# CCMLN: COMBINATORIAL CORRECTION FOR MULTI-LABEL CLASSIFICATION WITH NOISY LABELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Multi-label classification aims to learn classification models from instances associated with multiple labels. It is pivotal to learn and utilize the label dependence among multiple labels in multi-label classification. As a result of todays big and complex data, noisy labels are inevitable, making it looming to target multi-label classification with noisy labels. Although the importance of label dependence has been shown in multi-label classification with clean labels, it is challenging and unresolved to bring label dependence to the problem of multi-label classification with noisy labels. The issues are, that we do not understand why the label dependence is helpful in the problem, and how to learn and utilize the label dependence only using training data with noisy multiple labels. In this paper, we bring label dependence to tackle the problem of multi-label classification with noisy labels. Specifically, we first provide a high-level understanding of why label dependence helps distinguish the examples with clean/noisy multiple labels. Benefiting from the memorization effect in handling noisy labels, a novel algorithm is then proposed to learn the label dependence by only employing training data with noisy multiple labels, and utilize the learned dependence to help correct noisy multiple labels to clean ones. We prove that the use of label dependence could bring a higher success rate for recovering correct multiple labels. Empirical evaluations justify our claims and demonstrate the superiority of our algorithm.

## 1 INTRODUCTION

Multi-label classification assigns a set of *multiple labels* for each instance (Zhang & Zhou, 2013). As a practical learning paradigm, multi-label classification has been widely applied in various domains, ranging from computer vision (Chen et al., 2019b) and natural language processing (Onoe et al., 2021), to recommendation systems (Zhang et al., 2021a) and bioinformatics (Cheng et al., 2017). Consensually, compared with multi-class classification (He et al., 2016), where each instance is assigned with a single label, multi-label classification is more challenging (Liu et al., 2017). Plenty of advanced methods are proposed in recent years for multi-label classification (Zhu & Wu, 2021; Ridnik et al., 2021; Gao & Zhou, 2021; Zhao et al., 2021; Liu et al., 2018; Chheda et al., 2021).

The great majority of the methods assume that training data are annotated precisely. However, noisy labels are *inevitable* in multi-label classification (Liu et al., 2021), especially for classification with big and complex data. They may be resulted by unintentional mistakes of manual and automatic annotators (Veit et al., 2017; Zheng et al., 2020), or intentional corruptions on clean labels (Vahdat, 2017; Pleiss et al., 2020). Noisy labels severely impair the generalization of learned models, *over-parameterized deep models* in particular (Kim et al., 2019). A straightforward way to address the problem of multi-label classification with noisy labels is to treat each label *in isolation* and convert the multi-label problem into a number of binary classification problems. Afterward, the methods in multi-class classification with noisy labels (Han et al., 2020; Song et al., 2022) are applied to train *independent* binary classifiers, which capture instance-label dependence robustly to strengthen classification. This way is a remedy to handle noisy labels, but ignores the label dependence among multiple labels. It is essential to learn and utilize the label dependence in multi-label classification (Zhang & Zhang, 2010; Hang & Zhang, 2021; Cui et al., 2020; Li et al., 2022).

Prior works (Ye et al., 2020; Chen et al., 2021; Wang et al., 2016) illustrate the successes of considering the label dependence among multiple labels in multi-label classification with clean labels. In different ways, e.g., helping learn *inter-dependent classifiers* (Chen et al., 2019b), the label de-

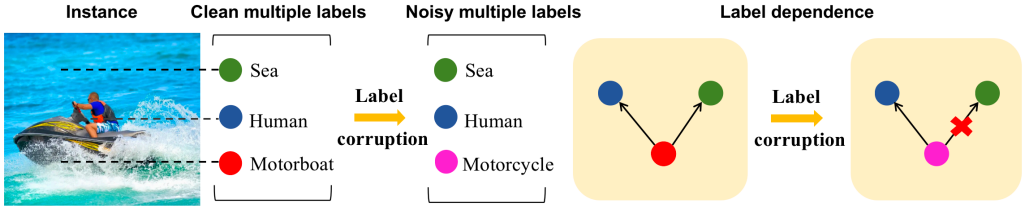


Figure 1: The illustration of why the label dependence among multiple labels help distinguish the examples with noisy/clean multiple labels. The arrow presents the label dependence between a label pair. For the labels "a" and "b", "a → b" means that, when "a" appears, "b" will also occur with high probability. The example comes from web search. The set of clean multiple labels is {Sea, Human, Motorboat}, where the label dependence is strong with both "Motorboat → Sea" and "Motorboat → Human". However, due to label corruption, Motorboat is flipped to be Motorcycle, which causes "Motorcycle ↛ Sea". Therefore, the label dependence among noisy multiple labels is *weaker* than the label dependence among corresponding clean ones.

pendence can be used to boost the learning of the instance-label dependence, which improves final classification. Inspired by the successes, it is concerned that the label dependence could be exploited to handle the problem of multi-label classification with noisy labels. However, there are few attempts before for this important problem. At least *three questions* make the solution remain mysterious. First, in intuition, we need to understand why the label dependence is helpful for the problem. Second, in technique, we need to know how to learn and utilize the label dependence into the problem. As we only have training data with noisy labels, both the accurate catch and application of the label dependence are challenging. Third, in verification, we need to know what improvements the label dependence can bring.

In this paper, we answer the three questions one by one. The first answer is illustrated in Figure 1. That is, compared with noisy multiple labels, the label dependence among clean multiple labels is *stronger with high probability*. Therefore, such dependence could help distinguish the examples with noisy/clean multiple labels for our problem. The second answer is given by the proposed combinatorial correction for multi-label classification with noisy labels (*aka* CCMLN). Specifically, CCMLN inherits the *memorization effect* in handling noisy labels (Arpit et al., 2017; Jiang et al., 2018; Wang et al., 2021): the deep model would firstly memorize the training examples with clean labels, leading to reliable model predictions in early training. In CCMLN, the label dependence is learned by a dynamic graph (Ye et al., 2020), and then applied to correcting noisy multiple labels. In more detail, the *combinatorial score* in CCMLN is proposed to *holistically measures* the instance-label and label dependences in an example. The stronger instance-label and label dependences make a larger combinatorial score. We compare the ratio between the combinatorial scores of the example with noisy multiple labels and its variant with predicted multiple labels, with an easily determined threshold. The noisy multiple labels are corrected or changeless based on the comparison result. Benefiting from the memorization effect, both dependence learning and multi-label correction are useful. Besides, they fulfill *a positive cycle* (Bai & Liu, 2021). Namely, better dependence learning results in better multi-label correction, and better multi-label correction makes better dependence learning, leading to final enhanced classification.

The third answer is given by both theoretical analyses and empirical evaluations. Theoretically, we show that the additional use of label dependence brings higher probability to handle noisy multiple labels successfully than the sole use of instance-label dependence under some conditions. Empirically, we demonstrate the power of label dependence by experiments and show that, in most situations, CCMLN outperforms comparison methods with large margins.

## 2 PRELIMINARIES

**Problem statement.** Let $\mathcal{X} \in \mathbb{R}^d$ denote the input space and $\mathcal{Y} \in \{l_1, \cdots, l_q\}$ denote the label space with $q$ class labels. An example with multiple labels is denoted as $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x} \in \mathcal{X}$ is the feature vector of an instance, and $\boldsymbol{y} \subseteq \mathcal{Y}$ is its set of associated labels. Denote the size of the label set $\boldsymbol{y}$ as $|\boldsymbol{y}|$. For the feature vector $\boldsymbol{x}$, its label set $\boldsymbol{y}$ may be corrupted and is flipped into $\bar{\boldsymbol{y}} \subseteq \mathcal{Y}$ with $|\bar{\boldsymbol{y}}| = |\boldsymbol{y}|$. We utilize a class-dependent noise transition matrix $T$ (Pene et al., 2021; Shu et al., 2020) to characterize the label flip process. Formally, for any $i \neq j$, $T_{ij} = \mathbb{P}(l_j \in \bar{\boldsymbol{y}} \wedge l_i \notin \bar{\boldsymbol{y}} | l_j \notin \boldsymbol{y} \wedge l_i \in \boldsymbol{y})$ represents the probability of the $i$-th class label to be flipped into the $j$-th class label. Consider a

noisy multi-label dataset comprising several examples $(\boldsymbol{x}, \bar{\boldsymbol{y}})$. The aim is to learn a classification model *robustly* by *only* using the noisy dataset. Given an instance in testing, with the learned model, we can predict its relevant label set precisely.

It should be noted that some works employ another problem setting that the total number of multiple labels can be changed after label flipping, which is referred to as multi-label classification with missing or redundant labels. For classification with missing labels, it is not accurate to consider it as classification with noisy labels, since all annotated labels are correct (Yu et al., 2014; Wu et al., 2015). Besides, for classification with redundant labels, it is normally called partial multi-label learning (Xie & Huang, 2018), which is different from the problem setting of this paper, as detailed in Appendix C.4. Our setting, i.e., the total number of labels is preserved after label flipping, is realistic. In many practical situations, it is easy to determine the number of objects in an image, in particular with object detection techniques. In contrast, it can be harder to annotate the objects perfectly, resulting in noisy labels.

**Preparation technology.** As discussed, we need both the instance-label dependence and the label dependence among multiple labels. Given an example $(\boldsymbol{x}, \boldsymbol{y})$, for the instance-label dependence, it can be learned with the conditional probability of $l_i \in \boldsymbol{y}$ given $\boldsymbol{x}$ according to model's probability outputs. For the label dependence among multiple labels, it is often estimated by counting the occurrence of label pairs in training data (Chen et al., 2019b).

Recently, the graph convolutional network (GCN) is used in multi-label classification and achieves great successes (Chen et al., 2019b; Ye et al., 2020; Chen et al., 2021). The advantage of the GCN-based methods is that they can capture the instance-label and label dependences simultaneously during training. In this paper, we inherit the advantage of the GCN-based methods and build CCMLN based on ADDGCN (Ye et al., 2020). ADDGCN designs a semantic attention module (SAM) to estimate the content-aware class-label representations for each class from the extracted feature map. The representations are fed into a GCN module (GCNM) for final classification. We provide the technical details of ADDGCN (Ye et al., 2020) in Appendix C.1. Before delving into the next section, readers only need to remember that the instance-label and label dependences can be learned during training. Note that we also review prior works on multi-class classification with noisy labels and multi-label classification with clean/noisy labels in Appendix C.2 and Appendix C.3.

## 3 Proposed Method

### 3.1 Combinatorial Judgment in Multi-Label Classification

**Combinatorial score.** We begin with an example with clean multiple labels. Given an example $(\boldsymbol{x}, \boldsymbol{y})$, we can measure the instance-label dependence $S^f$, and the label dependence $S^l$. Denote the variable of clean multiple labels by $\boldsymbol{Y}$. Mathematically, we define two dependences as $S^f_{\boldsymbol{z}}(\boldsymbol{x}) := \sum_{\{\boldsymbol{Y}=\boldsymbol{z}, l_i \in \boldsymbol{z}\}} \mathbb{P}(l_i|\boldsymbol{x})$ and $S^l_{\boldsymbol{z}}(\boldsymbol{x}) := \sum_{\{\boldsymbol{Y}=\boldsymbol{z}, l_i, l_j \in \boldsymbol{z}\}} \frac{1}{2} [\mathbb{P}(l_j|l_i, \boldsymbol{x}) + \mathbb{P}(l_i|l_j, \boldsymbol{x})]$. The combinatorial score of the example $(\boldsymbol{x}, \boldsymbol{y})$ considers two dependences at the same time. Formally, we denote the combinatorial score of $(\boldsymbol{x}, \boldsymbol{y})$ as $S_{\boldsymbol{y}}(\boldsymbol{x})$ and define it as

$$S_{\boldsymbol{y}}(\boldsymbol{x}) := S^f_{\boldsymbol{y}}(\boldsymbol{x}) + S^l_{\boldsymbol{y}}(\boldsymbol{x}). \tag{1}$$

Afterward, denote the variable of noisy multiple labels by $\bar{\boldsymbol{Y}}$. For the example with noisy multiple labels, i.e., $(\boldsymbol{x}, \bar{\boldsymbol{y}})$, the instance-label dependence and label dependence are measure by $\bar{S}^f_{\boldsymbol{z}}(\boldsymbol{x}) := \sum_{\{\bar{\boldsymbol{Y}}=\boldsymbol{z}, l_i \in \boldsymbol{z}\}} \mathbb{P}(l_i|\boldsymbol{x})$ and $\bar{S}^l_{\boldsymbol{z}}(\boldsymbol{x}) := \sum_{\{\bar{\boldsymbol{Y}}=\boldsymbol{z}, l_i, l_j \in \boldsymbol{z}\}} \frac{1}{2} [\mathbb{P}(l_j|l_i, \boldsymbol{x}) + \mathbb{P}(l_i|l_j, \boldsymbol{x})]$. Accordingly, the combinatorial score of the example $(\boldsymbol{x}, \bar{\boldsymbol{y}})$ is denoted by $\bar{S}_{\bar{\boldsymbol{y}}}(\boldsymbol{x})$, which is defined as $\bar{S}_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) := \bar{S}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) + \bar{S}^l_{\bar{\boldsymbol{y}}}(\boldsymbol{x})$. Note that, during training, we cannot access $\bar{S}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})$ and $\bar{S}^l_{\bar{\boldsymbol{y}}}(\boldsymbol{x})$. Instead, the estimated posterior probabilities are used. We denote the estimations of $\bar{S}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})$ and $\bar{S}^l_{\bar{\boldsymbol{y}}}(\boldsymbol{x})$ as $\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})$ and $\hat{\bar{S}}^l_{\bar{\boldsymbol{y}}}(\boldsymbol{x})$. The estimation of the combinatorial score is $\hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) = \hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) + \hat{\bar{S}}^l_{\bar{\boldsymbol{y}}}(\boldsymbol{x})$. With preparation technology discussed in Section 2 and Appendix C.1, $\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})$ and $\hat{\bar{S}}^l_{\bar{\boldsymbol{y}}}(\boldsymbol{x})$ can be obtained.

**Combinatorial correction.** For the example $(\boldsymbol{x}, \bar{\boldsymbol{y}})$, we feed it into the deep network $h$ included in ADDGCN (Ye et al., 2020). The memorization effect in handling noisy labels (Jiang et al., 2018; Liu et al., 2020b) shows that the deep network would first memorize the training data with clean labels and then the training data with noisy labels. Therefore, early in training, the outputs of the

deep network are relatively reliable, which can be used for label correction. For $(\boldsymbol{x}, \bar{\boldsymbol{y}})$, we denote its set of predicted multiple labels as $\boldsymbol{y}^*$. Here, the set of predicted labels is obtained with the top $|\bar{\boldsymbol{y}}|$ predictions based on model's probability outputs.

Recall that the combinatorial score of an example holistically measures the instance-label dependence and label dependence among multiple labels simultaneously. From both human and machine cognition, if an example is annotated accurately, both dependences should be strong (Zhang & Zhang, 2010; Hang & Zhang, 2021; Yu & Zhang, 2021; Li et al., 2016; Chen et al., 2019b) with high probability. Namely, the combinatorial score is large. We propose to check the ratio between the combinatorial score on $(\boldsymbol{x}, \bar{\boldsymbol{y}})$ and combinatorial score on $(\boldsymbol{x}, \boldsymbol{y}^*)$. Specifically, we check

**Algorithm 1** Combinatorial Correction.

**Input**: $(\boldsymbol{x}, \bar{\boldsymbol{y}})$, $h$, and $\hat{\delta}$.
**Output**: $\bar{\boldsymbol{y}}_{new}$.
$\boldsymbol{y}^* = h(\boldsymbol{x})$
$\kappa(h, \boldsymbol{x}, \bar{\boldsymbol{y}}) = \hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) / \hat{\bar{S}}_{\boldsymbol{y}^*}(\boldsymbol{x})$
**if** $\kappa(h, \boldsymbol{x}, \bar{\boldsymbol{y}}) \leq \hat{\delta}$ **then**
  $\bar{\boldsymbol{y}}_{new} = \boldsymbol{y}^*$
**else**
  $\bar{\boldsymbol{y}}_{new} = \bar{\boldsymbol{y}}$
**end if**

$$\kappa(h, \boldsymbol{x}, \bar{\boldsymbol{y}}) = \hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) / \hat{\bar{S}}_{\boldsymbol{y}^*}(\boldsymbol{x}). \tag{2}$$

We compare this ratio with a predetermined threshold $\hat{\delta}$. The value of $\hat{\delta}$ is given in the next subsection. If $\kappa(h, \boldsymbol{x}, \bar{\boldsymbol{y}}) \leq \hat{\delta}$, we flip the labels $\bar{\boldsymbol{y}}_{new} = \boldsymbol{y}^*$. Otherwise, the labels remain unchanged with $\bar{\boldsymbol{y}}_{new} = \bar{\boldsymbol{y}}$. The detailed algorithm of combinatorial correction for multi-label classification with noisy labels (*aka* CCMLN) is provided in Algorithm 1. After combinatorial correction for noisy labels, we use $(\boldsymbol{x}, \bar{\boldsymbol{y}}_{new})$ to train the deep network $h$ based on ADDGCN (Ye et al., 2020).

### 3.2 Theoretical Insights

We extend the Tsybakov condition (Zheng et al., 2020; Bahri et al., 2020; Gao et al., 2016) from multi-class classification to multi-label classification. Specifically, denote by $\boldsymbol{a}_{\boldsymbol{x}}$ the label set predicted based on $S^f(\boldsymbol{x})$ with $\boldsymbol{a}_{\boldsymbol{x}} := h^*(\boldsymbol{x}) = \arg\max_{\boldsymbol{z}} S^f_{\boldsymbol{z}}(\boldsymbol{x})$. Besides, denote by $\boldsymbol{b}_{\boldsymbol{x}}$ the second best prediction with $\boldsymbol{b}_{\boldsymbol{x}} := \arg\max_{\boldsymbol{z} \neq \boldsymbol{a}_{\boldsymbol{x}}} S^f_{\boldsymbol{z}}(\boldsymbol{x})$. The maximum length of a label set is denoted as $m$ ($m \ll q$). In this paper, we call the predicted label set by the Bayes optimal classifier for an instance as the correct label set.

**Definition 1 (Tsybakov condition on instance-label dependence)** *$\exists C_1, \lambda_1 > 0$ and $\exists t_0 \in (0, m]$, such that for all $t \leq t_0$, we have*

$$\mathbb{P}[S^f_{\boldsymbol{a}_{\boldsymbol{x}}}(\boldsymbol{x}) - S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) \leq t] \leq C_1 t^{\lambda_1}. \tag{3}$$

**Definition 2 (Combinatorial Tsybakov condition)** *$\exists C_2, \lambda_2 > 0$, and $\exists t_0 \in (0, m]$, such that for all $t \leq t_0$, we have*

$$\mathbb{P}[S_{\boldsymbol{a}_{\boldsymbol{x}}}(\boldsymbol{x}) - S_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) \leq t] \leq C_2 t^{\lambda_2}. \tag{4}$$

**Remark 1** *Definition 1 stipulates that the uncertainty of $S^f$ is bounded. The margin region that is close to the decision boundary has a bounded volume. Definition 2 shares the similar idea and bound the uncertainty of $S$.*

**Theorem 1** *Suppose $S(\boldsymbol{x})$ fulfills the combinatorial Tsybakov condition for constants $C_2$, $\lambda_2 > 0$, and $t_0 \in (0, m]$. We define $\epsilon := \max_{\boldsymbol{x}, \boldsymbol{z}} \left[ |\hat{\bar{S}}^f_{\boldsymbol{z}}(\boldsymbol{x}) - \bar{S}^f_{\boldsymbol{z}}(\boldsymbol{x})|, |\hat{\bar{S}}^l_{\boldsymbol{z}}(\boldsymbol{x}) - \bar{S}^l_{\boldsymbol{z}}(\boldsymbol{x})|, |\bar{S}^l_{\boldsymbol{z}}(\boldsymbol{x}) - S^l_{\boldsymbol{z}}(\boldsymbol{x})| \right]$ and $\tau := \min_i T_{ii}$. We analyze two cases:*

*(1) If $\bar{\boldsymbol{y}}$ is corrected by $\kappa(h, \boldsymbol{x}, \bar{\boldsymbol{y}})$ with $\hat{\delta}$, let $\delta_1 = \min \left[ \frac{\tau S_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) + \sum_{l_j \in \bar{\boldsymbol{y}}} \sum_{i \neq j} T_{ij} \mathbb{P}(l_i | \boldsymbol{x})}{\hat{\bar{S}}_{\boldsymbol{y}^*}(\boldsymbol{x})} \right]$ and $\rho_1 := |\hat{\delta} - \delta_1|$. Assume that $\epsilon \leq \frac{t_0 \tau - \rho_1 m}{3}$. Then, $\mathbb{P}[\bar{\boldsymbol{y}}_{new} = h^*(\boldsymbol{x}), \bar{\boldsymbol{y}}$ is $flipped]$ is at least $1 - C_2[O(\max(\epsilon, \rho_1))]^{\lambda_2} - \mathbb{P}[\boldsymbol{a}_{\boldsymbol{x}} \neq \{\boldsymbol{y}^*, \bar{\boldsymbol{y}}\}]$.*

*(2) If $\bar{\boldsymbol{y}}$ is not corrected by $\kappa(h, \boldsymbol{x}, \bar{\boldsymbol{y}})$ with $\hat{\delta}$, let $\delta_2 = \max \left[ \frac{\hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\tau S_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) + \sum_{l_j \in \boldsymbol{y}^*} \sum_{i \neq j} T_{ij} \mathbb{P}(l_i | \boldsymbol{x})} \right]$ and $\rho_2 := |\hat{\delta} - \delta_2|$. Assume that $\epsilon \leq \frac{t_0 \delta_2^2 \tau - \rho_2 m - \rho_2^2 m}{3\delta_2^2}$. Then, $\mathbb{P}[\bar{\boldsymbol{y}}_{new} = h^*(\boldsymbol{x}), \bar{\boldsymbol{y}}$ is $accepted]$ is at least $1 - C_2[O(\max(\epsilon, \rho_2))]^{\lambda_2} - \mathbb{P}[\boldsymbol{a}_{\boldsymbol{x}} \neq \{\boldsymbol{y}^*, \bar{\boldsymbol{y}}\}]$.*

The proof of Theorem 1 is provided in Appendix B.1. Theorem 1 extends the theoretical results of (Zheng et al., 2020) to multi-label classification with noisy labels. It claims that, even though with noisy multiple labels, the combinatorial correction has a guaranteed success rate to make proper corrections. Besides, if we can reasonably approximate the optimal $\delta$ with $\hat{\delta}$, our algorithm flips noisy multiple labels to correct ones with a good chance. Below, as a corollary of Theorem 1, we show that, there are certain circumstances, the use of combinatorial scores has a better chance to make corrections satisfactorily, than the sole use of instance-label dependence.

**Corollary 1** *Suppose that $S(\boldsymbol{x})$ fulfills the combinatorial Tsybakov condition. Denote the set threshold $\hat{\delta}$ and optimal threshold $\delta$. We define $\rho := \max|\hat{\delta} - \delta|$. We have that, $\exists\epsilon$ and $\rho$, if $C_2[O(\max(\epsilon,\rho))]^{\lambda_2} < C_1[O(\max(\epsilon,\rho))]^{\lambda_1}$, combinatorial correction brings higher probability to handle noisy labels successfully than instance-label dependence.*

The proof of Corollary 1 is provided in Appendix B.2. Corollary 1 claims that there exist cases where combinatorial scores better combat noisy labels. Note that, from a theoretical view, we do not state that combinatorial scores can always work better. Nevertheless, with the determination of the threshold $\hat{\delta}$, combinatorial scores can perform better in experiments, which demonstrates the help of label dependence to handle noisy multiple labels.

# 4 Experiments

## 4.1 Experimental Setup

**Datasets.** We verify the effectiveness of the proposed method on the synthetic noisy versions of three datasets, i.e., Pascal-VOC 2007 (Everingham et al., 2008), Pascal-VOC 2012 (Everingham et al., 2008), and MS-COCO (Lin et al., 2014). Pascal-VOC 2007 contains 5,011 images in train and validation sets, while Pascal-VOC 2012 consists of 11,540 images in train and validation sets. The images come from 20 common object categories. For Pascal-VOC 2007 and Pascal-VOC 2012, we train methods using the noisy training and validation sets, and evaluate them on the test set of Pascal-VOC 2007 that has 4,952 images (Gao & Zhou, 2021). MS-COCO contains 82,081 training images and 40,137 validation images from 80 common object categories. As did in (Zhao et al., 2021; Chen et al., 2019a; Ye et al., 2020; Zhu & Wu, 2021), we evaluate the performance of methods using validation images.

**Noisy-label generation.** The class-dependent noise transition matrix $T$ (Patrini et al., 2017; Hendrycks et al., 2018; Shu et al., 2020; Zhang et al., 2021d) is used to corrupt the three datasets. Here, for any $i \neq j$, $T_{ij} = \mathbb{P}(l_j \in \bar{\boldsymbol{y}} \wedge l_i \notin \bar{\boldsymbol{y}} | l_j \notin \boldsymbol{y} \wedge l_i \in \boldsymbol{y})$ represents the probability of the $i$-th class label to be flipped into the $j$-th class label. We consider both symmetric (abbreviated as Sym.) and pairflip (abbreviated as Pair.) noise settings (Han et al., 2018). The details of the transition matrix are provided in Appendix D.2. For symmetric noise, the noise rate is set to 30%, 40%, and 50% . For pairflip noise, the noise rate is set to 20%, 30%, and 40%.

**Baselines.** We exploit three types of baselines in total. Specifically, Type-I baselines contain the methods that are designed for multi-label classification with clean labels. Type-II baselines consider the methods for multi-class classification with noisy labels. Type-III baselines consider the methods that focus on multi-label classification with noisy labels. It should be noted that, there are relatively few methods belonging to this type (Liu et al., 2021). More advanced methods belonging to Type-III baselines need to be investigated (Liu et al., 2021), which is also our focus in this paper.

In more detail, Type-I baselines include CSRA (Zhu & Wu, 2021) and ADDGCN (Ye et al., 2020). Type-II baselines include APL (Ma et al., 2020), CDR (Xia et al., 2021), and JOINT (Tanaka et al., 2018). Type-III baselines include WSIC (Hu et al., 2019) and CCMN (Xie & Huang, 2022). As a simple baseline, we compare our method with the standard deep network that directly trains on noisy datasets (abbreviated as BCE). We detail all baselines in Appendix D.1.

**Network & Optimizer.** We use a ResNet-50 network (He et al., 2016) pretrained on ImageNet as the backbone for all methods. We train the models for 30 epochs in total. We utilize Adam (Kingma & Ba, 2014) for the network optimization. The batch size is set to 128 for all the datasets. The learning rate is fixed to $5 \times 10^{-5}$. The images in Pascal-VOC 2007, Pascal-VOC 2012, and MS-COCO resize to $224 \times 224$. Note that, to make experiments more comprehensive, we also employ

| Metrics | Methods / Noise | Sym. 30% | Sym. 40% | Sym. 50% | Pair. 20% | Pair. 30% | Pair. 40% |
|---|---|---|---|---|---|---|---|
| mAP ↑ | BCE | 64.50±1.20 | 58.65±2.16 | 48.19±0.23 | 71.77±1.15 | 60.94±4.25 | 48.72±2.13 |
| | CSRA | 66.99±0.48 | 59.62±0.61 | 46.97±0.48 | 72.45±0.69 | 63.58±1.48 | 52.72±1.52 |
| | ADDGCN | 63.89±0.94 | 55.75±1.98 | 44.14±1.37 | 71.02±0.95 | 61.05±0.06 | 50.18±2.70 |
| | APL | 66.79±1.19 | 58.86±1.53 | 47.64±1.81 | 72.61±0.99 | 61.99±0.78 | 49.10±0.15 |
| | CDR | 67.35±1.70 | 60.05±1.06 | 49.12±0.59 | 72.66±0.79 | 64.58±0.60 | 50.51±2.49 |
| | JOINT | 67.43±0.73 | 63.37±0.92 | 53.27±4.70 | 70.28±1.85 | 68.70±2.88 | 58.57±2.75 |
| | WSIC | 65.43±0.55 | 59.53±0.73 | 48.34±0.47 | 72.57±1.03 | 61.88±2.57 | 50.15±0.86 |
| | CCMN | 69.97±1.36 | 62.58±1.47 | 53.20±1.28 | 70.68±1.08 | 60.94±3.12 | 48.62±1.26 |
| | CCMLN† | **72.07±0.67** | **70.20±0.46** | **68.00±0.89** | **74.83±0.64** | **69.86±1.61** | **60.09±1.73** |
| OF1 ↑ | BCE | 63.52±0.48 | 56.70±2.45 | 48.10±1.43 | 68.28±0.69 | 58.30±2.82 | 51.18±3.10 |
| | CSRA | 65.40±0.47 | 59.39±0.81 | 48.32±1.50 | 69.72±0.50 | 61.89±0.43 | 51.56±2.28 |
| | ADDGCN | 62.63±0.18 | 55.50±1.87 | 44.38±2.92 | 68.95±0.64 | 59.64±0.56 | 53.12±0.62 |
| | APL | 64.85±1.46 | 56.51±1.70 | 47.54±2.40 | 68.89±0.89 | 58.04±0.97 | 52.27±2.20 |
| | CDR | 65.31±0.99 | 57.93±1.05 | 48.86±1.71 | 69.53±0.65 | 59.89±1.07 | 51.68±3.83 |
| | JOINT | 69.72±0.88 | 67.93±0.77 | 61.62±1.40 | 71.24±1.03 | 64.20±0.88 | 60.30±1.24 |
| | WSIC | 63.45±0.97 | 57.96±1.25 | 48.38±2.41 | 69.88±1.22 | 57.97±2.19 | 51.99±1.65 |
| | CCMN | 69.66±1.55 | 60.43±1.31 | 53.84±0.69 | 67.12±0.61 | 59.55±1.45 | 53.46±1.04 |
| | CCMLN† | **71.03±0.33** | **69.08±1.00** | **68.62±0.48** | **72.09±0.74** | **65.76±2.39** | **60.71±1.37** |
| CF1 ↑ | BCE | 58.91±1.34 | 53.21±2.04 | 43.66±0.53 | 65.93±0.81 | 57.03±3.43 | 47.21±1.89 |
| | CSRA | 62.31±0.50 | 55.67±0.61 | 43.11±0.76 | 67.39±0.80 | 59.66±1.04 | 51.13±1.12 |
| | ADDGCN | 60.41±1.04 | 53.72±1.38 | 42.42±0.59 | 66.05±0.97 | 57.81±0.58 | 48.89±2.64 |
| | APL | 60.23±1.53 | 52.85±2.18 | 42.38±1.67 | 66.59±0.71 | 58.33±0.49 | 47.67±1.83 |
| | CDR | 61.37±1.47 | 54.17±0.86 | 43.60±0.82 | 67.11±0.63 | 59.91±0.39 | 48.40±1.98 |
| | JOINT | 63.13±0.38 | 60.22±1.68 | 48.17±5.01 | 66.03±1.25 | 62.05±2.98 | 54.03±3.17 |
| | WSIC | 59.54±1.10 | 54.22±0.53 | 43.82±0.62 | 66.97±1.00 | 58.04±1.70 | 48.19±0.96 |
| | CCMN | 65.19±1.10 | 58.55±1.31 | 49.85±1.06 | 65.47±0.93 | 58.05±2.24 | 48.46±0.80 |
| | CCMLN† | **68.87±0.10** | **66.62±0.81** | **64.82±0.48** | **69.95±1.19** | **65.13±1.04** | **57.54±1.84** |

Table 1: Comparisons with advanced methods on noisy Pascal-VOC 2007. The mean and standard deviation of results (%) are presented.

different experimental settings, e.g., different networks and different image sizes. The details are provided in Section 4.3.

**Measurement.** As did in multi-label classification (Zhu & Wu, 2021; Chen et al., 2019b), evaluation metrics include the mean average precision (mAP) (Zhang & Zhou, 2013), the average F1-measure (OF1), and the average per-class F1-measure (CF1). For fair comparison, we implement all methods with default parameters by PyTorch, and conduct all experiments on NVIDIA GTX3090 GPUs. All experiments are repeated three times with different random seeds. Following the works in learning with noisy labels (Han et al., 2018; Wang et al., 2018; Li et al., 2020; 2021b), the mean and standard deviation of results in the last epoch are reported. In addition, for different evaluation metrics, we report the mean and standard deviation of best results. Supplementary results are shown in Appendix E. Afterwards, the best mean results are highlighted and **bolded**. The second best mean results are also highlighted.

### 4.2 Comparison with the State-of-the-Arts

The results on noisy Pascal-VOC 2007, Pascal-VOC 2012, and MS-COCO are shown in Table 1, Table 2, and Table 3 respectively. In summary, CCMLN consistently works best across all noise settings. In many cases, the best results achieved by CCMLN outperform the second best results by a large margin, especially when the noise level is high. Below, we further discuss the results based on the comparisons with three different types of baselines.

**Compared with Type-I baselines.** We first notice that Type-I baselines are fragile to noisy labels in multi-label classification. Without considering the side-effect of noisy labels, in many cases, they perform worse than BCE, which clearly illustrates the necessity for attention to handling noisy labels. Second, we compare CCMLN with ADDGCN. Without the proposed correction method for combating noisy labels, CCMLN will reduce to ADDGCN. As shown in the reported results, CCMLN performs much better than ADDGCN. To be specific, on noisy Pascal-VOC 2007, for Sym. 40%, CCMLN brings about +15% performance improvement *w.r.t.* three evaluation metrics over ADDGCN. For Sym. 50%, the performance improvement is increased to more than +20%. Also, for Pair. 30% and Pair. 40%, CCMLN enhances ADDGCN with about +10% improvement. On noisy Pascal-VOC 2012 and MS-COCO, the performance improvement is also very clear.

**Compared with Type-II baselines.** On noisy Pascal-VOC 2007, with Sym. noise, we can see that CCMLN outperforms APL, CDR, and JOINT clearly, especially for Sym. 50%. Additionaly,

| Metrics | Methods / Noise | Sym. 30% | Sym. 40% | Sym. 50% | Pair. 20% | Pair. 30% | Pair. 40% |
|---|---|---|---|---|---|---|---|
| mAP ↑ | BCE | 66.74±0.80 | 56.07±0.50 | 45.15±1.56 | 70.91±1.13 | 57.61±1.14 | 49.85±0.36 |
| | CSRA | 66.35±0.50 | 56.20±1.35 | 45.54±1.14 | 71.29±0.83 | 60.71±1.18 | 47.63±1.56 |
| | ADDGCN | 63.34±0.96 | 54.54±0.86 | 44.88±1.71 | 70.41±0.54 | 57.96±0.68 | 47.66±1.08 |
| | APL | 67.07±1.04 | 56.79±1.86 | 43.51±1.93 | 71.32±1.60 | 59.59±1.27 | 48.14±1.16 |
| | CDR | 66.13±1.49 | 56.85±0.48 | 44.84±1.11 | 71.55±1.87 | 60.13±1.89 | 49.44±1.81 |
| | JOINT | 65.19±2.17 | 58.40±2.87 | 45.13±1.69 | 68.93±2.54 | 61.64±1.78 | 53.64±1.61 |
| | WSIC | 65.96±0.79 | 56.34±0.41 | 44.80±0.54 | 70.40±1.11 | 59.40±1.87 | 48.95±1.34 |
| | CCMN | 69.15±0.66 | 61.00±1.01 | 50.71±0.26 | 69.08±1.78 | 59.72±2.32 | 46.67±2.78 |
| | CCMLN† | **72.14±0.66** | **70.11±0.27** | **68.69±1.04** | **74.51±0.67** | **69.90±0.43** | **64.20±1.26** |
| OF1 ↑ | BCE | 64.99±1.10 | 56.92±2.08 | 45.49±2.23 | 68.48±2.28 | 60.21±1.35 | 54.05±1.95 |
| | CSRA | 64.08±0.37 | 56.25±2.57 | 48.67±3.14 | 69.06±0.65 | 59.75±1.70 | 52.89±0.95 |
| | ADDGCN | 63.53±1.41 | 54.28±0.86 | 47.56±2.67 | 47.62±2.39 | 57.90±1.78 | 52.33±0.56 |
| | APL | 64.70±1.17 | 58.05±1.68 | 45.74±1.55 | 70.68±1.03 | 60.22±1.58 | 51.38±1.55 |
| | CDR | 64.06±1.38 | 57.31±1.21 | 46.51±0.95 | 70.45±1.44 | 60.57±1.24 | 52.26±2.42 |
| | JOINT | 67.35±1.86 | 64.57±2.39 | 54.37±3.33 | 70.81±1.40 | 64.40±1.76 | 56.27±1.29 |
| | WSIC | 62.74±2.10 | 57.13±0.73 | 45.52±1.28 | 69.72±1.19 | 59.11±2.04 | 52.49±1.38 |
| | CCMN | 65.77±0.23 | 59.91±0.93 | 51.45±0.94 | 67.93±1.73 | 59.26±0.51 | 48.61±4.71 |
| | CCMLN† | **71.14±0.60** | **69.50±0.40** | **67.80±0.33** | **72.13±0.26** | **67.59±0.96** | **64.28±0.81** |
| CF1 ↑ | BCE | 62.47±0.44 | 53.26±0.41 | 43.43±1.67 | 66.03±1.69 | 55.90±0.70 | 49.29±0.64 |
| | CSRA | 62.08±0.70 | 53.23±1.27 | 43.23±1.25 | 66.02±0.74 | 57.71±1.02 | 47.46±1.68 |
| | ADDGCN | 59.67±1.14 | 52.61±0.52 | 44.33±1.99 | 65.22±0.86 | 55.32±0.76 | 47.30±1.12 |
| | APL | 62.99±1.07 | 53.69±1.80 | 41.72±1.42 | 66.44±1.40 | 57.52±0.91 | 48.14±1.02 |
| | CDR | 62.18±1.04 | 53.61±0.45 | 42.83±0.87 | 66.29±2.12 | 57.23±1.43 | 49.03±1.50 |
| | JOINT | 60.57±2.82 | 54.39±3.72 | 40.48±7.70 | 66.30±2.33 | 59.72±2.12 | 55.06±0.36 |
| | WSIC | 61.70±0.92 | 53.10±0.74 | 42.72±0.54 | 65.34±1.48 | 57.21±1.62 | 48.51±1.12 |
| | CCMN | 64.46±0.62 | 57.45±0.99 | 48.27±0.68 | 67.48±1.44 | 56.93±1.69 | 47.01±1.82 |
| | CCMLN† | **69.54±0.56** | **67.35±0.48** | **65.72±1.48** | **70.07±0.41** | **65.68±0.94** | **60.57±1.27** |

Table 2: Comparisons with advanced methods on noisy Pascal-VOC 2012. The mean and standard deviation of results (%) are presented.

| Metrics | Methods / Noise | Sym. 30% | Sym. 40% | Sym. 50% | Pair. 20% | Pair. 30% | Pair. 40% |
|---|---|---|---|---|---|---|---|
| mAP ↑ | BCE | 53.23±0.15 | 47.33±0.79 | 40.25±0.26 | 56.58±0.22 | 49.16±0.04 | 41.57±0.64 |
| | CSRA | 53.89±0.40 | 47.64±0.86 | 39.58±0.19 | 58.27±0.23 | 50.95±0.07 | 43.07±0.64 |
| | ADDGCN | 51.08±0.95 | 44.75±1.15 | 38.66±1.30 | 56.94±0.61 | 50.28±0.81 | 41.45±0.19 |
| | APL | 54.34±0.32 | 48.61±0.72 | 43.55±1.43 | 57.73±0.20 | 50.87±0.34 | 41.77±0.50 |
| | CDR | 54.01±0.04 | 49.01±0.26 | 43.94±1.25 | 57.03±0.28 | 50.99±0.77 | 42.71±0.09 |
| | JOINT | 53.93±0.41 | 48.01±1.04 | 45.27±0.68 | 57.30±0.33 | 51.94±0.20 | 42.74±0.55 |
| | WSIC | 52.99±0.53 | 46.84±0.86 | 39.76±0.64 | 56.66±0.31 | 49.46±0.25 | 42.52±0.62 |
| | CCMN | 51.73±0.18 | 50.36±0.71 | 45.32±0.89 | 58.13±0.44 | 51.17±0.29 | 42.12±0.76 |
| | CCMLN† | **54.87±0.68** | **51.09±0.53** | **48.15±0.50** | **58.55±0.09** | **53.41±0.13** | **45.91±0.39** |
| OF1 ↑ | BCE | 51.34±1.70 | 44.36±0.82 | 34.85±1.24 | 59.16±0.95 | 52.44±0.81 | 42.94±1.13 |
| | CSRA | 52.03±1.86 | 41.63±1.41 | 33.47±3.18 | 59.17±0.14 | 50.27±0.88 | 41.75±1.36 |
| | ADDGCN | 55.67±1.48 | 47.79±0.40 | 35.95±3.73 | 60.96±0.65 | 55.05±1.78 | 47.47±0.77 |
| | APL | 51.07±1.32 | 43.93±2.70 | 33.90±4.00 | 60.04±1.16 | 50.64±2.86 | 44.34±1.99 |
| | CDR | 53.43±1.16 | 45.10±0.83 | 34.91±0.90 | 59.34±0.61 | 52.72±0.63 | 44.17±0.61 |
| | JOINT | 54.56±0.06 | 49.00±1.66 | 37.78±0.93 | 58.20±0.40 | 53.21±0.17 | 46.55±0.61 |
| | WSIC | 50.91±0.52 | 42.93±0.85 | 35.47±1.52 | 58.89±1.13 | 51.63±1.57 | 43.99±1.47 |
| | CCMN | 52.71±1.04 | 43.24±1.19 | 34.62±1.38 | 58.61±1.18 | 52.18±0.76 | 45.92±0.59 |
| | CCMLN† | **59.92±0.65** | **57.84±0.38** | **55.47±0.95** | **62.28±0.06** | **58.56±0.37** | **51.09±0.60** |
| CF1 ↑ | BCE | 45.92±0.23 | 38.96±1.61 | 31.34±0.27 | 52.54±0.58 | 45.54±0.63 | 39.79±0.99 |
| | CSRA | 44.97±1.88 | 37.49±1.73 | 28.96±1.16 | 52.18±0.44 | 44.96±0.43 | 36.88±0.21 |
| | ADDGCN | 46.77±1.80 | 39.35±1.83 | 30.57±1.57 | 54.18±0.23 | 47.55±0.18 | 39.44±0.33 |
| | APL | 42.91±0.54 | 38.38±0.77 | 28.17±2.50 | 52.87±1.07 | 46.27±1.27 | 37.76±1.02 |
| | CDR | 46.62±0.42 | 39.47±0.54 | 29.59±2.52 | 52.51±0.69 | 45.75±0.81 | 39.15±0.53 |
| | JOINT | 49.51±0.81 | 42.38±1.21 | 24.24±0.61 | 54.39±0.17 | 49.90±0.85 | 38.34±0.55 |
| | WSIC | 45.30±1.09 | 39.15±1.62 | 31.42±0.94 | 52.04±0.28 | 45.76±0.70 | 39.44±1.11 |
| | CCMN | 44.20±1.19 | 35.18±1.01 | 27.90±1.25 | 53.23±0.58 | 46.88±0.92 | 40.55±0.89 |
| | CCMLN† | **51.94±0.63** | **49.24±0.30** | **46.69±0.66** | **55.44±0.13** | **50.91±0.48** | **43.35±0.82** |

Table 3: Comparisons with advanced methods on noisy MS-COCO. The mean and standard deviation of results (%) are presented.

with Pair. noise, although the improvement is less than the cases with Sym. noise, CCMLN still performs best. On noisy Pascal-VOC 2012, for both Sym. and Pair. noise, the improvement is significant. Lastly, for noisy MS-COCO, CCMLN works better than all Type-II baselines with varying enhancement.

Note that, compared with APL and CDR, JOINT seems to be a stronger baseline. Benefiting from label correction, after a few training epochs, JOINT less overfits to wrong labels, following better performance. Nevertheless, the proposed label-correction paradigm is argued to be more advanced. As shown in all results, CCMLN surpasses JOINT, which verifies the effectiveness of our method.

**Compared with Type-III baselines.** On noisy Pascal-VOC 2007 and noisy Pascal-VOC 2012, CCMLN outperforms WSIC and CCMN distinctly. For example, with Sym. 50% noise, more than +10% performance promotion is brought by our method. On noisy MS-COCO, although WSIC and CCMN are sometimes competitive *w.r.t.* mAP, they are inferior *w.r.t.* both OF1 and CF1.

### 4.3 More Analyses and Justifications

In this subsection, we conduct performance analysis in more detail. The experiments are conducted with Sym. 50% noise, which is more challenging than the experiments in low-noise-rate cases.

**Role of label dependence.** We study the effect of removing the consideration of label dependence to provide insights into what makes CCMLN successful. The experiments are conducted on noisy Pascal-VOC 2007, Pascal-VOC 2012, and MS-COCO. The ResNet-50 network pretrained on ImageNet is used as the backbone. The image size is set to $224 \times 224$. Recall that CCMLN considers instance-label and label dependences simultaneously. When we remove the consideration of the label dependence in CCMLN, the corresponding method is named as CCMLN w/o l. here. For both CCMLN w/o l. and CCMLN, the value of the threshold $\hat{\delta}$ is searched in the range $\{0.25, 0.30, 0.35, 0.40, 0.45\}$. We use the 10% noisy training data as a validation set for the threshold determination and performance report. The results are shown in Table 4. As can be seen, CCMLN outperforms CCMLN w/o l.. The results justify our claims that the label dependence could help combat the noisy labels in multi-label classification, which demonstrate the effectiveness of the proposed combinatorial correction.

| Dataset | Noisy Pascal-VOC 2007 | | |
|---|---|---|---|
| Methods | mAP ↑ | OF1 ↑ | CF1 ↑ |
| CCMLN w/o l. | 67.06±0.41 | 67.23±1.92 | 63.42±0.58 |
| CCMLN | 68.00±0.89 (+0.94) | 68.62±0.48 (+1.39) | 64.82±0.48 (+1.40) |
| Dataset | Noisy Pascal-VOC 2012 | | |
| Methods | mAP ↑ | OF1 ↑ | CF1 ↑ |
| CCMLN w/o l. | 67.88±0.75 | 66.30±1.28 | 64.33±1.67 |
| CCMLN | 68.69±1.04 (+0.81) | 67.80±0.33 (+1.50) | 65.72±1.48 (+1.39) |
| Dataset | Noisy MS-COCO | | |
| Methods | mAP ↑ | OF1 ↑ | CF1 ↑ |
| CCMLN w/o l. | 46.21±0.36 | 52.90±0.92 | 44.51±1.29 |
| CCMLN | 48.15±0.50 (+1.94) | 55.47±0.95 (+2.57) | 46.69±0.66 (+2.18) |

Table 4: Ablation study results on noisy Pascal-VOC 2007, Pascal-VOC 2012, and MS-COCO. The mean and standard deviation of results are presented. The performance improvement is highlighted.

**Analysis of the threshold $\hat{\delta}$.** We analyze the influence of different values of the threshold $\hat{\delta}$. The experiments are conducted on noisy Pascal-VOC 2007 and Pascal-VOC 2012. The ResNet-50 network pretrained on ImageNet is used as the backbone. The image size is set to $224 \times 224$. The value of the threshold $\hat{\delta}$ is chosen in $\{0.25, 0.30, 0.35, 0.40, 0.45\}$. Figure 2 shows that CCMLN is robust to the determination of the threshold $\hat{\delta}$ in the certain range, which facilitates the practical application of our method.

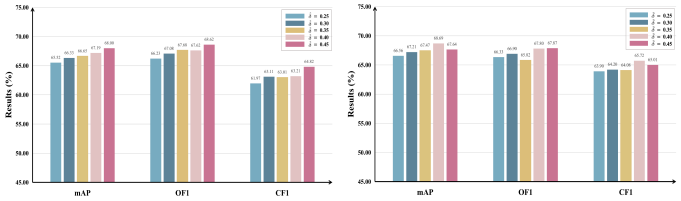


Figure 2: Ablation study results with different values of the set threshold $\hat{\delta}$. The experiments are conducted on noisy Pascal-VOC 2007 (**Left**) and noisy Pascal-VOC 2012 (**Right**).

**Evaluations with different networks.** We exploit pretrained ResNet-50 before. To show that our method is robust to the choice of network structures, we use different networks in experiments. Specifically, we employ pretrained ResNet-34 (He et al., 2016) and pretrained ResNet-101 (He et al., 2016) respectively. The noisy MS-COCO is considered. The image size is $224 \times 224$. The experimental results on mAP are reported in Table 5. As can be seen, with different networks, CCMLN still works well.

**Evaluations with different image sizes.** We resize the image size to $224 \times 224$ before. To test the performance of advanced methods with different image sizes, we further consider $112 \times 112$ $384 \times 384$, and $448 \times 448$ image sizes. Pretrained ResNet-50 is used. The results are reported in Table 6. For mAP, we can see that CCMLN is competitive compared with CCMN and CSRA. For OF1 and CF1, CCMLN works better than all baselines with a clear margin.

| Metrics | Methods/ Networks | ResNet-34 | ResNet-101 |
|---|---|---|---|
| mAP ↑ | BCE | 42.63±0.74 | 38.17±0.41 |
| | CSRA | 41.35±0.18 | 37.24±1.20 |
| | ADDGCN | 40.15±0.98 | 36.13±0.69 |
| | APL | 44.82±0.70 | 40.90±1.51 |
| | CDR | 45.43±0.65 | 41.00±0.38 |
| | JOINT | 44.81±0.77 | 39.96±1.30 |
| | WSIC | 41.86±0.62 | 37.49±0.68 |
| | CCMN | 45.31±0.47 | 46.01±1.01 |
| | CCMLN† | **46.05±0.81** | **46.24±2.13** |
| OF1 ↑ | BCE | 37.65±2.46 | 38.65±2.50 |
| | CSRA | 35.05±0.78 | 34.28±2.40 |
| | ADDGCN | 35.11±1.26 | 37.18±0.50 |
| | APL | 34.31±1.73 | 37.89±1.30 |
| | CDR | 36.67±3.43 | 39.88±1.15 |
| | JOINT | 39.66±1.13 | 41.36±0.88 |
| | WSIC | 35.08±1.74 | 38.44±0.57 |
| | CCMN | 32.86±1.50 | 37.03±1.48 |
| | CCMLN† | **44.11±0.80** | **47.79±4.19** |
| CF1 ↑ | BCE | 28.95±2.09 | 34.11±1.13 |
| | CSRA | 27.40±0.44 | 31.21±1.57 |
| | ADDGCN | 26.11±0.86 | 30.41±0.72 |
| | APL | 26.64±2.39 | 31.97±1.95 |
| | CDR | 27.13±2.39 | 35.17±1.06 |
| | JOINT | 30.77±1.63 | 37.63±0.81 |
| | WSIC | 27.62±0.65 | 33.83±0.61 |
| | CCMN | 24.75±0.48 | 26.65±0.26 |
| | CCMLN† | **34.79±1.41** | **40.88±2.42** |

Table 5: Comparisons with advanced methods on noisy MS COCO with different networks. The mean and standard deviation of results (%) are presented.

| Metrics | Methods/ Image sizes | 112 × 112 | 384 × 384 | 448 × 448 |
|---|---|---|---|---|
| mAP ↑ | BCE | 32.22±0.69 | 39.40±1.36 | 35.24±1.73 |
| | CSRA | 29.55±0.16 | 43.55±0.70 | **44.56±0.75** |
| | ADDGCN | 32.34±0.46 | 38.72±1.64 | 34.87±1.89 |
| | APL | 34.41±0.48 | 43.65±0.28 | 41.44±1.21 |
| | CDR | 34.75±0.39 | 43.26±0.72 | 39.97±1.40 |
| | JOINT | 32.89±0.16 | 42.95±0.88 | 40.17±1.26 |
| | WSIC | 31.98±0.23 | 39.57±1.02 | 36.08±0.23 |
| | CCMN | **36.17±0.41** | 44.39±0.39 | 44.03±0.17 |
| | CCMLN† | 35.98±1.05 | **45.12±0.13** | 44.23±1.20 |
| OF1 ↑ | BCE | 26.70±0.88 | 34.71±2.76 | 26.72±3.35 |
| | CSRA | 20.14±1.28 | 36.41±0.71 | 38.54±0.78 |
| | ADDGCN | 26.36±2.29 | 42.83±2.05 | 40.73±1.04 |
| | APL | 24.02±1.22 | 34.68±1.46 | 30.73±2.64 |
| | CDR | 26.50±1.23 | 31.31±1.76 | 31.15±3.46 |
| | JOINT | 34.11±0.95 | 38.67±1.25 | 38.11±0.69 |
| | WSIC | 24.61±1.10 | 34.09±2.94 | 30.69±0.97 |
| | CCMN | 23.89±1.49 | 36.16±2.12 | 25.03±2.48 |
| | CCMLN† | **39.05±2.68** | **46.55±3.77** | **45.14±2.34** |
| CF1 ↑ | BCE | 19.61±0.61 | 30.77±2.28 | 24.94±3.31 |
| | CSRA | 13.34±1.29 | 32.66±0.98 | 32.68±0.22 |
| | ADDGCN | 18.67±1.47 | 35.63±1.37 | 33.89±2.41 |
| | APL | 17.21±0.78 | 29.24±0.09 | 25.01±1.31 |
| | CDR | 18.04±1.07 | 29.62±1.19 | 26.28±1.23 |
| | JOINT | 20.76±0.75 | 34.90±1.88 | 33.75±1.31 |
| | WSIC | 19.07±0.57 | 31.63±2.49 | 28.50±1.36 |
| | CCMN | 16.75±1.21 | 30.09±2.32 | 26.68±1.57 |
| | CCMLN† | **29.70±2.07** | **40.34±2.14** | **37.48±2.36** |

Table 6: Comparisons with advanced methods on noisy MS COCO. The mean and standard deviation of results (%) are presented. Difference image sizes are considered here.

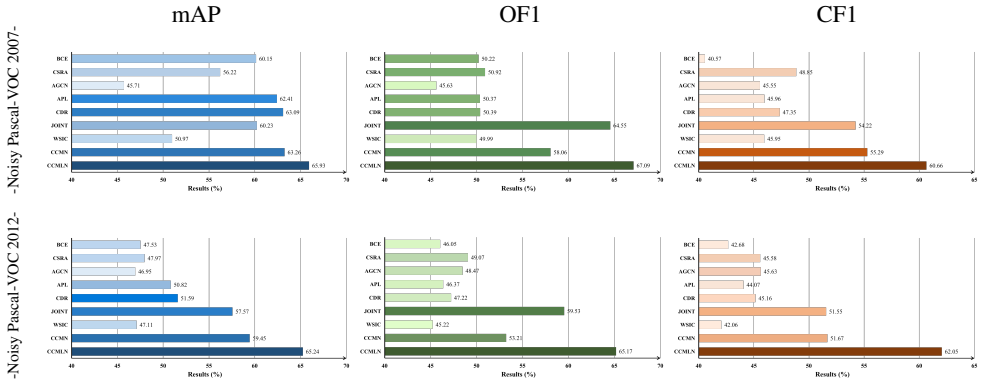


Figure 3: Ablation study results with different optimizations.

**Evaluations with different optimizations.** We use the Adam optimizer (Kingma & Ba, 2014) before. Here, the optimizer is changed to RAdam (Liu et al., 2020a). The experiments are conducted on noisy Pascal-VOC 2007 and Pascal-VOC 2012. The ResNet-50 network pretrained on ImageNet is used as the backbone. As shown in Figure 3, with RAdam, CCMLN still outperforms all baselines clearly.

## 5 Conclusion

In this paper, we focus on the realistic problem of multi-label classification with noisy labels. We learn and utilize the label dependence among multiple labels to handle this problem. With the help of the label dependence, a novel algorithm named CCMLN is proposed to correct noisy multiple labels to clean ones. We demonstrate the effectiveness of our algorithm both theoretically and empirically. For future work, we are interested in adapting CCMLN to other domains such as natural language processing and recommendation systems. We are also interested in promoting our algorithm to tackle instance-dependent label noise (Zhang et al., 2021c; Berthon et al., 2020; Zhu et al., 2021; Liu, 2022) in multi-label classification.

## REFERENCES


Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

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*, pp. 233–242, 2017.

Dara Bahri, Heinrich Jiang, and Maya Gupta. Deep k-nn for noisy labels. In *ICML*, pp. 540–550, 2020.

Yingbin Bai and Tongliang Liu. Me-momentum: Extracting hard confident examples from noisily labeled data. In *ICCV*, pp. 9312–9321, 2021.

Antonin Berthon, Bo Han, Gang Niu, Tongliang Liu, and Masashi Sugiyama. Confidence scores make instance-dependent label-noise learning possible. *arXiv preprint arXiv:2001.03772*, 2020.

Tianshui Chen, Muxin Xu, Xiaolu Hui, Hefeng Wu, and Liang Lin. Learning semantic-specific graph representation for multi-label image recognition. In *ICCV*, pp. 522–531, 2019a.

Tianshui Chen, Liang Lin, Xiaolu Hui, Riquan Chen, and Hefeng Wu. Knowledge-guided multi-label few-shot learning for general image recognition. *Transactions on Pattern Analysis and Machine Intelligence*, 2020.

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

Zhaomin Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Learning graph convolutional networks for multi-label recognition and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Xiang Cheng, Shu-Guang Zhao, Xuan Xiao, and Kuo-Chen Chou. iatc-misf: a multi-label classifier for predicting the classes of anatomical therapeutic chemicals. *Bioinformatics*, 33(3):341–346, 2017.

Tejas Chheda, Purujit Goyal, Trang Tran, Dhruvesh Patel, Michael Boratko, Shib Sankar Dasgupta, and Andrew McCallum. Box embeddings: An open-source library for representation learning using geometric structures. *arXiv preprint arXiv:2109.04997*, 2021.

Zijun Cui, Yong Zhang, and Qiang Ji. Label error correction and generation through label relationships. In *AAAI*, pp. 3693–3700, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Mark Everingham, Andrew Zisserman, Christopher KI Williams, Luc Van Gool, Moray Allan, Christopher M Bishop, Olivier Chapelle, Navneet Dalal, Thomas Deselaers, Gyuri Dorkó, et al. The pascal visual object classes challenge 2007 (voc2007) results. 2008.

Lei Feng, Senlin Shu, Zhuoyi Lin, Fengmao Lv, Li Li, and Bo An. Can cross entropy loss be robust to label noise? In *IJCAI*, pp. 2206–2212, 2021.

Bin-Bin Gao and Hong-Yu Zhou. Learning to discover multi-class attentional regions for multi-label image recognition. *IEEE Transactions on Image Processing*, 30:5920–5932, 2021.

Wei Gao, Bin-Bin Yang, and Zhi-Hua Zhou. On the resistance of nearest neighbor to random noisy labels. *arXiv preprint arXiv:1607.07526*, 2016.

Hao Guo, Kang Zheng, Xiaochuan Fan, Hongkai Yu, and Song Wang. Visual attention consistency under image transforms for multi-label image classification. In *CVPR*, pp. 729–739, 2019.

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*, pp. 8527–8537, 2018.

Bo Han, Gang Niu, Xingrui Yu, Quanming Yao, Miao Xu, Ivor Tsang, and Masashi Sugiyama. Sigua: Forgetting may make learning with noisy labels more robust. In *ICML*, pp. 4006–4016, 2020.

Jun-Yi Hang and Min-Ling Zhang. Collaborative learning of label semantics and deep label-specific features for multi-label classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

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

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *NeurIPS*, 2018.

Mengying Hu, Hu Han, Shiguang Shan, and Xilin Chen. Weakly supervised image classification through noise regularization. In *CVPR*, pp. 11517–11525, 2019.

Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *ICLR*, 2020.

Jinchi Huang, Lie Qu, Rongfei Jia, and Binqiang Zhao. O2u-net: A simple noisy label detection approach for deep neural networks. In *ICCV*, pp. 3326–3334, 2019.

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*, pp. 2309–2318, 2018.

Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. Beyond synthetic noise: Deep learning on controlled noisy labels. In *ICML*, pp. 4804–4815, 2020.

Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. Nlnl: Negative learning for noisy labels. In *ICCV*, pp. 101–110, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kaustav Kundu and Joseph Tighe. Exploiting weakly supervised visual patterns to learn from partial annotations. In *NeurIPS*, pp. 561–572, 2020.

Cheng Li, Bingyu Wang, Virgil Pavlu, and Javed Aslam. Conditional bernoulli mixtures for multi-label classification. In *ICML*, pp. 2482–2491, 2016.

Jingling Li, Mozhi Zhang, Keyulu Xu, John Dickerson, and Jimmy Ba. How does a neural network's architecture impact its robustness to noisy labels? In *NeurIPS*, 2021a.

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

Junnan Li, Caiming Xiong, and Steven CH Hoi. Learning from noisy data with robust representation learning. In *ICCV*, pp. 9485–9494, 2021b.

Shikun Li, Xiaobo Xia, Hansong Zhang, Yibing Zhan, Shiming Ge, and Tongliang Liu. Estimating noise transition matrix with label correlations for noisy multi-label learning. In *NeurIPS*, 2022.

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*, pp. 740–755, 2014.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *ICLR*, 2020a.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *NeurIPS*, 2020b.

Weiwei Liu, Ivor W Tsang, and Klaus-Robert Müller. An easy-to-hard learning paradigm for multiple classes and multiple labels. *Journal of Machine Learning Research*, 18, 2017.

Weiwei Liu, Donna Xu, Ivor W Tsang, and Wenjie Zhang. Metric learning for multi-output tasks. *Transactions on Pattern Analysis and Machine Intelligence*, 41(2):408–422, 2018.

Weiwei Liu, Haobo Wang, Xiaobo Shen, and Ivor Tsang. The emerging trends of multi-label learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Yang Liu. Identifiability of label noise transition matrix. *arXiv preprint arXiv:2202.02016*, 2022.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *ICML*, pp. 6448–6458, 2020.

Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah M Erfani, Shu-Tao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *ICML*, pp. 3361–3370, 2018.

Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *ICML*, pp. 6543–6553, 2020.

Aditya Krishna Menon, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Can gradient clipping mitigate label noise? In *ICLR*, 2019.

Baharan Mirzasoleiman, Kaidi Cao, and Jure Leskovec. Coresets for robust training of neural networks against noisy labels. In *NeurIPS*, 2020.

Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. Self: Learning to filter noisy labels with self-ensembling. In *ICLR*, 2020.

Curtis G Northcutt, Tailin Wu, and Isaac L Chuang. Learning with confident examples: Rank pruning for robust classification with noisy labels. In *UAI*, 2017.

Yasumasa Onoe, Michael Boratko, Andrew McCallum, and Greg Durrett. Modeling fine-grained entity types with box embeddings. In *ACL*, 2021.

Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Multi-objective interpolation training for robustness to label noise. In *CVPR*, pp. 6606–6615, 2021.

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*, pp. 1944–1952, 2017.

Cosmin Octavian Pene, Amirmasoud Ghiassi, Taraneh Younesian, Robert Birke, and Lydia Y Chen. Multi-label gold asymmetric loss correction with single-label regulators. *arXiv preprint arXiv:2108.02032*, 2021.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pp. 1532–1543, 2014.

Geoff Pleiss, Tianyi Zhang, Ethan R Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. In *NeurIPS*, 2020.

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*, 2021.

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

Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *ICML*, pp. 5907–5915, 2019.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, 2018.

Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In *NeurIPS*, pp. 5596–5605, 2017.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, 2017.

Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *CVPR*, pp. 2285–2294, 2016.

Xinshao Wang, Yang Hua, Elyor Kodirov, David A Clifton, and Neil M Robertson. Proselflc: Progressive self label correction for training robust deep neural networks. In *CVPR*, pp. 752–761, 2021.

Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. In *CVPR*, pp. 8688–8696, 2018.

Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, pp. 13726–13735, 2020.

Yunchao Wei, Wei Xia, Min Lin, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. Hcp: A flexible cnn framework for multi-label image classification. *Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1901–1907, 2015.

Baoyuan Wu, Siwei Lyu, and Bernard Ghanem. Ml-mg: Multi-label learning with missing labels using a mixed graph. In *ICCV*, pp. 4157–4165, 2015.

Pengxiang Wu, Songzhu Zheng, Mayank Goswami, Dimitris Metaxas, and Chao Chen. A topological filter for learning with label noise. In *NeurIPS*, 2020.

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*, pp. 6835–6846, 2019.

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.

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

Ming-Kun Xie and Sheng-Jun Huang. Multi-label learning with pairwise relevance ordering. In *NeurIPS*, 2021a.

Ming-Kun Xie and Sheng-Jun Huang. Partial multi-label learning with noisy label identification. *Transactions on Pattern Analysis and Machine Intelligence*, 2021b.

Ming-Kun Xie and Sheng-Jun Huang. Ccmn: A general framework for learning with class-conditional multi-label noise. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

Quanming Yao, Hansi Yang, Bo Han, Gang Niu, and James Tin-Yau Kwok. Searching to exploit memorization effect in learning with noisy labels. In *ICML*, pp. 10789–10798, 2020.

Yazhou Yao, Zeren Sun, Chuanyi Zhang, Fumin Shen, Qi Wu, Jian Zhang, and Zhenmin Tang. Jo-src: A contrastive approach for combating noisy labels. In *CVPR*, pp. 5192–5201, 2021.

Vacit Oguz Yazici, Abel Gonzalez-Garcia, Arnau Ramisa, Bartlomiej Twardowski, and Joost van de Weijer. Orderless recurrent models for multi-label classification. In *CVPR*, 2020.

Jin Ye, Junjun He, Xiaojiang Peng, Wenhao Wu, and Yu Qiao. Attention-driven dynamic graph convolutional network for multi-label image recognition. In *ECCV*, pp. 649–665, 2020.

Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit Dhillon. Large-scale multi-label learning with missing labels. In *ICML*, pp. 593–601, 2014.

Ze-Bang Yu and Min-Ling Zhang. Multi-label classification with label-specific feature generation: A wrapped approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Jiong Zhang, Wei-cheng Chang, Hsiang-fu Yu, and Inderjit Dhillon. Fast multi-resolution transformer fine-tuning for extreme multi-label text classification. In *NeurIPS*, 2021a.

Min-Ling Zhang and Lei Wu. Lift: Multi-label learning with label-specific features. *Transactions on Pattern Analysis and Machine Intelligence*, 37(1):107–120, 2014.

Min-Ling Zhang and Kun Zhang. Multi-label learning by exploiting label dependency. In *KDD*, pp. 999–1008, 2010.

Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2013.

Mingyuan Zhang, Jane Lee, and Shivani Agarwal. Learning from noisy labels with no change to the training process. In *ICML*, pp. 12468–12478, 2021b.

Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. Learning with feature-dependent label noise: A progressive approach. In *ICLR*, 2021c.

Yivan Zhang, Gang Niu, and Masashi Sugiyama. Learning noise transition matrix from only noisy labels via total variation regularization. In *ICML*, 2021d.

Jiawei Zhao, Ke Yan, Yifan Zhao, Xiaowei Guo, Feiyue Huang, and Jia Li. Transformer-based dual relation graph for multi-label image recognition. In *ICCV*, pp. 163–172, 2021.

Wenting Zhao and Carla Gomes. Evaluating multi-label classifiers with noisy labels. *arXiv preprint arXiv:2102.08427*, 2021.

Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen. Error-bounded correction of noisy labels. In *ICML*, pp. 11447–11457, 2020.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pp. 2921–2929, 2016.

Xiong Zhou, Xianming Liu, Junjun Jiang, Xin Gao, and Xiangyang Ji. Asymmetric loss functions for learning with noisy labels. In *ICML*, pp. 12846–12856, 2021.

Feng Zhu, Hongsheng Li, Wanli Ouyang, Nenghai Yu, and Xiaogang Wang. Learning spatial regularization with image-level supervisions for multi-label image classification. In *CVPR*, pp. 5513–5522, 2017.

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

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

# Appendix of "CCMLN: Combinatorial Correction for Multi-Label Classification with Noisy Labels"

## A Supplementary Theoretical Results

**Lemma 1** *Suppose $S^f(\boldsymbol{x})$ fulfills the Tsybakov condition on instance-label dependence for constants $C_1$, $\lambda_1 > 0$, and $t_0 \in (0, m]$. Let $\kappa^f(h, \boldsymbol{x}, \bar{\boldsymbol{y}}) := \hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) / \hat{\bar{S}}^f_{\boldsymbol{y}^*}(\boldsymbol{x})$. We define $\epsilon := \max_{\boldsymbol{x},\boldsymbol{z}} \left[ |\hat{\bar{S}}^f_{\boldsymbol{z}}(\boldsymbol{x}) - \bar{S}^f_{\boldsymbol{z}}(\boldsymbol{x})|, |\hat{\bar{S}}^l_{\boldsymbol{z}}(\boldsymbol{x}) - \bar{S}^l_{\boldsymbol{z}}(\boldsymbol{x})|, |\bar{S}^l_{\boldsymbol{z}}(\boldsymbol{x}) - S^l_{\boldsymbol{z}}(\boldsymbol{x})| \right]$ and $\tau := \min_i T_{ii}$. We analyze two cases:*
*(1) If $\bar{\boldsymbol{y}}$ is corrected by $\kappa^f(h, \boldsymbol{x}, \bar{\boldsymbol{y}})$ with the threshold $\hat{\delta}$, let $\delta_1 = \min \left[ \frac{\tau S^f_{\boldsymbol{b}_{\boldsymbol{x}}} + \sum_{l_j \in \bar{\boldsymbol{y}}} \sum_{i \neq j} T_{ij} \mathbb{P}(l_i | \boldsymbol{x})}{\hat{\bar{S}}^f_{\boldsymbol{y}^*}} \right]$ and $\rho_1 := |\hat{\delta} - \delta_1|$. Assume that $\epsilon \leq t_0 \tau - \rho_1 m$. Then, $\mathbb{P}[\bar{\boldsymbol{y}}_{new} = h^*(\boldsymbol{x}), \bar{\boldsymbol{y}}\ is\ flipped]$ is at least $1 - C_1[\max(\epsilon, \rho_1)]^{\lambda_1} - \mathbb{P}[\boldsymbol{a}_{\boldsymbol{x}} \neq \{\boldsymbol{y}^*, \bar{\boldsymbol{y}}\}]$.*
*(2) If $\bar{\boldsymbol{y}}$ is not corrected by $\kappa^f(h, \boldsymbol{x}, \bar{\boldsymbol{y}})$ with the threshold $\hat{\delta}$, let $\delta_2 = \max \left[ \frac{\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\tau S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) + \sum_{l_j \in \boldsymbol{y}^*} \sum_{i \neq j} T_{ij} \mathbb{P}(l_i | \boldsymbol{x})} \right]$ and $\rho_2 := |\hat{\delta} - \delta_2|$. Assume that $\epsilon \leq \frac{t_0 \delta_2^2 \tau - \rho_2 m - \rho_2^2 m}{\delta_2^2}$. Then, $\mathbb{P}[\bar{\boldsymbol{y}}_{new} = h^*(\boldsymbol{x}), \bar{\boldsymbol{y}}\ is\ accepted]$ is at least $1 - C_1[\max(\epsilon, \rho_2)]^{\lambda_1} - \mathbb{P}[\boldsymbol{a}_{\boldsymbol{x}} \neq \{\boldsymbol{y}^*, \bar{\boldsymbol{y}}\}]$.*

Lemma 1 claims that, even though with noisy multiple labels, there is a guaranteed success rate to make proper label corrections by instance-label dependency.

## B Proofs of Theoretical Results

### B.1 Proof of Theorem 1

**Proof 1** *For the case (1),*

$$\mathbb{P}[\bar{\boldsymbol{y}}_{new} \neq h^*(\boldsymbol{x}), \bar{\boldsymbol{y}}\ is\ flipped] = \mathbb{P}\left[ \bar{\boldsymbol{y}}_{new} \neq h^*(\boldsymbol{x}), \frac{\hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\hat{\bar{S}}_{\boldsymbol{y}^*}(\boldsymbol{x})} < \hat{\delta} \right] \tag{5}$$

$$\leq \mathbb{P}\left[ h^*(\boldsymbol{x}) = \bar{\boldsymbol{y}}, \frac{\hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\hat{\bar{S}}_{\boldsymbol{y}^*}(\boldsymbol{x})} < \hat{\delta} \right] + \mathbb{P}[\boldsymbol{a}_{\boldsymbol{x}} \neq \{\boldsymbol{y}^*, \bar{\boldsymbol{y}}\}]. \tag{6}$$

*For the first term,*

$$\mathbb{P}\left[ h^*(\boldsymbol{x}) = \bar{\boldsymbol{y}}, \frac{\hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\hat{\bar{S}}_{\boldsymbol{y}^*}(\boldsymbol{x})} < \hat{\delta} \right] = \mathbb{P}\left[ S^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) > S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), \hat{\bar{S}}_{\bar{\boldsymbol{y}}} < \hat{\delta} \hat{\bar{S}}_{\boldsymbol{y}^*}(\boldsymbol{x}) \right] \tag{7}$$

$$= \mathbb{P}\left[ S_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) - S^l_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) > S_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) - S^l_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), \hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) + \hat{\bar{S}}^l_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) < \hat{\delta} \hat{\bar{S}}_{\boldsymbol{y}^*}(\boldsymbol{x}) \right] \tag{8}$$

$$\leq \mathbb{P}\left[ S_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) \geq S_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), \bar{S}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) + \hat{\bar{S}}^l_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) < \hat{\delta} \hat{\bar{S}}_{\boldsymbol{y}^*}(\boldsymbol{x}) + \epsilon \right] \tag{9}$$

$$\leq \mathbb{P}\left[ S_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) \geq S_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), \bar{S}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) + S^l_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) < \hat{\delta} \hat{\bar{S}}_{\boldsymbol{y}^*}(\boldsymbol{x}) + 3\epsilon \right] \tag{10}$$

$$\leq \mathbb{P}\left[ S_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) \geq S_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), S_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) < \frac{\delta \hat{\bar{S}}_{\boldsymbol{y}^*}(\boldsymbol{x}) - \sum_{l_j \in \bar{\boldsymbol{y}}} \sum_{i \neq j} T_{ij} \mathbb{P}(l_i | \boldsymbol{x})}{\tau} + \frac{3\epsilon + m\rho_1}{\tau} \right]. \tag{11}$$

*If $\delta = \min \left[ \frac{\tau S_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) + \sum_{l_j \in \bar{\boldsymbol{y}}} \sum_{i \neq j} T_{ij} \mathbb{P}(l_i | \boldsymbol{x})}{\hat{\bar{S}}_{\boldsymbol{y}^*}(\boldsymbol{x})} \right]$, we have $\mathbb{P}\left[ h^*(\boldsymbol{x}) = \bar{\boldsymbol{y}}, \frac{\hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\hat{\bar{S}}_{\boldsymbol{y}^*}(\boldsymbol{x})} < \hat{\delta} \right] \leq C_2[O(\max(\epsilon, \rho_1)]^{\lambda_2}$. Therefore, for the case (1),*

$$\mathbb{P}[\bar{\boldsymbol{y}}_{new} = h^*(\boldsymbol{x}), \bar{\boldsymbol{y}}\ is\ flipped] \geq 1 - C_2[O(\max(\epsilon, \rho_1))]^{\lambda_2} - \mathbb{P}[\boldsymbol{a}_{\boldsymbol{x}} \neq \{\boldsymbol{y}^*, \bar{\boldsymbol{y}}\}]. \tag{12}$$

*For the case (2),*

$$\mathbb{P}[\bar{\boldsymbol{y}}_{new} \neq h^*(\boldsymbol{x}), \bar{\boldsymbol{y}} \; is \; accepted] = \mathbb{P}\left[\bar{\boldsymbol{y}}_{new} \neq h^*(\boldsymbol{x}), \frac{\hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\hat{\bar{S}}_{\boldsymbol{y}^*}(\boldsymbol{x})} \geq \hat{\delta}\right] \tag{13}$$

$$\leq \mathbb{P}\left[S^f_{\boldsymbol{y}^*}(\boldsymbol{x}) > S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), \hat{\bar{S}}_{\boldsymbol{y}^*} \leq \hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x})/\hat{\delta}\right] + \mathbb{P}[\boldsymbol{a}_{\boldsymbol{x}} \neq \{\boldsymbol{y}^*, \bar{\boldsymbol{y}}\}]. \tag{14}$$

*For the first term,*

$$\mathbb{P}\left[S^f_{\boldsymbol{y}^*}(\boldsymbol{x}) > S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), \hat{\bar{S}}_{\boldsymbol{y}^*} \leq \hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x})/\hat{\delta}\right] \tag{15}$$

$$\leq \mathbb{P}\left[S_{\boldsymbol{y}^*}(\boldsymbol{x}) - S^l_{\boldsymbol{y}^*}(\boldsymbol{x}) > S_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) - S^l_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), \bar{S}^f_{\boldsymbol{y}^*}(\boldsymbol{x}) + \hat{\bar{S}}^l_{\boldsymbol{y}^*}(\boldsymbol{x}) \leq \hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x})/\hat{\delta} + \epsilon\right] \tag{16}$$

$$\leq \mathbb{P}\left[S_{\boldsymbol{y}^*}(\boldsymbol{x}) \geq S_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), S_{\boldsymbol{y}^*}(\boldsymbol{x}) \leq \frac{\hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x})/\hat{\delta} - \sum_{l_j \in \boldsymbol{y}^*} \sum_{i \neq j} T_{ij} \mathbb{P}(l_i|\boldsymbol{x})}{\tau} + \frac{3\epsilon}{\tau}\right] \tag{17}$$

$$\leq \mathbb{P}\left[S_{\boldsymbol{y}^*}(\boldsymbol{x}) \geq S_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), S_{\boldsymbol{y}^*}(\boldsymbol{x}) \leq \frac{\hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x})/\delta - \sum_{l_j \in \boldsymbol{y}^*} \sum_{i \neq j} T_{ij} \mathbb{P}(l_i|\boldsymbol{x})}{\tau} + \frac{3\epsilon}{\tau} + \frac{\frac{\rho_2 \hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\delta(\delta-\rho_2)}}{\tau}\right]. \tag{18}$$

*If* $\delta = \max\left[\frac{\hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\tau S_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) + \sum_{l_j \in \boldsymbol{y}^*} \sum_{i \neq j} T_{ij} \mathbb{P}(l_i|\boldsymbol{x})}\right]$*, we have*

$$\mathbb{P}\left[S^f_{\boldsymbol{y}^*}(\boldsymbol{x}) > S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), \hat{\bar{S}}_{\boldsymbol{y}^*} \leq \hat{\bar{S}}_{\bar{\boldsymbol{y}}}(\boldsymbol{x})/\hat{\delta}\right] \tag{19}$$

$$\leq \mathbb{P}\left[S_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) \leq S_{\boldsymbol{y}^*}(\boldsymbol{x}) \leq S_{\boldsymbol{b}_{\boldsymbol{x}}} + \frac{3\epsilon}{\tau} + \frac{\rho_2 m}{\delta^2 \tau} + \frac{\rho_2^2 m}{\delta^2 \tau}\right] \leq C_2[O(\max(\epsilon, \rho_2))]^{\lambda_2}. \tag{20}$$

*Therefore, we have*

$$\mathbb{P}[\bar{\boldsymbol{y}}_{new} = h^*(\boldsymbol{x}), \bar{\boldsymbol{y}} \; is \; accepted] \geq 1 - C_2[O(\max(\epsilon, \rho_2))]^{\lambda_2} - \mathbb{P}[\boldsymbol{a}_{\boldsymbol{x}} \neq \{\boldsymbol{y}^*, \bar{\boldsymbol{y}}\}]. \tag{21}$$

### B.2 Proofs of Lemma 1 and Corollary 1

We first prove Lemma 1. Lemma 1 uses the similar proof skill of Theorem 3 of (Zheng et al., 2020). We extend it into multi-label classification.

**Proof 2** *For the case (1),*

$$\mathbb{P}[\bar{\boldsymbol{y}}_{new} \neq h^*(\boldsymbol{x}), \bar{\boldsymbol{y}} \; is \; flipped] = \mathbb{P}\left[\bar{\boldsymbol{y}}_{new} \neq h^*(\boldsymbol{x}), \frac{\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\hat{\bar{S}}^f_{\boldsymbol{y}^*}(\boldsymbol{x})} < \hat{\delta}\right] \tag{22}$$

$$= \mathbb{P}\left[\bar{\boldsymbol{y}}_{new} = \boldsymbol{y}^* \neq h^*(\boldsymbol{x}) = \bar{\boldsymbol{y}}, \frac{\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\hat{\bar{S}}^f_{\boldsymbol{y}^*}(\boldsymbol{x})} < \hat{\delta}\right] + \mathbb{P}\left[\bar{\boldsymbol{y}}_{new} = \boldsymbol{y}^* \neq h^*(\boldsymbol{x}) = \boldsymbol{a}_{\boldsymbol{x}} \neq \bar{\boldsymbol{y}}, \frac{\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\hat{\bar{S}}^f_{\boldsymbol{y}^*}(\boldsymbol{x})} < \hat{\delta}\right] \tag{23}$$

$$\leq \mathbb{P}\left[h^*(\boldsymbol{x}) = \bar{\boldsymbol{y}}, \frac{\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\hat{\bar{S}}^f_{\boldsymbol{y}^*}(\boldsymbol{x})} < \hat{\delta}\right] + \mathbb{P}\left[\bar{\boldsymbol{y}}_{new} = \boldsymbol{y}^* \neq h^*(\boldsymbol{x}) = \boldsymbol{a}_{\boldsymbol{x}} \neq \bar{\boldsymbol{y}}, \frac{\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\hat{\bar{S}}^f_{\boldsymbol{y}^*}(\boldsymbol{x})} < \hat{\delta}\right] \tag{24}$$

$$\leq \mathbb{P}\left[h^*(\boldsymbol{x}) = \bar{\boldsymbol{y}}, \frac{\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\hat{\bar{S}}^f_{\boldsymbol{y}^*}(\boldsymbol{x})} < \hat{\delta}\right] + \mathbb{P}[\boldsymbol{a}_{\boldsymbol{x}} \neq \{\boldsymbol{y}^*, \bar{\boldsymbol{y}}\}]. \tag{25}$$

*For the first term, we have*

$$\mathbb{P}\left[h^*(\boldsymbol{x}) = \bar{\boldsymbol{y}}, \frac{\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\hat{\bar{S}}^f_{\boldsymbol{y}^*}(\boldsymbol{x})} < \hat{\delta}\right] = \mathbb{P}\left[S^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) > S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), \hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) < \hat{\delta}\hat{\bar{S}}^f_{\boldsymbol{y}^*}(\boldsymbol{x})\right] \tag{26}$$

$$\leq \mathbb{P}\left[S^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) > S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), \bar{S}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) < \hat{\delta}\hat{\bar{S}}^f_{\boldsymbol{y}^*}(\boldsymbol{x}) + \epsilon\right] \tag{27}$$

$$= \mathbb{P}\left[S^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) > S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), \sum_{l_j \in \bar{\boldsymbol{y}}} T_{jj}\mathbb{P}(l_j|\boldsymbol{x}) + \sum_{l_j \in \bar{\boldsymbol{y}}}\sum_{i \neq j} T_{ij}\mathbb{P}(l_i|\boldsymbol{x}) < \hat{\delta}\hat{\bar{S}}^f_{\boldsymbol{y}^*}(\boldsymbol{x}) + \epsilon\right] \tag{28}$$

$$\leq \mathbb{P}\left[S^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) > S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), S^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) < \frac{\delta\hat{\bar{S}}^f_{\boldsymbol{y}^*}(\boldsymbol{x}) - \sum_{l_j \in \bar{\boldsymbol{y}}}\sum_{i \neq j} T_{ij}\mathbb{P}(l_i|\boldsymbol{x})}{\tau} + \frac{\epsilon + \rho_1}{\tau}\right]. \tag{29}$$

*If* $\delta = \min\left[\frac{\tau S^f_{\boldsymbol{b}_{\boldsymbol{x}}} + \sum_{l_j \in \bar{\boldsymbol{y}}}\sum_{i \neq j} T_{ij}\mathbb{P}(l_i|\boldsymbol{x})}{\hat{\bar{S}}^f_{\boldsymbol{y}^*}}\right]$, *we have*

$$\mathbb{P}\left[h^*(\boldsymbol{x}) = \bar{\boldsymbol{y}}, \frac{\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\hat{\bar{S}}^f_{\boldsymbol{y}^*}(\boldsymbol{x})} < \delta\right] \leq \mathbb{P}\left[S^f_{\boldsymbol{b}_{\boldsymbol{x}}} < S^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x}) < S^f_{\boldsymbol{b}_{\boldsymbol{x}}} + \frac{\epsilon + \rho_1}{\tau}\right] = C_1[O(\max(\epsilon, \rho_1))]^{\lambda_1}. \tag{30}$$

*Therefore,*

$$\mathbb{P}[\bar{\boldsymbol{y}}_{new} = h^*(\boldsymbol{x}), \bar{\boldsymbol{y}}\ is\ flipped] \geq 1 - C_1[\max(\epsilon, \rho_1)]^{\lambda_1} - \mathbb{P}[\boldsymbol{a}_{\boldsymbol{x}} \neq \{\boldsymbol{y}^*, \bar{\boldsymbol{y}}\}]. \tag{31}$$

*The case (2) shares the similar proof with the case (1). Specifically,*

$$\mathbb{P}[\bar{\boldsymbol{y}}_{new} \neq h^*(\boldsymbol{x}), \bar{\boldsymbol{y}}\ is\ accepted] = \mathbb{P}\left[\bar{\boldsymbol{y}}_{new} \neq h^*(\boldsymbol{x}), \frac{\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\hat{\bar{S}}^f_{\boldsymbol{y}^*}(\boldsymbol{x})} \geq \hat{\delta}\right] \tag{32}$$

$$\leq \mathbb{P}\left[S^f_{\boldsymbol{y}^*}(\boldsymbol{x}) > S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), \hat{\bar{S}}^f_{\boldsymbol{y}^*} \leq \hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})/\hat{\delta}\right] + \mathbb{P}[\boldsymbol{a}_{\boldsymbol{x}} \neq \{\boldsymbol{y}^*, \bar{\boldsymbol{y}}\}] \tag{33}$$

*For the first term, we have*

$$\mathbb{P}\left[S^f_{\boldsymbol{y}^*}(\boldsymbol{x}) > S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), \hat{\bar{S}}^f_{\boldsymbol{y}^*} \leq \hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})/\hat{\delta}\right] \tag{34}$$

$$\leq \mathbb{P}\left[S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) < S^f_{\boldsymbol{y}^*}(\boldsymbol{x}) \leq \frac{\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})/\hat{\delta} - \sum_{l_j \in \boldsymbol{y}^*}\sum_{i \neq j} T_{ij}\mathbb{P}(l_i|\boldsymbol{x})}{\tau} + \frac{\epsilon}{\tau}\right] \tag{35}$$

$$\leq \mathbb{P}\left[S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) < S^f_{\boldsymbol{y}^*}(\boldsymbol{x}) \leq \frac{\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})/(\delta - \rho_2) - \sum_{l_j \in \boldsymbol{y}^*}\sum_{i \neq j} T_{ij}\mathbb{P}(l_i|\boldsymbol{x})}{\tau} + \frac{\epsilon}{\tau}\right] \tag{36}$$

$$= \mathbb{P}\left[0 < S^f_{\boldsymbol{y}^*}(\boldsymbol{x}) - S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) < \frac{\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})/\delta - \sum_{l_j \in \boldsymbol{y}^*}\sum_{i \neq j} T_{ij}\mathbb{P}(l_i|\boldsymbol{x})}{\tau} - S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) + \frac{\epsilon}{\tau} + \frac{\frac{\rho_2\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\delta(\delta-\rho_2)}}{\tau}\right] \tag{37}$$

*If* $\delta = \max\left[\frac{\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\tau S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) + \sum_{l_j \in \boldsymbol{y}^*}\sum_{i \neq j} T_{ij}\mathbb{P}(l_i|\boldsymbol{x})}\right]$, *we have*

$$\mathbb{P}\left[S^f_{\boldsymbol{y}^*}(\boldsymbol{x}) > S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}), \hat{\bar{S}}^f_{\boldsymbol{y}^*} \leq \hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})/\hat{\delta}\right] \tag{38}$$

$$\leq \mathbb{P}\left[0 < S^f_{\boldsymbol{y}^*}(\boldsymbol{x}) - S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) \leq \frac{\epsilon}{\tau} + \frac{\frac{\rho_2\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\delta(\delta-\rho_2)}}{\tau}\right] \tag{39}$$

$$\leq \mathbb{P}\left[0 \leq S^f_{\boldsymbol{y}^*}(\boldsymbol{x}) - S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) \leq \frac{\epsilon}{\tau} + \frac{\rho_2\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\delta^2\tau} + \frac{\rho_2 O(\rho_2)\hat{\bar{S}}^f_{\bar{\boldsymbol{y}}}(\boldsymbol{x})}{\delta^2\tau}\right] \tag{40}$$

$$\leq \mathbb{P}\left[0 \leq S^f_{\boldsymbol{y}^*}(\boldsymbol{x}) - S^f_{\boldsymbol{b}_{\boldsymbol{x}}}(\boldsymbol{x}) \leq \frac{\epsilon}{\tau} + \frac{\rho_2 m}{\delta^2\tau} + \frac{\rho_2^2 m}{\delta^2\tau}\right]. \tag{41}$$

*Here, since* $\epsilon \leq \frac{t_0\delta^2\tau-\rho_2 m-\rho_2^2 m}{\delta^2}$, *we have* $\frac{\epsilon}{\tau}+\frac{\rho_2 m}{\delta^2\tau}+\frac{\rho_2^2 m}{\delta^2\tau} \leq t_0$. *Therefore,*

$$\mathbb{P}[\bar{\boldsymbol{y}}_{new} = h^*(\boldsymbol{x}), \bar{\boldsymbol{y}} \; is \; accepted] \geq 1 - C_1[O(\max(\epsilon, \rho_2))] - \mathbb{P}[\boldsymbol{a}_{\boldsymbol{x}} \neq \{\boldsymbol{y}^*, \bar{\boldsymbol{y}}\}]. \tag{42}$$

The proof of Lemma 1 is completed. Combining Lemma 1 and Theorem 1, Corollary 1 can be achieved.

# C Related Literature

## C.1 Procedure of ADDGCN

ADDGCN is the preparation technology of our CCMLN. We detail ADDGCN (Ye et al., 2020) as follows.

**SAM.** Given an example $(\boldsymbol{x}, \boldsymbol{y})$, we feed $\boldsymbol{x}$ into a deep network and obtain its corresponding feature map $\boldsymbol{x}'$. SAM first calculates label-specific activation maps $\mathbf{M} = [\boldsymbol{m}_1, \ldots, \boldsymbol{m}_q]$ by using class-activation-mapping (Zhou et al., 2016). Then, $\mathbf{M}$ is used to convert the feature map $\boldsymbol{x}'$ into the content-aware class-label representations $\mathbf{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_q]$. Let $[q] = \{1, \ldots, q\}$. Mathematically, for $k \in [q]$, we have $\boldsymbol{c}_k = \boldsymbol{m}_k^{\mathrm{T}}\boldsymbol{x}'$. That is, $\boldsymbol{c}_k$ selectively aggregate features related to its specific class label $k$.

**GCNM.** With the content-aware class-label representations $\mathbf{C}$ achieved by SAM, GCNM is introduced to adaptively transform their coherent correlation for multi-label classification. Specifically, GCNM consists of two parts: a static GCN and a dynamic GCN. The representations $\mathbf{C}$ are taken by GCNM as input node features and sequentially fed into the static GCN and dynamic GCN.

The single layer of the static GCN is defined as $\mathbf{H} = \mathrm{LReLU}(\mathbf{A}^s\mathbf{C}\mathbf{W}^s)$, where $\mathbf{A}^s$ denotes the correlation matrix shared for all instances, $\mathbf{W}^s$ denotes state-update weights, and $\mathrm{LReLU}(\cdot)$ denotes the LeakyReLU activation function (Xu et al., 2015). Besides, $\mathbf{A}^s$ and $\mathbf{W}^s$ are randomly initialized and learned by gradient decent during training. The dynamic GCN transforms $\mathbf{H}$. Its correlation matrix $\mathbf{A}^d$ is constructed dynamically dependent on input features $\mathbf{H}$. Namely, each examples have different $\mathbf{A}^d$. Formally, the output of the dynamic GCN is formulated as $\mathbf{Z} = \mathrm{LReLU}(\mathbf{A}^d\mathbf{H}\mathbf{W}^d)$, where $\mathbf{W}^d$ are state-update weights. Later, we use $\mathbf{A}^d(\boldsymbol{x})$ to denote the correlation matrix of $\boldsymbol{x}$, where $\mathbf{A}^d(\boldsymbol{x})_{jk} = \hat{\mathbb{P}}(l_k|l_j, \boldsymbol{x})$ for any $j, k \in [q]$.

**Classification and Loss.** The label-specific activation map $\mathbf{M} = [\boldsymbol{m}_1, \ldots, \boldsymbol{m}_q]$ and final category representation $\mathbf{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_q]$ are employed simultaneously for multi-label classification. Specifically, we use global spatial pooling on $\mathbf{M}$ to obtain a score vector $\mathbf{s}^m = [s_1^m, \ldots, s_q^m]$. Besides, each category representation $\mathbf{Z}$ is put into a binary classifier to obtain another score vector $\mathbf{s}^z = [s_1^z, \ldots, s_q^z]$. We simply average two score vectors to predict more reliable results. The aggregated score vector is denoted as $\mathbf{s} = [s_1, \ldots, s_q] = [(s_1^m + s_1^z)/2, \ldots, (s_q^m + s_q^z)/2]$. The Sigmoid activation function $\sigma(\cdot)$ is then used on $\mathbf{s}$ for probabilistic interpretation. That is to say, $\sigma(\mathbf{s}) = [\sigma(s_1), \ldots, \sigma(s_q)] = [\hat{\mathbb{P}}(l_1|\boldsymbol{x}), \ldots, \hat{\mathbb{P}}(l_q|\boldsymbol{x})]$. The binary cross-entropy loss is exploited for the updates of all weights, i.e., $\mathcal{L} = \sum_{l_i \in \boldsymbol{y}} \log(\sigma(s_i))$.

Given a multi-label example $(\boldsymbol{x}, \bar{\boldsymbol{y}})$, for the two dependences $\hat{S}^f$ and $\hat{S}^l$, based on $\sigma(\mathbf{s})$ and $\mathbf{A}^d(\boldsymbol{x})$ achieved by learning with multiple *noisy* labels, they can be estimated as

$$\hat{\tilde{S}}_{\boldsymbol{z}}^f(\boldsymbol{x}) = \sum_{\{\bar{\boldsymbol{Y}}=\boldsymbol{z}, l_i \in \boldsymbol{z}\}} \hat{\mathbb{P}}(l_i|\boldsymbol{x}) \quad \text{and} \quad \hat{\tilde{S}}_{\boldsymbol{z}}^l(\boldsymbol{x}) := \sum_{\{\bar{\boldsymbol{Y}}=\boldsymbol{z}, l_i, l_j \in \boldsymbol{z}\}} \frac{1}{2}\left[\hat{\mathbb{P}}(l_j|l_i, \boldsymbol{x}) + \hat{\mathbb{P}}(l_i|l_j, \boldsymbol{x})\right]. \tag{43}$$

## C.2 Related Literature on Multi-Class Classification with Noisy Labels

Multi-class classification with noisy labels can date back to three decades ago (Angluin & Laird, 1988), and keeps vibrant in recent years (Han et al., 2018). There is a large body of recent works that include but do not limit to the estimation of the noise transition matrix (Patrini et al., 2017; Hendrycks et al., 2018; Shu et al., 2020; Xia et al., 2019; Zhu et al., 2021; Liu, 2022; Zhang et al., 2021d;b), confident sample selection (Wei et al., 2020; Yao et al., 2020; 2021; Wu et al., 2020; Huang et al., 2019; Mirzasoleiman et al., 2020; Song et al., 2019; Northcutt et al., 2017), robust loss function design (Ma et al., 2020; Zhou et al., 2021; Menon et al., 2019; Feng et al., 2021),

implicit/explicit regularization (Hu et al., 2020; Lukasik et al., 2020; Ma et al., 2018; Li et al., 2021a; Jiang et al., 2020), and the integration of diverse techniques (Nguyen et al., 2020; Li et al., 2020; Liu et al., 2020b; Ortego et al., 2021). We refer readers to (Song et al., 2022; Han et al., 2018) for comprehensive review on multi-class classification with noisy labels.

In addition, the methods belonging to *label correction* have attracted much attention in multi-class classification with noisy labels (Tanaka et al., 2018; Zheng et al., 2020; Zhang et al., 2021c). Generally speaking, this kind of methods relies the prediction of a classifier trained on the noisy dataset, which recalibrates labels to the mislabeled data. Benefiting from the memorization effect of deep networks (Arpit et al., 2017), the prediction is a good indicator to determine the clean label of mislabeled data. The dataset after label correction is then less noisy, which brings better generalization. However, few label-correction methods are investigated for multi-label classification with noisy labels, which is much more challenging than multi-class classification with noisy labels (Liu et al., 2021).

### C.3 Related Literature on Multi-Label Classification with Clean & Noisy Labels

We briefly review works on multi-label classification with clean labels. If there is no confusion, we directly state multi-label classification. Multi-label classification has been studied for many years (Zhang & Zhou, 2013; Liu et al., 2017; Zhang & Wu, 2014; Li et al., 2016; Chheda et al., 2021; Xie & Huang, 2021a). In consideration of the increasing needs of todays big data, lots of methods based on deep learning are proposed (Zhu & Wu, 2021; Ridnik et al., 2021; Gao & Zhou, 2021; Zhao et al., 2021; Yazici et al., 2020; Chen et al., 2020; Zhu et al., 2017; Wang et al., 2016; Wei et al., 2015; Guo et al., 2019; Chen et al., 2019a). In addition to the above works, some works (Chen et al., 2019b; Ye et al., 2020) claim that the label dependence can be used to enhance the learning of the instance-label dependence. They exploit graph convolutional networks to capture the -label dependence and inject the captured information into multi-label classification, following promising classification performance. Recently, imperfect training data make us consider the side-effect of noisy labels in multi-label classification. Till now, there are relatively few methods specifically targeting this realistic problem. More advanced methods need to be excavated.

Normally, these methods perform an overall model adjustment to combat noisy labels. However, these methods highly rely on additional information except for provided training data with noisy labels. For example, partial methods (Hu et al., 2019; Veit et al., 2017; Vahdat, 2017; Pene et al., 2021) learn an overall transition between noisy and clean labels to handle noisy labels, where a small dataset with clean labels is relied to guide the transition learning. Partial methods (Zhao & Gomes, 2021) introduce overall semantics-based regularization on training data to relieve the model's overfitting to noisy labels, where semantic label embeddings are injected with large-scale predefined word embeddings (Pennington et al., 2014; Devlin et al., 2018). Although the additional information is helpful, in many actual scenarios, it is luxurious or not feasible at all. Without the additional information, these methods become weak in multi-label classification with noisy labels (Zhao & Gomes, 2021), which greatly limits their practical applications (Liu et al., 2021).

### C.4 Setting Difference between Multi-Label Classification with Noisy Labels and Partial Multi-Label Learning

It should be noted that the problem settings of multi-label classification with noisy labels and partial multi-label learning (Xie & Huang, 2021b; Kundu & Tighe, 2020) are different. Partial multi-label learning deals with the problem where each instance is assigned with a candidate label set, which contains multiple relevant labels and some irrelevant labels. The size of the candidate label set is usually much smaller than the size of label space. We need to detect the relevant labels for training. However, for our problem, there is no small candidate label set for reference, where we can only observe the whole label space. Intuitively, the methods in partial multi-label learning could be applied to multi-label classification with noisy labels. That is, we can identify some clean labels from noisy labels for training. However, this paradigm is inefficient, since only fractional labels are considered. Additionally, it is rather hard to accurately determine the number of identified labels for each instance.

# D SUPPLEMENTARY EXPERIMENTAL SETTINGS

## D.1 THE DETAILS OF BASELINES

In the main paper, we consider three types of baselines in experiments. Here, we detail the baselines.

1. Type-I baselines are designed for multi-label classification without considering noisy labels, which include

- CSRA (Zhu & Wu, 2021) proposes a simple and effective residual attention for multi-label learning. CSRA generates class-specific features for different labels by using spatial attention scores, and then combines them with the class-agnostic average pooling features.
- ADDGCN (Ye et al., 2020) proposes to exploit a semantic attention module and a GCN module for multi-label classification. As we discussed in Section 2, ADDGCN is the preparation technology of our CCMLN.

2. Type-I baselines are designed for multi-class classification with noisy labels, which include

- APL (Ma et al., 2020) combines two mutually reinforcing robust loss functions. For this baseline, we employ its combination of normalized BCE and MAE for comparison. The trade-off hyperparameter for the combinations of NBCE and MAE is set to 1.
- CDR (Xia et al., 2021) handles multi-class noisy labels using network pruning. A parameter judgment criteria is proposed to distinguish the critical/non-critical parameters for memorizing clean labels. The non-critical ones are forbidden to update, which mitigates the overfitting to mislabeled data.
- JOINT (Tanaka et al., 2018) shares a similar philosophy compared with our method, i.e., label correction. It uses a joint optimization framework to handle noisy labels. The pseudo labels are generated dynamically by using the network's prediction to improve robustness. Meanwhile, regularizations about the class prior and entropy of prediction probabilities are used. In experiments, we utilize the hard-label version of JOINT (Tanaka et al., 2018).

3. Type-III baselines are designed for multi-label classification with noisy labels, which include:

- WSIC (Hu et al., 2019) consists of a clean net and a residual net. The aim is to learn a mapping from feature space to clean label space and a residual mapping from feature space to the residual between clean labels and noisy labels respectively. For fair comparison with our method, we only provide noisy training examples to WSIC.
- CCMN (Xie & Huang, 2022) establishes unbiased estimators with error bounds for solving the problem of multi-label learning with noisy labels, and further prove that the estimators are consistent with commonly used multi-label loss functions under some conditions.

4. The simple baseline that trains deep models on multi-label noisy datasets directly:

- BCE (Zhang & Zhou, 2013) uses the binary cross-entropy loss to train deep models in noisy datasets, without considering the side-effect of mislabeled data for generalization.

## D.2 THE DETAILS OF THE LABEL TRANSITION MATRIX

In this paper, we consider both symmetric and pairflip cases for the generation of noisy labels. Specifically, if the overall noise rate is $\varrho$, the label transition matrix for symmetric cases are defined as

$$\text{Sym. } \varrho\text{:} \quad T := \begin{bmatrix} 1-\varrho & \frac{\varrho}{q-1} & \cdots & \frac{\varrho}{q-1} & \frac{\varrho}{q-1} \\ \frac{\varrho}{q-1} & 1-\varrho & \frac{\varrho}{q-1} & \cdots & \frac{\varrho}{q-1} \\ \vdots & & \ddots & & \vdots \\ \frac{\varrho}{q-1} & \cdots & \frac{\varrho}{q-1} & 1-\varrho & \frac{\varrho}{q-1} \\ \frac{\varrho}{q-1} & \frac{\varrho}{q-1} & \cdots & \frac{\varrho}{q-1} & 1-\varrho \end{bmatrix}_{q\times q}. \tag{44}$$

The label transition matrix for pariflip cases are defined as

$$\text{Pair. } \varrho\text{:} \quad T = \begin{bmatrix} 1-\varrho & \varrho & \dots & 0 & 0 \\ 0 & 1-\varrho & \varrho & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & 0 & 1-\varrho & \varrho \\ \varrho & 0 & \dots & 0 & 1-\varrho \end{bmatrix}_{q \times q}. \tag{45}$$

# E SUPPLEMENTARY EXPERIMENTAL RESULTS

| Metrics | Methods / Noise | Sym. 30% | Sym. 40% | Sym. 50% | Pair. 20% | Pair. 30% | Pair. 40% |
|---|---|---|---|---|---|---|---|
| mAP ↑ | BCE | 82.01±0.61 | 80.50±0.62 | 76.80±0.31 | 80.97±0.24 | 75.95±1.12 | 65.54±2.67 |
| | CSRA | **83.15±0.08** | 80.39±1.17 | 77.93±2.73 | **82.36±0.35** | 76.02±1.58 | 65.38±1.61 |
| | ADDGCN | 81.70±0.96 | 80.29±0.44 | 74.22±2.86 | 80.33±1.50 | 74.92±2.64 | 63.11±1.80 |
| | APL | 82.13±1.44 | 79.92±0.63 | 76.68±2.47 | 82.20±0.09 | 76.02±1.80 | 66.92±2.09 |
| | CDR | 82.35±1.17 | 78.33±1.04 | 77.01±1.61 | 81.00±0.20 | 76.37±1.04 | 66.21±2.35 |
| | JOINT | 82.12±0.55 | 81.00±0.39 | 76.84±1.12 | 81.33±0.60 | **76.77±0.55** | 66.50±1.86 |
| | WSIC | 82.17±0.19 | 78.14±1.06 | 77.25±0.90 | 81.06±1.06 | 75.22±1.37 | 65.88±2.80 |
| | CCMN | 81.80±0.73 | 80.20±1.10 | 76.77±1.73 | 82.27±0.41 | 76.03±1.39 | 66.93±2.03 |
| | CCMLN† | 82.40±0.17 | **81.19±1.22** | **78.04±0.29** | 82.30±0.61 | 76.40±1.82 | **67.61±2.12** |
| OF1 ↑ | BCE | 70.97±0.65 | 62.99±0.75 | 55.43±1.80 | 75.95±0.77 | 71.26±0.88 | 63.33±2.74 |
| | CSRA | 73.52±1.06 | 65.21±0.93 | 52.84±2.11 | 78.02±0.92 | 72.66±1.75 | 62.77±3.06 |
| | ADDGCN | 71.11±1.16 | 63.05±1.84 | 48.62±2.31 | 74.29±1.72 | 67.83±0.98 | 59.12±2.73 |
| | APL | 71.10±0.50 | 61.44±0.88 | 51.77±2.84 | 74.50±1.29 | 68.04±1.98 | 63.30±1.60 |
| | CDR | 71.65±1.63 | 63.06±1.50 | 54.83±2.26 | 76.88±2.16 | 72.06±1.90 | 62.89±3.17 |
| | JOINT | 74.08±1.12 | 70.22±2.31 | 65.27±2.66 | 77.82±1.01 | 72.56±0.71 | 65.82±1.73 |
| | WSIC | 71.05±0.16 | 63.86±1.00 | 52.88±2.27 | 76.05±1.10 | 70.39±1.16 | 60.88±1.37 |
| | CCMN | 72.33±0.18 | 65.44±1.26 | 57.29±1.10 | 77.19±0.11 | 72.04±0.50 | 62.05±1.18 |
| | CCMLN† | **76.33±0.19** | **74.83±1.29** | **72.11±3.06** | **78.05±0.13** | **73.88±1.90** | **66.32±0.30** |
| CF1 ↑ | BCE | 68.33±0.92 | 59.63±1.29 | 49.77±2.81 | 73.17±1.46 | 66.82±2.96 | 57.19±2.32 |
| | CSRA | 70.59±1.26 | 62.33±1.60 | 48.15±2.90 | 75.06±0.77 | 68.72±1.63 | 56.25±3.28 |
| | ADDGCN | 67.83±0.64 | 59.75±1.06 | 46.72±3.50 | 71.33±0.65 | 64.02±0.65 | 55.82±4.91 |
| | APL | 67.33±1.85 | 59.11±2.02 | 47.86±3.13 | 74.80±0.77 | 66.92±2.84 | 57.02±1.90 |
| | CDR | 68.03±1.62 | 60.02±1.17 | 48.94±2.65 | 73.77±1.04 | 67.06±1.84 | 57.38±2.10 |
| | JOINT | 71.17±0.29 | 66.11±1.59 | 57.93±1.82 | 75.25±0.73 | 70.01±1.99 | 56.28±2.19 |
| | WSIC | 68.11±0.52 | 60.39±1.14 | 46.25±4.74 | 74.02±1.26 | 67.09±2.84 | 55.76±3.66 |
| | CCMN | 68.58±0.44 | 64.82±2.17 | 54.82±1.06 | 74.15±0.92 | 67.79±2.33 | 58.06±2.37 |
| | CCMLN† | **73.11±0.91** | **72.08±1.16** | **68.31±0.77** | **76.00±0.71** | **71.07±1.95** | **61.52±2.28** |

Table 7: Comparisons with advanced methods on noisy Pascal-VOC 2007. The mean and standard deviation of the best results (%) during training are presented.

In the main paper, we report results based on the performance of the last epoch during training, as did in (Han et al., 2018; Wang et al., 2018; Wei et al., 2020; Li et al., 2020). Here, to make comparison more comprehensive, we report results on noisy Pascal-VOC 2007 based on the best performance achieved during training. The results are provided in Table 7. Due to the memorization effect of deep networks (Arpit et al., 2017), the networks would first memorize clean training data and then noisy training data. Therefore, in the early training, all methods could achieve good performance. We compared CCMLN with other advanced methods. Specifically, for mAP, although CCMLN does not always achieve the best results like the results in the main paper, the results are still competitive. For OF1 and CF1, CCMLN outperforms the other methods consistently.

It is worth mentioning that, the results in Table 1 are much lower than the results in Table 7 in some cases. The experimental phenomenon means that one method severely overfits training data with incorrect labels as training progresses, which is pessimistic. Therefore, we should strive to design more robust methods to address the problem of multi-label classification with noisy labels. In this paper, we try and give a potential method, which outperforms baselines clearly. More efforts are expected to be put in by the community.