frequency counting, and then estimate transition matrix by solving a simple bilinear decomposition problem. For class label j, we first choose another class label i and estimate $P(\bar{Y}^i, \bar{Y}^j)$ and $P(\bar{Y}^i \mid Y^j)$ via counting, i.e.,

$$\hat{P}(\bar{Y}^i = v, \bar{Y}^j = k) = \frac{1}{n} \sum_{(\boldsymbol{x}, \bar{\boldsymbol{y}}) \in \mathcal{D}_t} \mathbb{I}[\bar{y}^j = v, \bar{y}^i = k] \quad \text{and} \tag{2}$$

$$\hat{P}(\bar{Y}^i = v \mid Y^j = k) = \frac{\sum_{(\boldsymbol{x}, \bar{\boldsymbol{y}}) \in \mathcal{D}_s^j} \mathbb{I}[\bar{y}^i = v, \bar{y}^j = k]}{\sum_{(\boldsymbol{x}, \bar{\boldsymbol{y}}) \in \mathcal{D}_s^j} \mathbb{I}[\bar{y}^j = k]}, \tag{3}$$

where $\mathbb{I}[\cdot]$ is the indicator function which takes 1 if the identity index is true and 0 otherwise.

Then, these co-occurrence probabilities, which imply the mismatch of label correlations, can lead to four equations involving $\boldsymbol{T}^j$:

$$\begin{aligned}
P\left(\bar{Y}^j = 0, \bar{Y}^i = 0\right) &= P(Y^j = 0) T_{00}^j P(\bar{Y}^i = 0 | Y^j = 0) + P(Y^j = 1) T_{10}^j P(\bar{Y}^i = 0 | Y^j = 1), \\
P\left(\bar{Y}^j = 0, \bar{Y}^i = 1\right) &= P(Y^j = 0) T_{00}^j P(\bar{Y}^i = 1 | Y^j = 0) + P(Y^j = 1) T_{10}^j P(\bar{Y}^i = 1 | Y^j = 1), \\
P\left(\bar{Y}^j = 1, \bar{Y}^i = 0\right) &= P(Y^j = 0) T_{01}^j P(\bar{Y}^i = 0 | Y^j = 0) + P(Y^j = 1) T_{11}^j P(\bar{Y}^i = 0 | Y^j = 1), \\
P\left(\bar{Y}^j = 1, \bar{Y}^i = 1\right) &= P(Y^j = 0) T_{01}^j P(\bar{Y}^i = 1 | Y^j = 0) + P(Y^j = 1) T_{11}^j P(\bar{Y}^i = 1 | Y^j = 1).
\end{aligned}$$

For simplicity, we denote

$$\begin{aligned}
\boldsymbol{E} &= \begin{pmatrix} P(\bar{Y}^j = 0, \bar{Y}^i = 0) & P(\bar{Y}^j = 0, \bar{Y}^i = 1) \\ P(\bar{Y}^j = 1, \bar{Y}^i = 0) & P(\bar{Y}^j = 1, \bar{Y}^i = 1) \end{pmatrix} = \begin{pmatrix} e_{00} & e_{01} \\ e_{10} & e_{11} \end{pmatrix}, \\
\boldsymbol{P} &= \begin{pmatrix} P(Y^j = 0) & 0 \\ 0 & P(Y^j = 1) \end{pmatrix} = \begin{pmatrix} 1-p & 0 \\ 0 & p \end{pmatrix}, \\
\boldsymbol{T}^j &= \begin{pmatrix} P(\bar{Y}^j = 0 \mid Y^j = 0) & P(\bar{Y}^j = 1 \mid Y^j = 0) \\ P(\bar{Y}^j = 0 \mid Y^j = 1) & P(\bar{Y}^j = 1 \mid Y^j = 1) \end{pmatrix} = \begin{pmatrix} 1-\rho_- & \rho_- \\ \rho_+ & 1-\rho_+ \end{pmatrix}, \quad \text{and} \\
\boldsymbol{M} &= \begin{pmatrix} P(\bar{Y}^i = 0 \mid Y^j = 0) & P(\bar{Y}^i = 1 \mid Y^j = 0) \\ P(\bar{Y}^i = 0 \mid Y^j = 1) & P(\bar{Y}^i = 1 \mid Y^j = 1) \end{pmatrix} = \begin{pmatrix} 1-\rho'_- & \rho'_- \\ \rho'_+ & 1-\rho'_+ \end{pmatrix}.
\end{aligned}$$

Then the system of equations can be expressed as $\boldsymbol{E} = (\boldsymbol{T}^j)^\top \boldsymbol{P} \boldsymbol{M}$, i.e.

$$\begin{pmatrix} e_{00} & e_{01} \\ e_{10} & e_{11} \end{pmatrix} = \begin{pmatrix} 1-\rho_- & \rho_- \\ \rho_+ & 1-\rho_+ \end{pmatrix}^\top \begin{pmatrix} 1-p & 0 \\ 0 & p \end{pmatrix} \begin{pmatrix} 1-\rho'_- & \rho'_- \\ \rho'_+ & 1-\rho'_+ \end{pmatrix}.$$

Denote the estimation of $\boldsymbol{E}$,$\boldsymbol{P}$, $\boldsymbol{T}^j$ and $\boldsymbol{M}$ as

$$\hat{\boldsymbol{E}} = \begin{pmatrix} \hat{e}_{00} & \hat{e}_{01} \\ \hat{e}_{10} & \hat{e}_{11} \end{pmatrix}, \hat{\boldsymbol{P}} = \begin{pmatrix} 1-\hat{p} & 0 \\ 0 & \hat{p} \end{pmatrix}, \hat{\boldsymbol{T}}^j = \begin{pmatrix} 1-\hat{\rho}_- & \hat{\rho}_- \\ \hat{\rho}_+ & 1-\hat{\rho}_+ \end{pmatrix}, \hat{\boldsymbol{M}} = \begin{pmatrix} 1-\hat{\rho}'_- & \hat{\rho}'_- \\ \hat{\rho}'_+ & 1-\hat{\rho}'_+ \end{pmatrix}$$

As $\hat{\boldsymbol{E}}$ and $\hat{\boldsymbol{M}}$ can be derived from Eq. (2) and Eq. (3), the problem is hence equivalent to a bilinear decomposition problem:

$$\hat{\boldsymbol{E}}(\hat{\boldsymbol{M}})^{-1} = (\hat{\boldsymbol{T}}^j)^\top \hat{\boldsymbol{P}}. \tag{4}$$

By solving the above matrix equation, we can get

$$p = \frac{(1-\hat{\rho}'_-) - (\hat{e}_{00} + \hat{e}_{10})}{1 - \hat{\rho}'_- - \hat{\rho}'_+}, \tag{5}$$

and the estimation of the transition matrix

$$\hat{\boldsymbol{T}}^j = [\hat{\boldsymbol{E}}(\hat{\boldsymbol{M}})^{-1}(\hat{\boldsymbol{P}})^{-1}]^\top. \tag{6}$$

Implementation of our estimator. The pseudo code of our estimator is described in Algorithm 1. A little difference from the above is that in order to make better use of correlations among labels, we perform R times co-occurrence estimation and bilinear decomposition for different classes i in the second stage to get R estimations, $\hat{\boldsymbol{T}}_r^j, r = 1, 2, ..., R$. Finally, we estimating the transition matrix $\boldsymbol{T}^j$ by Eq. (7).

$$\hat{\boldsymbol{T}}^j = \arg\min_{\hat{\boldsymbol{T}}_r^j} \sum_{i=1}^{R} \|\hat{\boldsymbol{T}}_r^j - \hat{\boldsymbol{T}}_i^j\|_1, \tag{7}$$

where $\|\cdot\|_1$ denotes ℓ_1 norm.

Algorithm 1 Estimating Label-Noise Transition Matrices using Label Correlations

Require: Noisy training samples $\mathcal{D}_t$, the number of classes q, the early warmup training epoch E_{warm}, clean probability threshold of sample selection τ, and repeated estimation times R.

Stage1: Standard Training and Sample Selection

Standard training with the standard multi-label loss for E_{warm} epochs.
for $j = 1, 2, ..., q$ **do**
  Model per-sample loss with trained classifier to obtain clean proabability for class label j on $\mathcal{D}_t$ by a GMM.
  Get the selected sample set $\mathcal{D}_s^j$ for class label j with clean probability threshold τ.
end for

Stage2: Co-occurrence Estimation and Bilinear Decomposition

for $j = 1, 2, ..., q$ **do**
  for $r = 1, 2, ..., R$ **do**
    Choose another class label i.
    Estimating $P(\bar{Y}^i, \bar{Y}^j)$ by $\hat{P}(\bar{Y}^i, \bar{Y}^j)$ with Eq. (2) on $\mathcal{D}_t$, and $P(\bar{Y}^i \mid Y^j)$ by $\hat{P}(\bar{Y}^i \mid Y^j)$ with Eq. (3) on $\mathcal{D}_s^j$.
    Solve a bilinear decomposition problem (Eq. (4)) to get a estimation $\hat{\boldsymbol{T}}_r^j$ by Eq. (6).
  end for
  Estimating transition matrix $\boldsymbol{T}^j$ by $\hat{\boldsymbol{T}}^j$ which has the minimum error with Eq. (7) from R estimations.
end for

Ensure: The estimated transition matrices $\{\hat{\boldsymbol{T}}^j\}_{j=1}^q$.

4 Experiments

Dataset We verify the effectiveness of the proposed method on three synthetic noisy multi-label datasets, i.e., Pascal-VOC2007 [7], Pascal-VOC2012 [8], and MS-COCO [19]. Pascal-VOC2007 [7] and Pascal-VOC2012 [8] datasets are two popular datasets for object recognition. They both contain images from same 20 object classes, with an average of $n_a = 1.5$ labels per image. Pascal-VOC2007 contains a training set of 5,011 images and a test set of 4,952 images. Pascal-VOC2012 consists of 11,540 images as training set and 10,991 images as the test set [3]. As the labels of the test set in Pascal-VOC2012 are not publicly available, we use the test set in Pascal-VOC2007 for Pascal-VOC2012 evaluation. MS-COCO [19] is a widely used multi-label dataset. It contains 82,081 images as the training set and 40,137 images as the test set and covers 80 object classes with an average of $n_a = 2.9$ labels per image. For these datasets, we corrupted the training sets manually according to true transition matrices $\{\boldsymbol{T}^j\}_{j=1}^q$. For convenience, we use the same true transition matrices for all classes, i.e. $\boldsymbol{T}^j = \boldsymbol{T} = \begin{pmatrix} 1-\rho_- & \rho_- \\ \rho_+ & 1-\rho_+ \end{pmatrix}$,but do not divulge this information for algorithms. we generate four different types of synthetic datasets by using different transition matrices: 1) $\rho_- = 0, \rho_+ = \rho$, which annotates some positive examples as negative examples, also known as multi-label learning with missing labels [35, 34]; 2) $\rho_- = \rho, \rho_+ = 0$, which annotates some negative examples as positive examples, also known as partial multi-label learning [39, 41]; 3) $\rho_- = \rho, \rho_+ = \rho$, where positive samples and negative samples are mislabeled with the same probability ρ; 4) $\rho_- = \frac{n_a}{q-n_a}\rho, \rho_+ = \rho$, where positive samples and negative samples are mislabeled with the same number. In the experiments, we test the algorithms on various ρ. For all datasets, we leave out 10% of the noisy training examples as a noisy validation set. We use mAP on noisy validation set as the criterion for model selection.

Implementation details For a fair comparison, we implement all methods with default parameters by PyTorch on NVIDIA RTX 3090. We use a ResNet-50 network [10] pre-trained on ImageNet [29] for all datasets, and the optimizer is Adam optimizer [14] with $\beta = (0.9, 0.999)$. The batch size is 128, the learning rate is 5e-5. The number of training epochs is 20 for Pascal-VOC2007/VOC2012, and 30 for MS-COCO. For the transition matrix estimation method, E_{warm} is the same as the normal training epoch. For our estimator, we perform sample selection based on the average losses of 5 epochs before a certain warmup epoch (10th epoch for Pascal-VOC2007/VOC2012, 15th epoch for MS-COCO), $R = q - 1$ and $\tau = 0.5$ in all experiments. All experiments are run at least three times

with different random seeds, and we report the average and standard deviation value of results. The best results are in **bold**, and the second-best results are in blue.

4.1 Comparison for estimating transition matrices

Baselines For evaluating the effectiveness of estimating the transition matrix under multi-label cases, we compare the proposed method with the following methods: (1) T-estimator max [20, 26], which estimates the transition matrix via the noisy class posterior probabilities for anchor points that have the largest estimated noisy class posteriors. (2) T-estimator 97% [20, 26], which selects the points with 97% largest estimated noisy class posteriors to be anchor points. (3) Dual T-estimator max [44], which introduces an intermediate class to avoid directly estimating the noisy class posterior, and selects the points with the largest estimated intermediate class posteriors to be the anchor points. (4) Dual T-estimator 97% [44], which selects the points with the 97% largest estimated intermediate class posteriors to be the anchor points.

Metrics We use the sum of estimation error for the transition matrices as the estimation evaluation metric, i.e. $\sum_{j=1}^{q} \|\boldsymbol{T}^j - \hat{\boldsymbol{T}}^j\|_1 / \|\boldsymbol{T}^j\|_1$.

Results In Tab. 1, 2 and 3, we can see that for all cases on three datasets, the proposed estimation method leads to the smallest or second smallest estimator errors across various noise rates. Note that since the fitting of noisy or intermediate class posterior is hard to be accurate in noisy multi-label learning, T-estimator and Dual T-estimator need to carefully tune a hyperparameter for better estimation under different noise rates, and it's very sensitive in some cases, e.g. MS-COCO datasets with noise rates $(0.1, 0.1)$. In contrast, our method uses the same hyperparameters on one dataset to get good results for all cases, which reflects its robustness to various noise rates. Besides, to study the ablation of sampling bias, we also run our method with an unbias sample selection, named "our estimator gold". We can see that sample bias is the main factor that contributes to the error for our estimator, but the little error gap between our estimator and our estimator gold shows it will not lead to large estimation error.

Table 1: Comparison for estimating transition matrices on Pascal-VOC2007 dataset.

Noise rates (ρ_-, ρ_+)	(0,0.2)	(0,0.6)	(0.2,0)	(0.6,0)	(0.1,0.1)	(0.2,0.2)	(0.017,0.2)	(0.034,0.4)
T-estimator max	3.89±0.03	10.52±0.58	3.01±0.12	4.47±0.22	3.18±0.22	5.28±0.20	3.99±0.10	6.28±0.44
T-estimator 97%	4.95±0.17	4.42±0.18	1.77±0.03	2.13±0.12	6.99±0.10	6.94±0.17	5.38±0.14	5.17±0.09
Dual T-estimator max	1.94±0.13	7.29±0.16	1.03±0.04	2.68±0.13	**2.13±0.23**	4.02±0.18	**1.71±0.08**	2.67 ±0.27
Dual T-estimator 97%	12.59±0.06	7.43±0.06	1.09±0.03	2.41±0.33	14.39±0.10	11.78±0.06	13.71±0.16	11.15±0.09
Our estimator	**1.51±0.12**	**2.30±0.13**	**0.37±0.08**	**1.34±0.33**	3.06±0.38	**3.21±0.32**	2.03±0.19	**1.84±0.32**
Our estimator gold	0.44±0.05	0.51±0.09	0.38±0.08	0.37±0.11	0.83±0.05	2.15±0.30	0.65±0.10	1.40±0.20

Table 2: Comparison for estimating transition matrices on Pascal-VOC2012 dataset.

Noise rates (ρ_-, ρ_+)	(0,0.2)	(0,0.6)	(0.2,0)	(0.6,0)	(0.1,0.1)	(0.2,0.2)	(0.017,0.2)	(0.034,0.4)
T-estimator max	3.90±0.01	10.28±0.33	2.87±0.09	4.55±0.08	3.29±0.07	5.25±0.15	4.05±0.04	6.82±0.20
T-estimator 97%	5.42±0.09	3.98±0.09	1.53±0.06	1.91±0.07	6.43±0.16	6.20±0.17	5.76±0.27	5.16±0.14
Dual T-estimator max	1.02±0.20	5.13±0.26	1.07±0.07	2.06±0.12	1.94±0.05	2.59±0.16	1.17±0.13	1.93±0.08
Dual T-estimator 97%	12.94±0.06	7.49±0.03	1.14±0.04	2.94±0.18	14.23±0.08	11.56±0.05	13.97±0.09	11.10±0.08
Our estimator	**0.83±0.10**	**1.94±0.15**	**0.26±0.03**	**0.91±0.12**	**1.74±0.22**	**1.79±0.17**	**0.94±0.07**	**1.07±0.14**
Our estimator gold	0.33±0.05	0.34±0.05	0.25±0.05	0.45±0.05	0.51±0.05	1.67±0.29	0.42±0.06	0.91±0.16

Table 3: Comparison for estimating transition matrices on MS-COCO dataset.

Noise rates (ρ_-, ρ_+)	(0,0.2)	(0,0.6)	(0.2,0)	(0.6,0)	(0.1,0.1)	(0.2,0.2)	(0.008,0.2)	(0.015,0.4)
T-estimator max	16.14±0.33	39.09±0.47	10.39±0.21	11.49±0.60	13.95±0.41	20.50±0.04	16.70±0.06	28.16±0.45
T-estimator 97%	50.49±0.01	25.70±0.08	4.04±0.08	3.70±0.02	51.17±0.16	39.45±0.11	49.96±0.18	37.54±0.10
Dual T-estimator max	**5.04±0.04**	**11.22±0.70**	4.65±0.07	9.55±0.84	13.02±0.45	15.79±0.38	**7.04±0.31**	**6.34±0.11**
Dual T-estimator 97%	61.49±0.02	30.97±0.03	1.53±0.00	7.86±0.12	64.20±0.02	48.67±0.01	63.12±0.02	46.91±0.01
Our estimator	7.42±0.38	11.23±0.11	**0.50±0.03**	**0.83±0.06**	**8.88±0.10**	**10.27±0.19**	7.51±0.43	8.77±0.20
Our estimator gold	0.82±0.03	0.80±0.04	0.40±0.04	0.66±0.04	1.94±0.04	8.14±0.06	0.95±0.05	2.02±0.08

4.2 Comparison for classification performance

Baselines We exploit 10 baselines: (1)Standard, which trains with a standard multi-label classification loss. (2)GCE [46], that proposes Generalized Cross-Entropy loss for robustness. (3)CDR [36], that performs different update rules for two types of parameters to achieve robust learning. (4)AGCN [45], that adopts a Dynamic GCN to model the relation of content-aware class representations. (5)CSRA [47], that generates class-specific features for every category by proposing a spatial attention score. (6)WSIC [11] that proposes to use a small set with clean labels to learn a residual net for regularization in noisy multi-label learning, and we only provide noisy datasets to it for a

Table 4: Comparison for classification performance on Pascal-VOC2007 dataset.

	Noise rates (ρ_-, ρ_+)	(0,0.2)	(0,0.6)	(0.2,0)	(0.6,0)	(0.1,0.1)	(0.2,0.2)	(0.017,0.2)	(0.034,0.4)
mAP	Standard	84.25±1.07	77.16±0.94	82.70±0.54	68.65±1.57	83.07±0.45	78.87±0.52	83.92±0.59	80.97±0.42
	GCE	83.85±1.09	73.32±2.22	83.03±0.51	67.47±1.74	83.68±0.66	79.39±0.95	84.40±0.34	80.68±0.52
	CDR	84.60±0.43	77.45±1.23	82.76±0.53	68.86±2.05	83.22±0.57	79.02±0.62	84.37±0.25	81.14±0.28
	AGCN	83.24±0.67	75.50±0.56	81.09±0.51	66.47±1.29	81.09±0.48	73.79±0.76	82.21±0.42	76.55±1.11
	CSRA	**85.11±0.51**	**79.47±1.22**	82.93±0.65	67.36±2.25	83.69±0.69	78.10±0.53	**84.94±0.36**	81.51±0.14
	WSIC	84.14±0.26	76.17±1.31	82.30±0.64	66.82±3.87	83.41±0.31	77.93±1.00	84.17±0.48	80.74±0.44
	Reweight-T max	84.20±0.46	76.97±1.20	83.04±0.39	71.36±2.47	83.48±0.15	79.10±0.52	84.06±0.24	81.01±0.99
	Reweight-T 97%	84.00±0.68	78.97±0.69	83.07±0.29	73.96±1.69	82.71±0.30	78.80±0.28	84.37±0.22	81.42±0.25
	Reweight-DualT max	84.46±0.20	77.65±1.06	83.75±0.44	73.75±1.61	83.94±0.31	79.48±1.24	84.60±0.30	**81.77±0.26**
	Reweight-DualT 97%	82.36±0.45	77.72±0.73	**84.56±0.40**	**75.76±2.11**	79.69±1.40	75.26±1.70	81.84±0.81	77.40±1.86
	Reweight-Ours	84.43±0.46	78.72±0.41	84.08±0.24	74.46±0.56	**84.03±0.29**	**80.44±0.52**	84.09±0.62	80.97±1.03
OF1	Standard	75.24±1.40	32.02±5.49	78.85±0.43	15.08±0.25	79.24±0.43	75.85±0.84	75.98±1.04	59.67±1.65
	GCE	76.17±1.57	36.13±4.07	79.28±0.44	14.85±0.22	79.73±0.70	76.27±0.55	76.80±0.68	60.26±2.43
	CDR	76.05±0.68	34.11±3.43	79.04±0.46	14.99±0.19	79.34±0.60	76.00±0.47	76.56±0.52	59.31±1.04
	AGCN	74.92±1.02	30.97±3.78	75.45±2.06	16.85±0.56	78.69±0.31	72.64±0.51	75.16±0.58	56.56±1.64
	CSRA	76.94±1.03	33.65±2.73	77.71±1.23	15.94±0.32	**80.36±0.53**	**76.92±0.34**	77.91±0.63	62.19±1.97
	WSIC	75.01±1.18	16.48±6.78	79.02±0.59	14.88±0.21	78.55±1.05	72.88±3.44	72.30±2.82	53.26±9.44
	Reweight-T max	76.97±0.45	41.54±2.64	79.65±0.44	47.68±5.65	80.00±0.27	73.58±1.67	76.94±0.37	66.77±0.93
	Reweight-T 97%	77.71±0.65	68.28±2.03	80.16±0.24	70.67±0.70	75.28±0.97	65.03±2.20	78.45±0.63	74.11±0.67
	Reweight-DualT max	78.38±0.41	**68.81±1.41**	80.02±1.12	65.41±1.84	79.87±0.27	65.36±7.31	78.55±0.36	47.04±4.15
	Reweight-DualT 97%	68.17±3.53	61.81±3.29	80.74±0.50	**72.53±2.65**	52.99±5.06	36.75±6.71	67.55±0.64	57.41±0.70
	Reweight-Ours	**78.62±0.58**	65.68±1.67	**80.85±0.25**	67.43±4.65	79.64±0.29	75.52±0.86	**79.25±0.52**	**74.35±1.65**
CF1	Standard	72.53±1.11	30.64±3.90	76.83±0.65	14.97±0.24	75.86±1.23	70.68±1.76	73.11±0.54	52.07±2.34
	GCE	73.10±1.27	33.07±4.65	77.25±0.66	14.77±0.20	76.73±1.57	71.24±1.42	73.37±0.98	56.89±2.84
	CDR	73.08±0.47	33.06±1.84	76.95±0.79	14.88±0.17	76.09±1.42	70.78±1.09	73.33±0.85	54.16±3.19
	AGCN	73.45±1.04	33.41±1.65	72.65±1.97	16.67±0.55	76.20±0.51	69.09±0.49	72.81±1.02	55.09±3.28
	CSRA	74.10±0.56	33.44±3.65	75.28±1.32	15.71±0.23	77.52±0.94	73.44±0.62	74.98±0.48	58.60±2.24
	WSIC	70.13±2.04	13.64±6.97	75.32±2.00	14.77±0.16	74.17±1.87	64.95±6.47	65.41±4.40	43.43±11.31
	Reweight-T max	74.05±0.51	39.37±1.36	77.28±0.47	50.35±5.52	77.20±0.39	73.32±0.47	74.08±0.29	62.99±1.80
	Reweight-T 97%	76.81±0.74	**71.24±1.33**	77.22±0.56	67.65±1.37	74.99±0.37	68.07±0.74	**77.62±0.42**	**74.56±0.26**
	Reweight-DualT max	75.27±0.56	45.31±1.86	76.56±1.80	63.99±0.42	77.27±0.40	69.48±4.48	75.66±0.35	68.32±1.22
	Reweight-DualT 97%	71.93±1.64	63.85±3.69	**77.95±0.79**	**68.44±2.93**	62.84±2.51	50.49±2.43	70.99±0.98	60.58±1.67
	Reweight-Ours	**76.86±0.48**	61.29±1.94	77.89±0.42	66.79±2.50	**78.04±0.40**	**74.08±0.79**	77.28±0.48	72.18±0.74

Table 5: Comparison for classification performance on Pascal-VOC2012 dataset.

	Noise rates (ρ_-, ρ_+)	(0,0.2)	(0,0.6)	(0.2,0)	(0.6,0)	(0.1,0.1)	(0.2,0.2)	(0.017,0.2)	(0.034,0.4)
mAP	Standard	85.97±0.09	80.02±0.62	85.70±0.19	76.13±0.86	85.91±0.10	82.54±0.51	86.03±0.24	82.91±0.74
	GCE	86.02±0.21	78.71±0.72	85.96±0.19	75.02±0.50	**86.29±0.15**	83.18±0.33	85.84±0.37	82.96±0.83
	CDR	86.09±0.14	80.52±1.61	85.61±0.18	76.53±1.26	86.01±0.19	82.79±0.42	85.92±0.38	83.48±0.62
	AGCN	85.16±0.12	79.67±0.43	84.91±0.38	77.54±0.29	84.69±0.30	80.15±0.49	84.75±0.12	81.85±0.35
	CSRA	**86.88±0.23**	**81.75±0.97**	85.39±0.27	75.08±0.84	86.14±0.14	80.98±1.22	**86.51±0.15**	**83.86±0.49**
	WSIC	86.39±0.38	80.75±0.49	85.53±0.28	77.00±1.03	85.67±0.17	82.01±0.87	86.07±0.22	83.19±0.19
	Reweight-T max	85.40±0.43	79.06±1.69	85.80±0.32	78.59±0.69	85.98±0.32	82.88±0.41	85.51±0.51	82.38±1.67
	Reweight-T 97%	85.97±0.27	81.04±0.79	85.81±0.21	80.12±0.75	85.66±0.55	82.81±0.52	85.99±0.51	83.28±1.07
	Reweight-DualT max	85.93±0.41	78.69±2.62	**86.47±0.26**	80.00±0.66	86.23±0.34	**84.21±0.18**	86.15±0.44	83.42±1.26
	Reweight-DualT 97%	83.93±0.52	79.97±1.42	86.37±0.25	**81.84±0.87**	82.39±0.85	78.61±1.01	83.33±1.40	80.95±1.15
	Reweight-Ours	86.01±0.54	80.33±1.85	86.12±0.10	80.54±1.30	86.23±0.28	83.42±0.51	85.92±0.42	83.56±1.31
OF1	Standard	77.91±0.20	27.84±4.17	80.47±0.34	14.93±0.28	81.27±0.22	78.51±0.13	77.83±0.36	61.67±2.10
	GCE	78.25±0.28	24.67±3.13	80.89±0.19	14.73±0.28	81.47±0.34	**78.78±0.11**	78.32±0.37	63.26±1.40
	CDR	78.02±0.23	29.45±6.39	80.65±0.36	14.91±0.26	81.34±0.35	78.58±0.14	77.82±0.34	61.94±1.91
	AGCN	76.11±0.76	31.06±4.73	80.49±0.51	15.42±0.35	80.28±0.44	76.02±1.33	75.83±1.18	61.59±4.20
	CSRA	78.61±0.56	32.36±6.27	79.88±0.57	15.71±0.50	**81.82±0.35**	78.12±1.11	79.05±0.59	61.77±3.46
	WSIC	78.56±0.78	27.90±6.72	80.56±0.32	15.09±0.24	80.98±0.49	77.36±1.51	78.52±0.65	57.56±4.11
	Reweight-T max	77.76±0.72	43.96±6.52	81.24±0.31	48.88±2.45	81.53±0.49	78.63±0.17	78.66±0.22	68.04±2.70
	Reweight-T 97%	79.31±0.20	**71.79±2.35**	81.56±0.29	75.10±1.81	77.81±1.09	71.28±1.99	79.20±0.35	73.90±1.81
	Reweight-DualT max	80.44±0.69	63.68±5.75	82.01±0.28	74.17±2.34	81.04±0.36	78.71±0.83	80.47±0.35	75.56±1.72
	Reweight-DualT 97%	60.81±1.36	58.42±2.41	**82.20±0.20**	**75.96±1.47**	56.42±1.24	30.28±1.82	64.36±1.42	58.49±1.73
	Reweight-Ours	**80.54±0.61**	69.08±4.58	81.77±0.43	75.38±3.05	81.11±0.56	77.78±0.37	**80.75±0.43**	**77.31±1.49**
CF1	Standard	75.15±0.62	29.76±3.77	79.06±0.35	14.87±0.29	79.05±0.26	75.61±0.52	76.07±0.46	60.32±1.90
	GCE	75.25±0.84	26.25±4.40	79.68±0.32	14.69±0.27	79.13±0.32	75.84±0.51	76.05±0.84	61.99±0.71
	CDR	75.32±0.66	31.49±4.03	79.20±0.20	14.85±0.25	79.14±0.45	75.56±0.49	76.11±0.44	60.02±1.91
	AGCN	74.16±1.06	33.33±4.55	78.81±0.12	15.42±0.76	78.09±0.99	73.28±1.80	73.91±1.39	58.57±4.67
	CSRA	75.91±0.98	32.83±4.88	78.86±0.25	15.48±0.49	79.87±0.26	75.53±1.12	76.40±0.73	59.53±3.12
	WSIC	76.46±1.39	30.19±5.09	79.49±0.57	14.98±0.23	78.50±0.91	74.10±2.39	76.05±0.90	55.11±4.30
	Reweight-T max	75.54±0.59	39.66±5.13	79.79±0.27	49.60±3.31	79.60±0.48	**76.45±0.29**	76.82±0.42	65.53±2.66
	Reweight-T 97%	**78.94±0.10**	**73.66±1.10**	79.73±0.29	**73.38±1.41**	77.54±0.88	72.39±1.57	78.78±0.35	**75.54±1.13**
	Reweight-DualT max	78.37±0.54	58.96±3.99	80.06±0.22	72.61±1.76	79.29±0.35	76.37±1.44	78.69±0.25	65.68±1.37
	Reweight-DualT 97%	67.96±1.13	61.89±2.49	**80.11±0.33**	71.87±3.50	65.09±1.07	52.28±1.03	69.26±0.82	63.10±1.57
	Reweight-Ours	78.81±0.53	66.76±3.38	79.82±0.43	72.84±2.16	**79.90±0.39**	75.90±0.81	**79.37±0.35**	75.16±1.42

fair comparison. (7)Reweight-T max, which learns with Reweight algorithm using transition matrices estimated by T-estimator max [26]. (8)Reweight-T 97%, which learns with Reweight algorithm using transition matrices estimated by T-estimator 97% [26]. (9)Reweight-DualT max, which learns with Reweight algorithm using transition matrices estimated by Dual T-estimator max [44]. (10)Reweight-

Table 6: Comparison for classification performance on MS-COCO dataset.

	Noise rates (ρ_-, ρ_+)	(0,0.2)	(0,0.6)	(0.2,0)	(0.6,0)	(0.1,0.1)	(0.2,0.2)	(0.008,0.2)	(0.015,0.4)
mAP	Standard	69.92±0.06	63.81±0.16	66.77±0.52	55.45±0.48	67.77±0.28	62.50±0.23	69.76±0.09	66.82±0.05
	GCE	69.90±0.05	62.58±0.17	67.32±0.11	54.01±0.70	68.62±0.16	63.21±0.35	69.99±0.11	66.72±0.19
	CDR	70.06±0.05	63.85±0.28	67.32±0.08	55.20±1.62	68.01±0.08	62.65±0.21	69.87±0.09	66.85±0.19
	AGCN	**71.48±0.14**	**65.75±0.32**	69.44±0.10	55.71±0.61	**69.42±0.23**	63.96±0.11	**70.90±0.13**	67.86±0.26
	CSRA	71.18±0.10	65.28±0.11	67.93±0.18	51.49±0.73	68.83±0.12	61.80±0.98	70.76±0.16	**68.02±0.15**
	WSIC	68.92±0.09	63.09±0.28	66.22±0.06	53.61±0.36	67.41±0.15	62.33±0.18	68.95±0.15	66.29±0.21
	Reweight-T max	69.99±0.18	63.94±0.11	67.40±0.13	58.27±0.25	67.85±0.05	63.28±0.12	69.76±0.07	66.24±0.51
	Reweight-T 97%	67.98±0.57	62.52±0.46	68.00±0.17	59.44±0.81	65.69±0.48	60.03±0.11	68.13±0.04	64.40±0.18
	Reweight-DualT max	67.57±0.21	60.39±0.53	68.57±0.25	58.42±0.82	68.01±0.51	62.17±0.32	68.76±0.08	65.75±0.15
	Reweight-DualT 97%	64.97±0.20	58.85±0.43	**69.68±0.25**	49.17±3.02	56.36±0.62	49.41±0.38	63.27±0.36	58.21±0.86
	Reweight-Ours	70.57±0.11	63.28±0.92	69.38±0.36	**61.88±0.66**	68.70±0.15	**64.46±0.10**	70.06±0.06	67.03±0.08
OF1	Standard	66.48±0.50	19.18±0.97	69.58±0.38	7.05±0.01	68.64±0.18	64.84±0.54	66.07±0.15	51.70±0.42
	GCE	66.67±0.38	19.61±1.61	69.82±0.25	7.03±0.02	69.44±0.18	64.98±0.79	66.58±0.32	52.04±0.51
	CDR	66.46±0.54	19.60±2.86	69.72±0.31	7.06±0.04	68.75±0.09	64.68±0.54	66.03±0.39	52.49±1.18
	AGCN	67.14±0.52	16.02±0.76	**70.63±0.12**	7.04±0.02	69.66±0.25	**66.07±0.55**	66.61±0.48	52.38±1.05
	CSRA	67.98±0.40	24.08±2.60	70.14±0.17	7.06±0.02	**69.70±0.31**	64.89±1.22	67.37±0.28	51.81±0.53
	WSIC	66.67±0.15	23.02±4.70	69.02±0.06	7.02±0.00	67.78±0.58	62.31±0.69	66.38±0.25	52.07±1.94
	Reweight-T max	67.13±0.41	39.47±1.47	69.84±0.11	59.26±1.76	64.03±2.00	57.68±4.00	66.45±0.30	53.63±1.28
	Reweight-T 97%	57.66±0.56	54.06±1.19	69.72±0.39	64.79±0.85	43.81±0.54	33.65±2.93	55.44±0.56	51.78±1.37
	Reweight-DualT max	65.22±0.28	55.01±0.86	70.16±0.26	56.90±4.55	59.62±1.59	49.79±0.18	65.64±0.71	61.73±2.11
	Reweight-DualT 97%	29.39±0.36	29.30±0.74	70.24±0.26	48.87±6.87	25.83±0.16	8.51±0.26	39.29±0.45	35.78±1.10
	Reweight-Ours	**70.10±0.10**	**61.74±0.64**	70.52±0.26	**65.78±0.56**	64.45±0.47	58.60±3.30	**69.40±0.31**	**65.93±0.42**
CF1	Standard	60.27±0.52	22.73±0.15	64.66±0.82	7.07±0.02	62.38±0.27	56.78±1.31	60.04±0.17	45.35±0.60
	GCE	60.76±0.08	21.06±1.12	65.27±0.22	7.04±0.03	63.60±0.25	56.72±1.56	60.66±0.19	44.28±1.12
	CDR	60.26±0.55	22.42±1.78	65.27±0.10	7.07±0.04	62.63±0.08	56.41±1.36	59.85±0.38	45.41±0.85
	AGCN	61.79±0.98	19.30±1.20	66.29±0.11	7.05±0.04	64.09±0.39	58.97±1.15	60.35±0.54	43.89±1.46
	CSRA	62.46±0.53	24.17±2.76	65.80±0.02	7.06±0.02	63.90±0.48	55.97±2.29	61.30±0.42	44.25±2.33
	WSIC	61.12±0.08	27.16±1.54	63.71±0.34	7.03±0.01	61.12±1.30	52.47±0.99	60.51±0.25	45.52±1.32
	Reweight-T max	61.59±0.51	32.78±0.70	64.82±0.24	56.68±0.80	63.35±0.16	59.93±0.50	60.92±0.08	48.40±0.33
	Reweight-T 97%	55.67±0.45	52.79±1.04	63.97±0.74	56.70±2.11	48.47±0.70	40.20±0.82	55.33±0.12	52.19±0.29
	Reweight-DualT max	64.79±0.23	52.16±2.08	63.51±0.35	54.62±1.12	66.29±0.34	**61.33±0.09**	65.18±0.22	60.88±0.59
	Reweight-DualT 97%	32.15±0.43	30.32±0.91	65.23±0.28	33.76±8.88	29.99±0.10	12.77±0.21	41.29±0.83	37.08±1.29
	Reweight-Ours	**67.18±0.17**	**57.46±0.52**	**65.42±0.49**	**58.63±1.30**	**66.65±0.15**	61.13±1.01	**66.42±0.10**	**62.94±0.28**

DualT 97%, which learns with Reweight algorithm using transition matrices estimated by Dual T-estimator 97% [44]. Note that Standard, AGCN and CSRA are designed for clean multi-label data, and GCE and CDR are designed for noisy multi-class learning.

Metrics Following conventional setting [7, 5, 28], we compute the mean average precision (mAP), overall F1-measure (OF1) and per-class F1-measure (CF1) as classification evaluation metrics. For each image, we assign a positive label if its prediction probability is greater than 0.5.

Results As shown in Tab. 4, 5 and 6, first, we can find those statistically consistent methods achieve the best or second-best results on all three metrics in the vast majority of cases, while other methods can only achieve good results in some cases. For example, on the Pascal-VOC2012 dataset with noise rates $(0.0, 0.6)$, CSRA achieves the best result in the mAP metric with the help of its well-designed network, but its performance is far lower than those consistent methods on the OF1 and CF1 metrics, which shows the learned model can not approximate well the true class posterior $P(\boldsymbol{Y}|\boldsymbol{X})$. Note that since network structure and loss correction are compatible, the risk-consistent methods can also help AGCN and CSRA perform better (shown in Appendix K). Second, theoretically, the more accurate the transition matrix is estimated, the more likely the consistent method is to achieve better results by increasing the size of noisy examples, and our experimental results verify this. Among those consistent methods, Rewight algorithm with our estimator (named "Reweight-Ours") obtains the most best or second-best results on the three metrics, which is due to the smaller error of our estimation. Especially on the challenging and large-scale MS-COCO dataset, Reweight-Ours outperforms almost all state-of-the-art methods on the CF1 metric and significantly surpasses other baselines by a large margin in some cases. For example, on the MS-COCO dataset with noise rates $(0.6, 0)$, Reweight-Ours achieves the best CF1 result (58.63±1.30), while the suboptimal result is 56.70±2.11, and the result of Standard is only 7.07±0.02. In addition, the ablation studies about loss correction ways and base learning algorithms are provided in Appendix J and K, which shows that our estimator can achieve much better performance with the advanced frameworks.

5 Conclusion

In this paper, we study the estimation problem of the transition matrices in the noisy multi-label setting. We prove some identifiability results of class-dependent transition matrices in such setting, inspired by which we propose a new estimator to approximate the transition matrix. The proposed estimator utilizes the information of label correlations, and demands neither anchor points nor accurate fitting of noisy class posterior. Experiments in three popular multi-label datasets illustrate the effectiveness

of the proposed estimator to accurately estimate transition matrices, and the consistent algorithms with this estimator achieve better classification performance.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *ICML*, pages 312–321, 2019.

[2] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *ICML*, pages 233–242, 2017.

[3] Hong-Yu Zhou Bin-Bin Gao. Learning to Discover Multi-Class Attentional Regions for Multi-Label Image Recognition. *TIP*, 30:5920–5932, 2021.

[4] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. Concentration inequalities - a nonasymptotic theory of independence. In *Concentration Inequalities*, 2013.

[5] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *CVPR*, pages 5177–5186, 2019.

[6] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. In *ICLR*, 2021.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[9] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pages 8536–8546, 2018.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[11] M. Hu, H. Han, S. Shan, and X. Chen. Weakly supervised image classification through noise regularization. In *CVPR*, pages 11517–11525, 2019.

[12] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pages 2309–2318, 2018.

[13] Taehyeon Kim, Jongwoo Ko, Sangwook Cho, Jinhwan Choi, and Se-Young Yun. FINE samples for learning with noisy labels. In *NeurIPS*, pages 24137–24149, 2021.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.

[15] Wilhelmus Petrus Krijnen. *The analysis of three-way arrays by constrained PARAFAC methods*. DSWO Press, Leiden University, 1993.

[16] Joseph B Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.

[17] Junnan Li, Richard Socher, and Steven C. H. Hoi. DivideMix: learning with noisy labels as semi-supervised learning. In *ICLR*, 2020.

[18] Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. Provably end-to-end label-noise learning without anchor points. In *ICML*, pages 6403–6413, 2021.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[20] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *TPAMI*, 38(3):447–461, 3 2016.

[21] Weiwei Liu, Xiaobo Shen, Haobo Wang, and Ivor W. Tsang. The emerging trends of multi-label learning. *TPAMI*, 2021.

[22] Yang Liu. Identifiability of label noise transition matrix. *CoRR*, abs/2202.02016, 2022.

[23] Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *ICML*, pages 6226–6236, 2020.

[24] Aditya Krishna Menon, Brendan van Rooyen, Cheng Soon Ong, and Bob Williamson. Learning from corrupted binary labels via class-probability estimation. In *ICML*, volume 37, pages 125–134, 2015.

[25] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *NeurIPS*, volume 26, 2013.

[26] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pages 2233–2241, 2017.

[27] Cosmin Octavian Pene, Amirmasoud Ghiassi, Taraneh Younesian, Robert Birke, and Lydia Yiyu Chen. Multi-label gold asymmetric loss correction with single-label regulators. *ArXiv*, abs/2108.02032, 2021.

[28] Tal Ridnik, Emanuel Ben Baruch, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. In *ICCV*, pages 82–91, 2021.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[30] Eric Arazo Sanchez, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *ICML*, 2019.

[31] Jun Shu, Qian Zhao, Zongben Xu, and Deyu Meng. Meta transition adaptation for robust deep learning with noisy labels. *arXiv preprint arXiv:2006.05697*, 2020.

[32] Nicholas D Sidiropoulos and Rasmus Bro. On the uniqueness of multilinear decomposition of n-way arrays. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 14(3):229–239, 2000.

[33] Jos MF Ten Berge and Nikolaos D Sidiropoulos. On uniqueness in candecomp/parafac. *Psychometrika*, 67(3):399–409, 2002.

[34] Baoyuan Wu, Fan Jia, W. Liu, Bernard Ghanem, and Siwei Lyu. Multi-label learning with missing labels using mixed dependency graphs. *IJCV*, 126:875–896, 2018.

[35] Baoyuan Wu, Zhilei Liu, Shangfei Wang, Bao-Gang Hu, and Qiang Ji. Multi-label learning with missing labels. In *ICPR*, pages 1964–1968, 2014.

[36] Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *ICLR*, 2021.

[37] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. In *NeurIPS*, 2020.

[38] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? In *NeurIPS*, pages 6835–6846, 2019.

[39] Ming-Kun Xie and Sheng-Jun Huang. Partial multi-label learning. In *AAAI*, 2018.

[40] Ming-Kun Xie and Sheng-Jun Huang. CCMN: A general framework for learning with class-conditional multi-label noise. *TPAMI*, 2022.

[41] Ming-Kun Xie and Sheng-Jun Huang. Partial multi-label learning with noisy label identification. *TPAMI*, 44:3676–3687, 2022.

[42] Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: An information-theoretic noise-robust loss function. In *NeurIPS*, 2019.

[43] Yu Yao, Tongliang Liu, Mingming Gong, Bo Han, Gang Niu, and Kun Zhang. Instance-dependent label-noise learning under a structural causal model. In *NeurIPS*, 2021.

[44] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual T: reducing estimation error for transition matrix in label-noise learning. In *NeurIPS*, 2020.

[45] Jin Ye, Junjun He, Xiaojiang Peng, Wenhao Wu, and Yu Qiao. Attention-driven dynamic graph convolutional network for multi-label image recognition. In *ECCV*, 2020.

[46] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, pages 8778–8788, 2018.

[47] Ke Zhu and Jianxin Wu. Residual attention: A simple but effective method for multi-label recognition. In *CVPR*, pages 184–193, 2021.

[48] Zhaowei Zhu, Tongliang Liu, and Yang Liu. A second-order approach to learning with instance-dependent label noise. In *CVPR*, pages 10113–10123, 2021.

[49] Zhaowei Zhu, Yiwen Song, and Yang Liu. Clusterability as an alternative to anchor points when learning with noisy labels. In Marina Meila and Tong Zhang, editors, *ICML*, pages 12912–12923, 2021.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section 2.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]

(b) Did you describe the limitations of your work? [Yes] See Appendix G.

(c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix H.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3.1.

(b) Did you include complete proofs of all theoretical results? [Yes] See Appendix C,D,E and F.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We provide the source codes in the supplemental material.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We report the standard deviation value of results.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Statistically Consistent Algorithms for Noisy Multi-label Learning

Many statistically consistent algorithms [20, 26, 38, 48] have been proposed for noisy multi-class learning. By decomposing the multi-label task into multiple conditionally independent binary classification problems, we can apply these consistent algorithms to each binary classification problem for noisy multi-label learning. Without loss of generality, we present applying a risk-consistent algorithm, i.e. Reweight [20, 38], in noisy multi-label learning here.

First of all, we decompose the task into q independent binary classification problems given $\boldsymbol{X}$, which is a widely used assumption for deep multi-label learning [5, 28, 40] and the surrogate loss is as:

$$\mathcal{L}(\boldsymbol{f}(\boldsymbol{X}), \boldsymbol{Y}) = \sum_{j=1}^{q} \ell(f_j(\boldsymbol{X}), Y^j) \tag{8}$$

where $\boldsymbol{f} = (f_1, f_2, ..., f_q)$ is the learnable q classification functions, and ℓ is the base loss function. In the deep learning commonunity, $\boldsymbol{f}$ is usually modeled by a deep nerual network with the outputs of q sigmoid functions, and ℓ is usually the binary cross entropy function.

Similar to the single-label case [20, 38], Reweight method employs the importance reweighting technique to rewrite the expected risk w.r.t. clean data:

$$\begin{aligned}
R(\boldsymbol{f}) &= \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})\sim D}[\mathcal{L}(\boldsymbol{f}(\boldsymbol{X}), \boldsymbol{Y})] = \sum_{j=1}^{q} \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y})}[\ell(f_j(\boldsymbol{X}), Y^j)] \\
&= \sum_{j=1}^{q} \int_{\boldsymbol{x}} \sum_{i} P_D(\boldsymbol{X} = \boldsymbol{x}, Y^j = i) \ell(f_j(\boldsymbol{x}), i) d\boldsymbol{x} \\
&= \sum_{j=1}^{q} \int_{\boldsymbol{x}} \sum_{i} P_{\bar{D}}(\boldsymbol{X} = \boldsymbol{x}, \bar{Y}^j = i) \frac{P_D(\boldsymbol{X} = \boldsymbol{x}, Y^j = i)}{P_{\bar{D}}(\boldsymbol{X} = \boldsymbol{x}, \bar{Y}^j = i)} \ell(f_j(\boldsymbol{x}), i) d\boldsymbol{x} \\
&= \sum_{j=1}^{q} \int_{\boldsymbol{x}} \sum_{i} P_{\bar{D}}(\boldsymbol{X} = \boldsymbol{x}, \bar{Y}^j = i) \frac{P_D(Y^j = i \mid \boldsymbol{X} = \boldsymbol{x})}{P_{\bar{D}}(\bar{Y}^j = i \mid \boldsymbol{X} = \boldsymbol{x})} \ell(f_j(\boldsymbol{x}), i) d\boldsymbol{x} \\
&= \sum_{j=1}^{q} \mathbb{E}_{(\boldsymbol{X},\bar{\boldsymbol{Y}})}[\bar{\ell}_j(f_j(\boldsymbol{X}), \bar{Y}^j)] = \mathbb{E}_{(\boldsymbol{X},\bar{\boldsymbol{Y}})\sim \bar{D}}[\bar{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{X}), \bar{\boldsymbol{Y}})],
\end{aligned} \tag{9}$$

where D denotes the distribution for clean data, $\bar{D}$ for noisy data, $\bar{\ell}_j(f_j(\boldsymbol{x}), i) = \frac{P_D(Y^j=i|\boldsymbol{X}=\boldsymbol{x})}{P_{\bar{D}}(\bar{Y}^j=i|\boldsymbol{X}=\boldsymbol{x})} \ell(f_j(\boldsymbol{x}), i)$, $\bar{\mathcal{L}}(\boldsymbol{f}(\boldsymbol{x}), \bar{\boldsymbol{y}}) = \sum_{j=1}^{q} \bar{\ell}_j(f_j(\boldsymbol{x}), \bar{Y}^j)$, and the third last equation holds because label noise is assumed to be independent of instances. In the paper, we have omitted the subscript for P when no confusion is caused.

We use the output of the sigmoid function $g_j(\boldsymbol{x})$ to approximate $P(Y^j = 1|X = \boldsymbol{x})$, i.e, $P(Y^j = 1|X = \boldsymbol{x}) \approx \hat{P}(Y^j = 1|X = \boldsymbol{x}) = g_j(\boldsymbol{x})$ and $P(Y^j = 0|X = \boldsymbol{x}) \approx \hat{P}(Y^j = 0|\boldsymbol{X} = \boldsymbol{x}) = 1 - g_j(\boldsymbol{x})$. Then, $\hat{P}(\bar{Y}^j = k \mid \boldsymbol{X} = \boldsymbol{x}) = \sum_{i=0}^{1} T_{ik}^j \hat{P}(Y^j = i \mid \boldsymbol{X} = \boldsymbol{x})$ is an approximation for $P(\bar{Y}^j = k \mid \boldsymbol{X} = \boldsymbol{x})$. By employing Reweight algorithm, we build the risk-consistent estimator as:

$$\bar{R}_{n,w}(\{T^j\}_{j=1}^{q}, \boldsymbol{f}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{q} \frac{\hat{P}(Y^j = \bar{Y}_i^j \mid \boldsymbol{X} = \boldsymbol{x}_i)}{\hat{P}(\bar{Y}^j = \bar{Y}_i^j \mid \boldsymbol{X} = \boldsymbol{x}_i)} \ell\left(f_j\left(\boldsymbol{x}_i\right), \bar{Y}_i^j\right), \tag{10}$$

where $f_j(\boldsymbol{x}) = \mathbb{I}[g_j(\boldsymbol{x}) > 0.5]$, and $\mathbb{I}[.]$ is the indicator function which takes 1 if the identity index is true and 0 otherwise; the subscript w denotes that the loss function is weighted.

B Validation of Assumption 2

In order to verify the assumption 2, we count the frequencies of $Y^i = 0$ given $Y^{50} = 0/1$ on MS-COCO training dataset, i.e. $\hat{P}(Y^i = 0|Y^{50} = 0)/\hat{P}(Y^i = 0|Y^{50} = 1)$. According to Hoeffding's inequality [4], when the frequencies whose difference between given $Y^{50} = 0$ and $Y^{50} = 1$ is greater than 0.015, we have at least 95% confidence to make sure $P(Y^i = 0|Y^{50} = 0) \neq P(Y^i = 0|Y^{50} =$

Table 7: The frequencies of $Y^i = 0$ given $Y^{50} = 0/1$ on MS-COCO training dataset. The frequencies whose difference between given $Y^{50} = 0$ and $Y^{50} = 1$ is greater than 0.015 are in **bold**.

$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$	$i=7$	$i=8$	$i=9$	$i=10$
0.960/0.983	0.979/0.992	**0.991/0.920**	0.974/0.985	**0.999/0.961**	**0.999/0.959**	**0.984/0.999**	**0.955/0.980**	**0.974/0.936**	**0.989/0.958**
$i=11$	$i=12$	$i=13$	$i=14$	$i=15$	$i=16$	$i=17$	$i=18$	$i=19$	$i=20$
0.955/0.987	0.980/0.970	**0.943/0.964**	0.925/0.929	**0.920/0.954**	**0.969/0.996**	**0.983/0.952**	0.975/0.975	**0.931/0.866**	**0.975/0.995**
$i=21$	$i=22$	$i=23$	$i=24$	$i=25$	$i=26$	$i=27$	$i=28$	$i=29$	$i=30$
0.937/0.989	**0.982/0.941**	**0.912/0.874**	**0.950/0.971**	0.958/0.964	**0.974/0.990**	0.918/0.923	**0.886/0.908**	0.957/0.968	0.987/0.987
$i=31$	$i=32$	$i=33$	$i=34$	$i=35$	$i=36$	$i=37$	$i=38$	$i=39$	$i=40$
0.976/0.986	0.981/0.989	0.966/0.972	**0.994/0.972**	**0.959/0.993**	0.998/0.999	**0.989/0.902**	0.982/0.969	0.990/0.990	**0.972/0.991**
$i=41$	$i=42$	$i=43$	$i=44$	$i=45$	$i=46$	$i=47$	$i=48$	$i=49$	$i=50$
0.997/0.967	0.962/0.962	0.966/0.973	0.980/0.992	**0.987/0.957**	**0.974/0.993**	**0.977/0.992**	**0.965/0.984**	0.993/0.995	**1.000/0.000**
$i=51$	$i=52$	$i=53$	$i=54$	$i=55$	$i=56$	$i=57$	$i=58$	$i=59$	$i=60$
0.970/0.975	**0.954/0.969**	0.972/0.986	0.981/0.967	0.974/0.985	0.989/0.994	**0.977/0.994**	**0.930/0.984**	**0.999/0.945**	**0.999/0.952**
$i=61$	$i=62$	$i=63$	$i=64$	$i=65$	$i=66$	$i=67$	$i=68$	$i=69$	$i=70$
0.999/0.975	0.965/0.973	**0.997/0.936**	0.980/0.990	0.984/0.977	**0.998/0.950**	0.976/0.987	**0.999/0.948**	**0.997/0.944**	0.997/0.999
$i=71$	$i=72$	$i=73$	$i=74$	$i=75$	$i=76$	$i=77$	$i=78$	$i=79$	$i=80$
0.945/0.994	0.991/0.992	0.969/0.961	0.964/0.975	**0.959/0.938**	**0.950/0.970**	**0.991/0.947**	**0.948/0.987**	0.984/0.974	**0.966/0.998**

$1)$. As shown in Tab. 7, class 50 have the correlation with another 46 classes with a high probability, which means they are very likely to hold Assumption 2, accounting for the majority of all 79 classes.

In the implementation of our estimator, we always assume class j and class i have a strong correlation at first, and use them to estimate. While, when a reasonable solution cannot be obtained, we will abandon class i and choose another class.

C Proof of Theorem 1

Lemma 1. *$\bar{Y}^i$ and $\bar{Y}^j$ are independent given Y^j.*

Proof. Since we assume that the transition matrix is class-dependent and instance-independent, $\bar{Y}^j$ and any variable are independent given Y^j. Therefore, this lemma holds. □

Lemma 2. *The product of two row-stochastic matrices is still a row-stochastic one.*

Proof. Let $\boldsymbol{P}$ and $\boldsymbol{Q}$ be row-stochastic matrices of the following form: $\boldsymbol{P} = \begin{pmatrix} 1-p_- & p_- \\ p_+ & 1-p_+ \end{pmatrix}$ and $\boldsymbol{Q} = \begin{pmatrix} 1-q_- & q_- \\ q_+ & 1-q_+ \end{pmatrix}$. Then the product of $\boldsymbol{P}$ and $\boldsymbol{Q}$ is $\boldsymbol{PQ} = \begin{pmatrix} 1-q_-+p_-(-1+q_-+q_+) & q_- - p_-(-1+q_-+q_+) \\ q_+ - p_+(-1+q_-+q_+) & 1-q_+ + p_+(-1+q_-+q_+) \end{pmatrix}$. It can be readily verified that the sum of each row of $\boldsymbol{PQ}$ is equal to 1, meaning that $\boldsymbol{PQ}$ is a row-stochastic matrix. □

Theorem 1. *Noisy labels $\{\bar{Y}^j, \bar{Y}^i\}$ will not suffice to identify $\boldsymbol{T}^j$.*

Proof. First, the information from $\bar{Y}^j, \bar{Y}^i$ can be fully captured by the following four quantities: $P\left(\bar{Y}^j = 0, \bar{Y}^i = 0\right)$, $P\left(\bar{Y}^j = 1, \bar{Y}^i = 0\right)$, $P\left(\bar{Y}^j = 0, \bar{Y}^i = 1\right)$, and $P\left(\bar{Y}^j = 1, \bar{Y}^i = 1\right)$. According to Lemma 1, these four quantities can lead to four equations that depend on $\boldsymbol{T}^j$:

$$
\begin{aligned}
P\left(\bar{Y}^j = 0, \bar{Y}^i = 0\right) &= P(Y^j = 0)T^j_{00}P(\bar{Y}^i = 0|Y^j = 0) + P(Y^j = 1)T^j_{10}P(\bar{Y}^i = 0|Y^j = 1) \\
P\left(\bar{Y}^j = 0, \bar{Y}^i = 1\right) &= P(Y^j = 0)T^j_{00}P(\bar{Y}^i = 1|Y^j = 0) + P(Y^j = 1)T^j_{10}P(\bar{Y}^i = 1|Y^j = 1) \\
P\left(\bar{Y}^j = 1, \bar{Y}^i = 0\right) &= P(Y^j = 0)T^j_{01}P(\bar{Y}^i = 0|Y^j = 0) + P(Y^j = 1)T^j_{11}P(\bar{Y}^i = 0|Y^j = 1) \\
P\left(\bar{Y}^j = 1, \bar{Y}^i = 1\right) &= P(Y^j = 0)T^j_{01}P(\bar{Y}^i = 1|Y^j = 0) + P(Y^j = 1)T^j_{11}P(\bar{Y}^i = 1|Y^j = 1).
\end{aligned}
$$

For simplicity, we denote

$$\boldsymbol{E} = \begin{pmatrix} P(\bar{Y}^j = 0, \bar{Y}^i = 0) & P(\bar{Y}^j = 0, \bar{Y}^i = 1) \\ P(\bar{Y}^j = 1, \bar{Y}^i = 0) & P(\bar{Y}^j = 1, \bar{Y}^i = 1) \end{pmatrix} = \begin{pmatrix} e_{00} & e_{01} \\ e_{10} & e_{11} \end{pmatrix},$$

$$\boldsymbol{P} = \begin{pmatrix} P(Y^j = 0) & 0 \\ 0 & P(Y^j = 1) \end{pmatrix} = \begin{pmatrix} 1-p & 0 \\ 0 & p \end{pmatrix},$$

$$\boldsymbol{T}^j = \begin{pmatrix} P(\bar{Y}^j = 0 \mid Y^j = 0) & P(\bar{Y}^j = 1 \mid Y^j = 0) \\ P(\bar{Y}^j = 0 \mid Y^j = 1) & P(\bar{Y}^j = 1 \mid Y^j = 1) \end{pmatrix} = \begin{pmatrix} 1-\rho_- & \rho_- \\ \rho_+ & 1-\rho_+ \end{pmatrix}, \text{and}$$

$$\boldsymbol{M} = \begin{pmatrix} P(\bar{Y}^i = 0 \mid Y^j = 0) & P(\bar{Y}^i = 1 \mid Y^j = 0) \\ P(\bar{Y}^i = 0 \mid Y^j = 1) & P(\bar{Y}^i = 1 \mid Y^j = 1) \end{pmatrix} = \begin{pmatrix} 1-\rho'_- & \rho'_- \\ \rho'_+ & 1-\rho'_+ \end{pmatrix}.$$

Then, the system of equations can be expressed as $\boldsymbol{E} = (\boldsymbol{T}^j)^\top \boldsymbol{P}\boldsymbol{M}$, i.e.

$$\begin{pmatrix} e_{00} & e_{01} \\ e_{10} & e_{11} \end{pmatrix} = \begin{pmatrix} 1-\rho_- & \rho_- \\ \rho_+ & 1-\rho_+ \end{pmatrix}^\top \begin{pmatrix} 1-p & 0 \\ 0 & p \end{pmatrix} \begin{pmatrix} 1-\rho'_- & \rho'_- \\ \rho'_+ & 1-\rho'_+ \end{pmatrix}. \tag{11}$$

Assuming $\{\boldsymbol{T}^0, \boldsymbol{P}^0, \boldsymbol{M}^0\}$ satisfies

$$\boldsymbol{E} = (\boldsymbol{T}^0)^\top \boldsymbol{P}^0 \boldsymbol{M}^0, \tag{12}$$

Next, we will prove that by selecting proper parameters, a different solution of Eq. (11) can be derived, which ruins the identifiability of $\boldsymbol{T}^j$. Let $\boldsymbol{A} = \begin{pmatrix} 1-a_- & a_- \\ a_+ & 1-a_+ \end{pmatrix}$ and $\boldsymbol{B} = \begin{pmatrix} 1-b_- & b_- \\ b_+ & 1-b_+ \end{pmatrix}$ be invertible, row-stochastic matrices. Based on the invertibility of $\boldsymbol{A}$ and $\boldsymbol{B}$, Eq. (12) can be rewritten as $\boldsymbol{E} = (\boldsymbol{A}\boldsymbol{T}^0)^\top (\boldsymbol{A}^\top)^{-1} \boldsymbol{P}^0 \boldsymbol{B}^{-1} \boldsymbol{B}\boldsymbol{M}^0$. According to Lemma 2, $\boldsymbol{T}^1 = \boldsymbol{A}\boldsymbol{T}^0$ and $\boldsymbol{M}^1 = \boldsymbol{B}\boldsymbol{M}^0$ are row-stochastic matrices, which is consistent with the form of $\boldsymbol{T}^j$ and $\boldsymbol{M}$.

Last, denoting $\boldsymbol{P}^0 = \begin{pmatrix} 1-p_0 & 0 \\ 0 & p_0 \end{pmatrix}$ and $\boldsymbol{P}^1 = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix}$, by letting $\boldsymbol{P}^1 = (\boldsymbol{A}^\top)^{-1} \boldsymbol{P}^0 \boldsymbol{B}^{-1} = \left[\begin{pmatrix} 1-a_- & a_- \\ a_+ & 1-a_+ \end{pmatrix}^\top \right]^{-1} \begin{pmatrix} 1-p_0 & 0 \\ 0 & p_0 \end{pmatrix} \begin{pmatrix} 1-b_- & b_- \\ b_+ & 1-b_+ \end{pmatrix} = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix}$ be in the form of $\boldsymbol{P}$, i.e., solving the following equations:

$$\begin{cases} p_{00} + p_{11} = 1, \\ p_{01} = 0, \\ p_{10} = 0, \end{cases}$$

we can get $a_+ = \frac{b_-(1-p_0)}{b_- - p_0}$ and $b_+ = \frac{a_-(1-p_0)}{a_- - p_0}$. It means that when we have a solution $\{\boldsymbol{T}^0, \boldsymbol{P}^0, \boldsymbol{M}^0\}$ of Eq. (11), we can get another different solution $\{\boldsymbol{T}^1, \boldsymbol{P}^1, \boldsymbol{M}^1\}$ by setting appropriate values of b_- and a_-. Hence, $\boldsymbol{T}^j$ is unidentifiable in this situation.

□

D Proof of Theorem 2

To make the proof clear, following [22], we reproduce the Kruskal's identifiability result here. The setup of Kruskal's identifiability result is as follows: suppose that there is an unobserved variable Z that takes values in $\{0, 1, ..., K-1\}$. Z has a non-degenerate prior $P(Z = i) > 0$. Instead of observing Z, we observe a set of conditionally independent variables $\left\{O^{(t)}\right\}_{t=1}^N$. Each $O^{(t)}$ has a finite state space with cardinality κ_t. Let $\boldsymbol{M}^{(t)}$ be a matrix of size $K \times \kappa_t$, which j-th row is simply $\left[P\left(O^{(t)} = 1 \mid Z = j\right), \ldots, P\left(O^{(t)} = \kappa_t \mid Z = j\right)\right]$. The previous works [16, 32] have proved the following Theorem 5.

Definition 2 (Kruskal rank [22])**.** *For a matrix $\boldsymbol{M}$, the Kruskal rank of $\boldsymbol{M}$ is the largest number I such that every set of I rows of $\boldsymbol{M}$ are independent. The symbol is* $\mathrm{Kr}(\boldsymbol{M}) = I$.

Theorem 5 (Kruskal's identifiability result [16, 32])**.** *The model parameters are uniquely identifiable, up to label swapping, if*

$$\sum_{t=1}^{N} \mathrm{Kr}\left(\boldsymbol{M}^{(t)}\right) \geq 2K + N - 1.$$

We can prove Theorem 2 with the following lemmas:

Lemma 3. *$\bar{Y}^i$ and Y^j are independent given Y^i.*

Proof. Since we assume that the transition matrix is class-dependent and instance-independent, $\bar{Y}^i$ and any variable are independent given Y^i. Therefore, this lemma holds. □

Lemma 4. *If $Y^j \in \{0, 1\}$ corresponds to Z, $\{\bar{Y}^j, \bar{Y}^i, \bar{Y}^k\}$ correspond to the observations $\left\{O^{(t)}\right\}_{t=1}^{3}$, then $\mathrm{Kr}\left(\boldsymbol{M}^{(t)}\right) = 2, t \in [3]$.*

Proof. As $\boldsymbol{M}^{(1)} = \boldsymbol{T}^j$ is a row-stochastic matrix, according to Assumption 1, every set of 2 rows of it are independent, thus its Kruskal rank is 2. Therefore, according to Definition 2, $\mathrm{Kr}\left(\boldsymbol{T}^j\right) = 2$.

The (p, q) entry of $\boldsymbol{M}^{(2)}$ is

$$\begin{aligned} M_{pq}^{(2)} &= P(\bar{Y}^i = q \mid Y^j = p) = \sum_{c=0}^{1} P(\bar{Y}^i = q, Y^i = c \mid Y^j = p) \\ &= \sum_{c=0}^{1} P(\bar{Y}^i = q \mid Y^i = c, Y^j = p) P(Y^i = c \mid Y^j = p) \\ &= \sum_{c=0}^{1} P(\bar{Y}^i = q \mid Y^i = c) P(Y^i = c \mid Y^j = p), \end{aligned} \tag{13}$$

where the last equation holds because of Lemma. 3.

Denote $\boldsymbol{M}'^{(2)} = \begin{pmatrix} P(Y^i = 0 \mid Y^j = 0) & P(Y^i = 1 \mid Y^j = 0) \\ P(Y^i = 0 \mid Y^j = 1) & P(Y^i = 1 \mid Y^j = 1) \end{pmatrix}$. Then Eq. (13) can be rewritten as the following matrix form:

$$\boldsymbol{M}^{(2)} = \boldsymbol{M}'^{(2)} \boldsymbol{T}^i, \tag{14}$$

where $\boldsymbol{T}^i$ denotes the transition matrix of class i. According to Assumption 1, $\mathrm{Kr}\left(\boldsymbol{T}^i\right) = 2$. As $\boldsymbol{M}'^{(2)}$ is a row-stochastic matrix, according to Assumption 2, every set of 2 rows of it are independent. Hence, according to Definition 2, $\mathrm{Kr}\left(\boldsymbol{M}'^{(2)}\right) = 2$.

Since $\boldsymbol{T}^i$ and $\boldsymbol{M}'^{(2)}$ are full-rank matrices, based on Eq. (13), we can get the Kruskal rank of $\boldsymbol{M}^{(2)}$ as $\mathrm{Kr}\left(\boldsymbol{M}^{(2)}\right) = 2$. Similarly, $\mathrm{Kr}\left(\boldsymbol{M}^{(3)}\right) = 2$. □

Theorem 2. *If $\bar{Y}^i$ and $\bar{Y}^k$ are independent given Y^j, noisy labels $\{\bar{Y}^j, \bar{Y}^i, \bar{Y}^k\}$ are sufficient to identify $\boldsymbol{T}^j$.*

Proof. According to Lemma 1 and that $\bar{Y}^i$ and $\bar{Y}^k$ are independent given Y^j, we can relate our multi-label noise setting to the setup of Kruscal's identifiability scenario: $Y^j \in \{0, 1\}$ corresponds to the unobserved hidden variable Z; $P(Y^j = i)$ corresponds to the prior of this hidden variable; Noisy labels $\{\bar{Y}^j, \bar{Y}^i, \bar{Y}^k\}$ correspond to the observations $\left\{O^{(t)}\right\}_{t=1}^{3}$. κ_t is then simply the cardinality of the noisy label space, i.e. $\kappa_t = K = 2$; Each $O^{(t)}$ has a corresponding observation matrix $\boldsymbol{M}^{(t)}$, and $\boldsymbol{M}_{vk}^{(t)} = P\left(O^{(t)} = k \mid Y^j = v\right)$. Now we can get the following result about identifiability of $\boldsymbol{T}$.

According to Lemma 4, the Kruskal ranks satisfy

$$\sum_{t=1}^{3} \mathrm{Kr}\left(\boldsymbol{M}^{(t)}\right) = 3K = 2K + 2 \geq 2K + N - 1, \text{ when } N = 3.$$

Calling Theorem 5 proves the uniqueness of $\boldsymbol{M}^{(t)}$. As $\boldsymbol{M}^{(1)} = \boldsymbol{T}^j$, then $\boldsymbol{T}^j$ is identifiable. □

E Proof of Theorem 3

Lemma 5. *$\bar{Y}^i, \bar{Y}^j$ and $\bar{Y}^k$ are independent given Y^j and Y^i.*

Proof. Since we assume that the transition matrix is class-dependent and instance-independent, $\bar{Y}^j$ and any variable are independent given Y^j, and $\bar{Y}^i$ and any variable are independent given Y^i. Therefore, this lemma holds. □

Lemma 6. *If $\boldsymbol{M}_{4\times 2} = P(\bar{Y}^i \mid Y^j, Y^i)$, the first two rows and last two rows of $\boldsymbol{M}$ are identical respectively, i.e. $M_{0p} = M_{1p}$ and $M_{2p} = M_{3p}, p = 0, 1$.*

Proof.

$$\boldsymbol{M} = \begin{pmatrix} P(\bar{Y}^i = 0 \mid Y^j = 0, Y^i = 0) & P(\bar{Y}^i = 1 \mid Y^j = 0, Y^i = 0) \\ P(\bar{Y}^i = 0 \mid Y^j = 1, Y^i = 0) & P(\bar{Y}^i = 1 \mid Y^j = 1, Y^i = 0) \\ P(\bar{Y}^i = 0 \mid Y^j = 0, Y^i = 1) & P(\bar{Y}^i = 1 \mid Y^j = 0, Y^i = 1) \\ P(\bar{Y}^i = 0 \mid Y^j = 1, Y^i = 1) & P(\bar{Y}^i = 1 \mid Y^j = 1, Y^i = 1) \end{pmatrix}$$
$$= \begin{pmatrix} P(\bar{Y}^i = 0 \mid Y^i = 0) & P(\bar{Y}^i = 1 \mid Y^i = 0) \\ P(\bar{Y}^i = 0 \mid Y^i = 0) & P(\bar{Y}^i = 1 \mid Y^i = 0) \\ P(\bar{Y}^i = 0 \mid Y^i = 1) & P(\bar{Y}^i = 1 \mid Y^j = 1) \\ P(\bar{Y}^i = 0 \mid Y^j = 1) & P(\bar{Y}^i = 1 \mid Y^j = 1) \end{pmatrix},$$

where the second equation holds because $\bar{Y}^i$ is only dependent on Y^i. Accordingly, $M_{0p} = M_{1p}$ and $M_{2p} = M_{3p}, p = 0, 1$. □

Theorem 3. *If $\bar{Y}^i$ and $\bar{Y}^k$ are not independent given Y^j, noisy labels $\{\bar{Y}^j, \bar{Y}^i, \bar{Y}^k\}$ will not suffice to identify $\boldsymbol{T}^j$.*

Proof. The likelihood of noisy labels can be formulated as a third-order tensor $\boldsymbol{L}$, where the (p, q, m) entry is

$$L_{pqm} = P(\bar{Y}^i = p, \bar{Y}^j = q, \bar{Y}^k = m). \tag{15}$$

According to Lemma 5, Eq. (15) can be expanded by Bayes Rule:

$$\begin{aligned} L_{pqm} &= \sum_{c_0, c_1 = 0}^{c_0, c_1 = 1} P(\bar{Y}^i = p, \bar{Y}^j = q, \bar{Y}^k = m \mid Y^j = c_0, Y^i = c_1) P(Y^j = c_0, Y^i = c_1) \\ &= \sum_{c_0, c_1 = 0}^{c_0, c_1 = 1} P(\bar{Y}^i = p \mid Y^j, Y^i) P(\bar{Y}^j = q \mid Y^j, Y^i) P(\bar{Y}^k = m \mid Y^j, Y^i) P(Y^j, Y^i). \end{aligned} \tag{16}$$

where we omit the value of Y^i, Y^j in the last eqution for simplicity. Let $\boldsymbol{M}^{(1)}_{4\times 2} = P(\bar{Y}^j \mid Y^i, Y^j)$, $\boldsymbol{M}^{(2)}_{4\times 2} = P(\bar{Y}^i \mid Y^i, Y^j)$, $\boldsymbol{M}^{(3)}_{4\times 2} = P(\bar{Y}^k \mid Y^i, Y^j)$ and $\boldsymbol{\Lambda}_{4\times 1} = P(Y^i, Y^j)$, and the Eq. (16) can be expressed as $L_{pqm} = \sum_{v=0}^{3} M^{(2)}_{vp} M^{(1)}_{vq} M^{(3)}_{vm} \Lambda_v$.

Let $\{\boldsymbol{A}^0, \boldsymbol{B}^0, \boldsymbol{C}^0, \boldsymbol{\Lambda}^0\}$ be a solution of $\{\boldsymbol{M}^{(1)}, \boldsymbol{M}^{(2)}, \boldsymbol{M}^{(3)}, \boldsymbol{\Lambda}\}$, which means it fulfils the likelihood equations (Eq. (16)), i.e.

$$L_{pqm} = \sum_{v=0}^{3} B^0_{vp} A^0_{vq} C^0_{vm} \Lambda^0_v. \tag{17}$$

Note that as stated in Lemma. 6, the first two rows and last two rows of $\boldsymbol{B}^0$ are identical respectively. Next, we will show that a different solution can be constructed by simply switching the corresponding rows in $\boldsymbol{A}^0$, $\boldsymbol{C}^0$ and $\boldsymbol{\Lambda}^0$, which is consistent with the result in [15, 33] when $\mathrm{Kr}(\boldsymbol{M}^{(2)}) = 1$. By letting

$$\boldsymbol{A}^1 = \begin{pmatrix} A^0_{10} & A^0_{11} \\ A^0_{00} & A^0_{01} \\ A^0_{30} & A^0_{31} \\ A^0_{20} & A^0_{21} \end{pmatrix}, \quad \boldsymbol{C}^1 = \begin{pmatrix} C^0_{10} & C^0_{11} \\ C^0_{00} & C^0_{01} \\ C^0_{30} & C^0_{31} \\ C^0_{20} & C^0_{21} \end{pmatrix}, \quad \text{and} \quad \boldsymbol{\Lambda}^1 = \begin{pmatrix} \Lambda^0_1 \\ \Lambda^0_0 \\ \Lambda^0_3 \\ \Lambda^0_2 \end{pmatrix},$$

then the Eq. (17) is equivalent to

$$
\begin{aligned}
L_{pqm} &= B^0_{0p}A^0_{0q}C^0_{0m}\Lambda^0_0 + B^0_{1p}A^0_{1q}C^0_{1m}\Lambda^0_1 + B^0_{2p}A^0_{2q}C^0_{2m}\Lambda^0_2 + B^0_{3p}A^0_{3q}C^0_{3m}\Lambda^0_3 \\
&= B^0_{1p}A^0_{0q}C^0_{0m}\Lambda^0_0 + B^0_{0p}A^0_{1q}C^0_{1m}\Lambda^0_1 + B^0_{3p}A^0_{2q}C^0_{2m}\Lambda^0_2 + B^0_{2p}A^0_{3q}C^0_{3m}\Lambda^0_3 \\
&= B^0_{1p}A^1_{1q}C^1_{1m}\Lambda^1_1 + B^0_{0p}A^1_{0q}C^1_{0m}\Lambda^1_0 + B^0_{3p}A^1_{3q}C^1_{3m}\Lambda^1_3 + B^0_{2p}A^1_{2q}C^1_{2m}\Lambda^1_2 \\
&= \sum_{v=0}^{3} B^0_{vp}A^1_{vq}C^1_{vm}\Lambda^1_v
\end{aligned}
\tag{18}
$$

where the second equation holds because the first two rows and last two rows of $\boldsymbol{B}^0$ are identical respectively, i.e. $B^0_{0p} = B^0_{1p}$ and $B^0_{2p} = B^0_{3p}$. Note that $\boldsymbol{A}^1, \boldsymbol{C}^1$ and $\boldsymbol{\Lambda}^1$ are consistent with the form of $\boldsymbol{M}^{(1)}, \boldsymbol{M}^{(3)}$ and $\boldsymbol{\Lambda}$ respectively. Then, according to Eq. (18), it can be readily observed that the $\{\boldsymbol{A}^1, \boldsymbol{B}^0, \boldsymbol{C}^1, \boldsymbol{\Lambda}^1\}$ is a new solution of $\{\boldsymbol{M}^{(1)}, \boldsymbol{M}^{(2)}, \boldsymbol{M}^{(3)}, \boldsymbol{\Lambda}\}$, hence the uniqueness is not guaranteed under this circumstance. As $\boldsymbol{M}^{(1)} = \begin{pmatrix} \boldsymbol{T}^j \\ \boldsymbol{T}^j \end{pmatrix}$ is not unique, then $\boldsymbol{T}^j$ is unidentifiable. □

F Proof of Theorem 4

Theorem 4. *Noisy labels $\{\bar{Y}^j, \bar{Y}^i\}$ and $P(\bar{Y}^i \mid Y^j)$ are sufficient to identify $\boldsymbol{T}^j$.*

Proof. The proof of Theorem 4 is much similar to that of Theorem 1, we can get a system of equations expressed as $\boldsymbol{E} = (\boldsymbol{T}^j)^\top \boldsymbol{P}\boldsymbol{M}$. The difference lies in that in Theorem 4, the matrix $\boldsymbol{M}$ which is parameterized by $P(\bar{Y}^i \mid Y^j)$ is given. Since $\boldsymbol{M}$ is invertible (Similar to Lemma. 4), the problem can be converted to a simple bilinear decomposition problem:

$$\boldsymbol{E}(\boldsymbol{M})^{-1} = (\boldsymbol{T}^j)^\top \boldsymbol{P},$$

i.e.

$$\begin{pmatrix} e_{00} & e_{01} \\ e_{10} & e_{11} \end{pmatrix} \begin{pmatrix} 1-\rho'_- & \rho'_- \\ \rho'_+ & 1-\rho'_+ \end{pmatrix}^{-1} = \begin{pmatrix} 1-\rho_- & \rho_- \\ \rho_+ & 1-\rho_+ \end{pmatrix}^\top \begin{pmatrix} 1-p & 0 \\ 0 & p \end{pmatrix}. \tag{19}$$

Solving the above bilinear decomposition problem, the unique solution can be obtained as:

$$p = \frac{(1-\rho'_-) - (e_{00} + e_{10})}{1-\rho'_- - \rho'_+}. \tag{20}$$

Substituting Eq. (20) into Eq. (19), then right multiplying $\boldsymbol{P}^{-1}$ on both side of the equation, the matrix $\boldsymbol{T}^j$ can be derived as:

$$\boldsymbol{T}^j = [\boldsymbol{E}(\boldsymbol{M})^{-1}(\boldsymbol{P})^{-1}]^\top,$$

which indicates that $\boldsymbol{T}^j$ is identifiable given label correlation $P(\bar{Y}^i \mid Y^j)$. □

G Limitations of Our Work

Our work still has certain limitations, including:

1) This work exploits the memorization effect in deep learning to perform sample selection for estimating probabilities of one noisy label when another clean label appears, while the memorization effect has not been found in other traditional machine learning methods, and therefore, our estimation can not applied to such learning methods.

2) This work estimates occurrence probabilities using frequency counting. Although this estimation error will converge to zero exponentially fast [4], when the number of one label appearing is too small, e.g. less than 50, the estimation of the transition matrix for this class label is still difficult to be accurate.

H Broader Impact

In this era of deep learning, datasets are becoming bigger and bigger. However, it is much expensive to obtain large-scale dataset with high-quality annotated labels. Alternatively, people can collect labels based on non-expert annotators or automated labeling methods, e.g. web search and user tags, especially for non-profit research institutions and startups. However, these cheap annotation methods are likely to introduce noisy labels to the collected datasets, which may pose serious threats to the supervised learning algorithm designed for clean data. Label-noise learning, therefore, becomes a more and more important topic recently.

Existing label-noise learning methods typically focus on the single-label case by assuming that only one label is corrupted. In real applications, an example is usually associated with multiple class labels, which could be corrupted simultaneously with their respective different probabilities. The proposed method is to estimate the transition matrix using label correlations in noisy multi-label datasets. The transition matrix is essential to building the statistically consistent label-noise learning algorithms. We have shown that our method usually leads to a better estimation compared to the current estimator and can improves the classification performance of statistically consistent label-noise learning applications, which should has a positive impact on science, society, and economy. Hence, our approach can reduce the need for accurate labeling, and potentially, it may have a negative impact on the salary of label workers.

I Ablation study about the sample selection threshold

The sample selection we adopted [30, 17] is to estimate this clean probability of samples by modeling sample loss values with a GMM model using the Expectation-Maximization algorithm. If the clean sample can be distinguished according to loss values, and its estimated probability is accurate, the best threshold will be about 0.5. Hence, $\tau = 0.5$ is a typical value in related works [30, 17], and we follow this practice in our experiments. In this section, we conduct the ablation study about the sample selection threshold experiments on noisy VOC2007 datasets. As shown in Fig. **??** and Fig. **??**, $\tau = 0.5$ is a good choice both according to mAP scores on the noisy validation set and according to mAP scores on the clean test dataset.

(a) (b)

Figure 1: (a) the mAP scores on the noisy validation set of Reweight methods with our estimator with $\tau = 0.4, 0.5, 0.6$; (b) the mAP scores on the clean test dataset of Reweight methods with our estimator with $\tau = 0.4, 0.5, 0.6$.

J Comparison with different loss correction ways

Many statistically consistent algorithms [20, 26, 38] consist of a two-step training procedure. The first step estimates the transition matrix and the second step builds statistically consistent algorithms via modifying loss functions. Our proposed estimator can be seamlessly embedded into their frameworks. In this section, we compare classification performance of applying our estimation method to four different loss correction ways: Reweight [20] (named Reweight-Ours), Backward [25, 26, 40] (named Backward-Ours), Forward [26] (named Forward-Ours) and T-Reversion [38] (named Reversion-Ours) on Pascal-VOC2007 dataset. As shown in Tab. 8, T-Reversion with our estimation achieves the best

performance in most cases, which may be because it can further tune the transition matrices via classification learning.

Table 8: Comparison for classification performance with different loss correction ways on Pascal-VOC2007 dataset. The best peformances are in **bold**.

	Noise rates (ρ_-, ρ_+)	(0,0.2)	(0,0.6)	(0.2,0)	(0.6,0)	(0.1,0.1)	(0.2,0.2)	(0.017,0.2)	(0.034,0.4)
mAP	Reweight-Ours	84.43±0.46	78.72±0.41	84.08±0.24	**74.46±0.56**	84.03±0.29	80.44±0.52	84.09±0.62	80.97±1.03
	Backward-Ours	83.41±0.18	67.22±2.97	81.13±0.45	63.82±1.36	79.00±0.88	70.44±1.80	81.27±0.37	69.22±1.96
	Forward-Ours	84.96±0.28	**80.23±0.25**	83.41±0.72	71.45±3.65	83.19±0.24	76.77±2.14	84.31±0.51	80.46±1.02
	Reversion-Ours	**84.99±0.33**	79.26±0.51	**84.37±0.24**	74.44±1.16	**84.49±0.36**	**80.61±0.73**	**84.98±0.11**	**82.14±0.10**
OF1	Reweight-Ours	78.62±0.58	65.68±1.67	80.85±0.25	67.43±4.65	79.64±0.29	75.52±0.86	79.25±0.52	74.35±1.65
	Backward-Ours	74.98±0.56	16.35±5.06	78.55±0.33	14.57±0.29	74.35±1.12	63.60±4.39	71.68±0.78	36.38±4.13
	Forward-Ours	**80.09±0.45**	**70.38±0.29**	80.44±0.50	59.35±10.36	79.77±0.35	73.92±2.33	**79.92±0.62**	75.18±1.24
	Reversion-Ours	79.07±0.94	63.97±3.14	**81.20±0.32**	**68.74±4.18**	**80.22±0.40**	**75.68±1.16**	79.83±0.33	**75.23±0.56**
CF1	Reweight-Ours	76.86±0.48	61.29±1.94	77.89±0.42	66.79±2.50	78.04±0.40	**74.08±0.79**	77.28±0.48	**72.18±0.74**
	Backward-Ours	70.64±0.59	15.01±4.66	75.48±0.84	14.55±0.26	66.26±2.15	46.19±8.35	65.56±1.59	23.78±2.95
	Forward-Ours	**77.85±0.56**	**64.64±1.69**	77.14±0.35	62.26±3.06	77.72±0.25	70.45±4.38	77.74±0.80	71.76±1.45
	Reversion-Ours	76.82±0.90	58.56±3.72	**78.26±0.51**	**66.89±1.46**	**78.55±0.58**	73.94±1.06	**77.86±0.33**	72.12±1.16

K Comparison with different base learning algorithms

Recently, many advanced multi-label learning algorithms [45, 47, 45, 3], which designed exquisite networks for multi-label learning, have been proposed, and they perform well on the clean dataset. As their loss functions are similar to Eq. (8), we can also apply reweight method with our estimator to their frameworks. In this section, we compare the classification performance of applying Reweight method with our estimator to three different base learning algorithms: Standard (named "Reweight-Ours"), AGCN [45] (named "AGCN-R-Ours") and CSRA [47] (named "CSRA-R-Ours") on Pascal-VOC2007 dataset. Besides, to show the importance of the consistent algorithms, we also present the performance of base learning algorithms here. As shown in Tab. 9, the consistent algorithms with our estimator can help base learning algorithms perform much better, especially on the OF1 and CF1 metrics. Besides, AGCN-R-Ours and CSRA-R-Ours perform better than Reweight-Ours in some cases, which means that the consistent algorithms with our estimator can work well with more advanced network.

Table 9: Comparison for classification performance with different base learning algorithms on Pascal-VOC2007 dataset. The best peformances are in **bold**.

	Noise rates (ρ_-, ρ_+)	(0,0.2)	(0,0.6)	(0.2,0)	(0.6,0)	(0.1,0.1)	(0.2,0.2)	(0.017,0.2)	(0.034,0.4)
mAP	Standard	84.25±1.07	77.16±0.94	82.70±0.54	68.65±1.57	83.07±0.45	78.87±0.52	83.92±0.59	80.97±0.42
	AGCN	83.24±0.67	75.50±0.56	81.09±0.51	66.47±1.29	81.09±0.48	73.79±0.76	82.21±0.42	76.55±1.11
	CSRA	85.11±0.51	79.47±1.22	82.93±0.65	67.36±2.25	83.69±0.69	78.10±0.53	84.94±0.36	81.51±0.14
	Reweight-Ours	84.43±0.46	78.72±0.41	84.08±0.24	74.46±0.56	84.03±0.29	80.44±0.52	84.09±0.62	80.97±1.03
	AGCN-R-Ours	85.07±0.56	**80.35±0.69**	83.41±0.68	66.07±4.32	83.79±0.41	78.18±2.53	84.38±0.30	80.76±0.91
	CSRA-R-Ours	**85.83±0.53**	77.64±3.21	**84.66±0.60**	**75.41±1.77**	**84.68±0.44**	**81.01±0.57**	**85.74±0.29**	**82.64±0.35**
OF1	Standard	75.24±1.40	32.02±5.49	78.85±0.43	15.08±0.25	79.24±0.43	75.85±0.84	75.98±1.04	59.67±1.65
	AGCN	74.92±1.02	30.97±3.78	75.45±2.06	16.85±0.56	78.69±0.31	72.64±0.51	75.16±0.58	56.56±1.64
	CSRA	76.94±1.03	33.65±2.73	77.71±1.23	15.94±0.32	80.36±0.53	76.92±0.34	77.91±0.63	62.19±1.97
	Reweight-Ours	78.62±0.58	65.68±1.67	80.85±0.25	**67.43±4.65**	79.64±0.29	75.52±0.86	79.25±0.52	74.35±1.65
	AGCN-R-Ours	80.28±0.41	**71.36±2.52**	79.18±1.38	46.77±3.34	79.06±0.92	73.14±2.63	79.49±0.76	75.78±2.11
	CSRA-R-Ours	**80.61±0.80**	61.48±3.61	**81.22±0.45**	57.48±9.33	**80.43±0.44**	**76.04±0.80**	**81.22±0.34**	**77.21±0.45**
CF1	Standard	72.53±1.11	30.64±3.90	76.83±0.65	14.97±0.24	75.86±1.23	70.68±1.76	73.11±0.54	52.07±2.34
	AGCN	73.45±1.04	33.41±1.65	72.65±1.97	16.67±0.55	76.20±0.51	69.09±0.49	72.81±1.02	55.09±3.28
	CSRA	74.10±0.56	33.44±3.65	75.28±1.32	15.71±0.23	77.52±0.94	73.44±0.62	74.98±0.48	58.60±2.24
	Reweight-Ours	76.86±0.48	61.29±1.94	77.89±0.42	**66.79±2.50**	78.04±0.40	74.08±0.79	77.28±0.48	72.18±0.74
	AGCN-R-Ours	**78.74±0.95**	**68.58±3.35**	77.62±0.65	51.45±1.57	78.02±0.61	71.84±3.18	78.09±0.24	74.32±1.45
	CSRA-R-Ours	78.65±0.75	59.28±3.71	**78.32±0.73**	62.59±6.25	**78.98±0.45**	**75.28±0.37**	**79.52±0.39**	**74.38±0.98**