Data Augmentation with Regularization for Multi-labeled Complementary Label Learning

Yi Gao, Yuan-Yuan Meng, Miao Xu and Min-Ling Zhang*, *Senior Memeber, IEEE*

Abstract—*Multi-labeled complementary label learning* (MLCLL) is a resource-efficient paradigm aimed at reducing labeling efforts in *multi-label learning* (MLL). While existing methods address the MLCLL problem using neural network-based models, they often overfit to noisy information, leading to sharp decision boundaries. This overfitting issue is further exacerbated when the label correlation, which could help denoise the supervision, is not fully explored in existing works. In this paper, we propose a novel framework called NMCB to alleviate the impact of noisy information in MLCLL, which makes a first attempt to explore mixup for MLCLL problem. Specifically, a tailored version of mixup is employed to achieve a smoother decision boundary of the trained classifier, thereby reducing the sensitivity of NMCB to noisy labels and enhancing its generalization ability. Moreover, NMCB applies a model to automatically extract label correlations from non-complementary labels transformed by mixup during the learning process. These extracted correlations serve as alignment objectives for the output distribution of instance augmentations within a consistency regularization term of NMCB, further improving the model performance. Empirical studies demonstrate the effectiveness of the proposed method.

Index Terms—Complementary label learning, multi-label learning, mixup, label correlations, consistency regularization.

1 Introduction

In real-world scenarios, an instance can be relevant to multiple labels simultaneously, with these labels exhibiting complex and intertwined correlations [1], [2], [3]. This characteristic has sparked significant interest among researchers in *multi-label learning* (MLL), which aims to learn a classifier capable of assigning a set of relevant labels to an unseen instance [2], [4], [5], [6]. However, the uncertain number of relevant labels per instance and the complexity of semantic labels pose obstacles for annotators in precisely labeling MLL data [7]. For example, precisely annotating the image shown in Fig. 1 demands a high level of attention to distinguish the portrait on the wall as a "picture" rather than "people" and expertise to identify specific geographic location labels such as "France". Besides, identifying the remaining relevant labels requires exhaustive exploration of the entire label space, which can be burdensome for annotators, especially when the label space is large.

To alleviate the laborious efforts involved in precisely labeling multi-labeled data, the *multi-labeled complementary label learning* (MLCLL) paradigm has been proposed [7], where instances are associated with complementary (irrelevant) labels. Compared with collecting precise labels, acquiring complementary labels is significantly more cost-effective and less laborious, since it does not need the exhaustive examination of the entire label space and prior knowledge. Similar to fully supervised MLL, the goal of MLCLL is to train a classifier from complementary labels to identify relevant labels for unseen instances [8].

- *Yi Gao, Yuan-Yuan Meng and Min-Ling Zhang (* corresponding author) are with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China and the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China. E-mail: gao_yi, mengyy, zhangml@seu.edu.cn.*
- *Miao Xu is with the University of Queensland, Australia. E-mail: miao.xu@uq.edu.au.*

Fig. 1: An example. Relevant labels of this image include "sky", "cloud", "house", "lamp", "picture", "France", and others. Conversely, the label "river" is an complementary label of this image, indicating that there is no river in this image at all. Labeling the image with the label "France" requires specialized geographic knowledge, while annotating the label "picture" (as it is highlighted by a blue box in the image) needs a high-level attention of annotators.

As a pioneering work, Ishida et al. [9] proposed *complementary label learning* (CLL) for multi-class scenarios. Estimating transition matrix is a common and effective strategy to solve the CLL problem, in which estimated transition matrix can recover multi-class data from complementary labeled data [10], [11]. Furthermore, CLL problems are investigated for generative adversarial network [12], contrastive learning [13], easing the dependence on the transition matrix [14], or multiple complementary labels [15]. The design of these methods is based on the fact that only one relevant label exists for multi-class cases. They cannot be applied to the MLCLL problem since the uncertain number of relevant labels in MLL given one complementary label.

Subsequently, Gao *et al.* [7] were the first to develop a neural network-based method for MLCLL by constructing the true multi-labeled data distribution from complementary labels. In the former work [7], non-complementary labels (i.e., all labels in the label space excluding complementary ones) are used without fully

considering the label correlation, which is an important source of information for solving problems with limited supervision as MLCLL. Note that a substantial portion of the non-complementary label set consists of irrelevant (noisy) labels, since relevant labels are sparse [3], [7] and only one complementary label is provided for each instance. Due to the memorization effect, the learned neural network model gradually adapts to fit noisy labels after fitting the relevant ones during the learning process [16], [17]. As a result, the decision boundary of the classifier trained as in [7] may be sharp due to overfitting to noisy labels, resulting in weak generalization [18], [19], [20].

In this paper, we propose a novel framework called NMCB (**N**oise**M**ix **C**onsistency **B**oost) to address the weak generalization issues caused by noisy information. Motivated by the insight that smooth decision boundaries enhance generalization [20], [21], [22], NMCB pioneers the exploration of achieving smoother decision boundaries and reducing sensitivity to noisy labels. Building on the concept of mixup [18], [20], [23], NMCB introduces a version of mixup tailored for MLCLL, which creates a smooth set of new instances with soft supervision based on non-complementary labels—instead of unavailable relevant labels—thereby mitigating the effects of label noise and fostering smoother learning behavior. To address the label correlation distortion introduced by complementary labeling, NMCB introduces a label correlation extraction module that automatically learns correlations from the soft supervision of the non-complementary labels generated by the mixup component. To further enhance performance, we propose a consistency regularization term that aligns the model's predictions for augmented instances with the extracted label correlations. These components work synergistically to reduce noise sensitivity and improve predictive performance. Under a mild assumption, we establish a theoretical generalization bound for our method to characterize its performance on unseen instances (see Appendix A for completeness). Our experimental results demonstrate the effectiveness of the proposed method. The main contributions of this paper are summarized as follows:

- We propose a novel framework called NMCB, which is the first to explore a tailored version of mixup in MLCLL. This manner enables it to achieve smoother decision boundaries, thereby reducing sensitivity to noisy labels and enhancing its generalization.
- To alleviate label correlation distortion in MLCLL, NMCB adopts a model to automatically extract label correlation from non-complementary labels transformed by the tailored mixup during the training process.
- In NMCB, a new consistency regularization term is introduced to improve performance by emphasizing the alignment between the output distribution of instance augmentations and the extracted label correlations in the embedding space.

The rest of this paper is organized as follows. We review the related work of MLCLL in Section 2 and provide background information in Section 3. The details of the proposed method are presented in Section 4. Section 5 presents the experimental results, and Section 6 provides the conclusion.

2 Related Work

As a weakly supervised learning paradigm, MLCLL aims to solve the MLL problem while minimizing the cost of obtaining labeling information. In this section, we briefly review related work, including MLL, *partial multi-label learning* (PML), and CLL. Additionally, we discuss recent advances in mixup and label augmentation under weak supervision.

2.1 Multi-label Learning

In MLL, each instance is simultaneously associated with multiple relevant labels. According to the order of label correlations exploited during model training, previous MLL methods can be roughly divided into three categories: first-order strategy [24], second-order strategy [25], [26] and high-order strategy [27]. First-order methods address MLL problems by decomposing them into a sequence of binary classification tasks [24], [27]. Zhang et al. [2] observed that label correlations exist in multi-labeled data, while these are ignored by first-order methods. Subsequently, second-order methods and high-order methods are proposed to consider label correlations to address MLL problems. Among them, second-order methods focus on label correlations between label pairs, where the rankings between relevant and irrelevant labels [25], or any pair of labels [26]. Recognizing the more intricate relationships among labels beyond second-order in many real-world scenarios, high-order methods explore label correlations among label subsets or all labels [28], [29], [30]. For example, Zhao et al. [29] utilize variational autoencoders to exploit high-order correlations among labels, enhancing the learning process. On the other hand, Wang et al. [31] and Xun et al. [32] both employed specialized neural network components to automatically capture label correlations. It is worth noting that high-order methods have the capacity for stronger label correlation-modeling, while they may suffer from the cost of increased computational complexity when compared to first and second-order methods [33]. Compared with MLL, the task of MLCLL is more challenging than MLL because relevant labels are unavailable for the MLCLL problem.

2.2 Partial Multi-label Learning

Due to the challenge of collecting fully supervised data, many researchers have turned to explore weakly supervised learning as a way to alleviate the labeling burden in MLL [34]. PML, initially proposed by Xie et al. [35], is a weakly supervised learning paradigm where each instance is associated with a set of candidate labels composed of both relevant labels and irrelevant (noisy) labels. According to the training process, existing PML methods can be roughly categorized into two groups [36]: end-to-end strategy and two-stage strategy. In the case of methods falling under the end-to-end strategy, they leverage a unified framework to learn from PML data by estimating the confidence level for each candidate label [35], [37], [38]. Then, these estimated confidence scores are incorporated into an alternative optimization procedure for training the model. In the case of methods following the two-stage strategy, the training process is divided into two stages, where high-confidence labels are first selected from the candidate label set and utilized to train the desired model with conventional MLL methods [39], [40]. Regardless of the strategies employed by these methods, they handle PML problems with the assumption that noisy labels only compose a small portion of candidate labels [7], [33], [41]. In fact, MLCLL tackles the hardest version of this problem – a high-noise PML problem, in which the candidate label set for an instance involves all labels except its complementary label [7]. In such scenarios, existing PML methods

cannot be directly employed to MLCLL because the PML problem assumes that candidate labels contain sparse noisy labels.

2.3 Complementary Label Learning

As an emerging research field, CLL was initially proposed to tackle the problem of multi-class learning, which aims to ease the heavy burden of collecting precisely labeled data [9]. In CLL, Ishida et al. [10] utilized uniformly sampled complementary labels to derive an unbiased risk estimator that can accommodate arbitrary loss functions for solving the CLL problem. Furthermore, the exploration of CLL in multi-class settings has extended to encompass biased complementary labels, an extension that depends on estimating a transition matrix to recover relevant labels from complementary labels [11]. To reduce the reliance on an estimated transition matrix, Gao et al. [14] proposed a discriminative method to directly model the probabilities of complementary labels using the model's outputs. In addition, Feng et al. [15] further explored multiple complementary labels to enhance the labeling information during the learning process. The aforementioned methods benefit from the fact that each instance has a single relevant label, which poses challenges in effectively solving the MLCLL problem since the number of relevant labels is uncertain and can vary across instances in MLL. Recognizing the difficulty of collecting precisely multi-labeled data, Gao et al. [7] first proposed MLCLL to explore the application of CLL in solving the MLL problem. In their work, a neural network-based method was introduced to address the MLCLL problem by deriving an unbiased risk estimator, which constructs the multi-labeled data distribution from complementary labels under two mild assumptions. Moreover, they improved the unbiased risk estimator by designing a *gradient-descent friendly* (GDF) loss function. As discussed earlier, their method may gradually fit noisy labels due to the memorization effect of neural networks, which may lead to a sharp decision boundary and weak generalization. In addition, their method does not take full advantage of label correlations, which remains a key challenge in MLCLL.

2.4 Mixup and Label Augmentation under Weak Supervision

Recent studies have explored the use of mixup and label augmentation under weak supervision. For example, Li et al. [42] investigated mixup strategies in positive-unlabeled learning, which focuses on the importance of choosing suitable mixup partners to mitigate label noise. However, their work is limited to binary classification with partially labeled positive instances, which differs significantly from the MLCLL scenario, where the supervision is extremely weak and relevant labels are completely unavailable. Similarly, Lin et al. [43] proposed a label augmentation method in CLL to improve the efficiency of label information sharing. Their method is designed for multi-class classification and relies on class priors, making it unsuitable for multi-label scenarios. In contrast, our proposed NMCB framework is the first to introduce a tailored mixup strategy for the MLCLL problem. It facilitates smoother decision boundaries, thereby reducing sensitivity to noisy labels and improving generalization. Additionally, NMCB introduces a model to automatically extract label correlations and enhance its performance.

3 Preliminaries

In MLL, let $\mathcal{X} \subset \mathbb{R}^d$ denote the feature space with a dimension of d, and $\mathcal{Y} = \{1, 2, \ldots, K\}$ be the label space with K possible labels, where K is greater than 2. A multi-labeled instance $\boldsymbol{x} \in \mathcal{X}$ is associated with a set of relevant labels $Y \subseteq \mathcal{Y}$, which is sampled from the joint probability distribution $p(\boldsymbol{x}, Y)$. For convenience, we represent Y with a K-dimensional vector $\boldsymbol{y} = [y^1, y^2, \ldots, y^K] \in \{0,1\}^K$, where $y^k = 1$ indicates the label k being a relevant label of $\boldsymbol{x}$ (i.e., $k \in Y$), and 0 otherwise. The goal of MLL is to learn a multi-labeled classification model $\boldsymbol{f} : \mathcal{X} \mapsto [0,1]^K$ by minimizing the expected risk defined as follows:

$$R(\boldsymbol{f}) = \mathbb{E}_{p(\boldsymbol{x},Y)}[L(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{y})], \tag{1}$$

where $L : \mathbb{R}^K \times 2^{\mathcal{Y}} \mapsto \mathbb{R}_+$ is an MLL loss function, such as *binary cross-entropy* (BCE) loss, *mean squared error* (MSE) loss, etc. We denote $f^k(\boldsymbol{x})$ as the k-th prediction of $\boldsymbol{f}(\boldsymbol{x})$, which represents the estimation of $p(y^k = 1|\boldsymbol{x})$.

In MLCLL study, only the complementary labeled dataset $\bar{D} = \{(\boldsymbol{x}_i, \bar{y}_i)\}_{i=1}^n$ is given, which consists of n instances. Here, each instance $\boldsymbol{x}_i \in \mathcal{X}$ is paired with a complementary label $\bar{y}_i \in \{\mathcal{Y} - Y_i\}$. The complementary labeled instance $(\boldsymbol{x}, \bar{y})$ follows the joint probability distribution $\bar{p}(\boldsymbol{x}, \bar{y})$ from which $\bar{D}$ is drawn. We denote $\bar{y}$ as a K-dimensional vector $\bar{\boldsymbol{y}} = [\bar{y}^1, \bar{y}^2, \ldots, \bar{y}^K] \in \{0,1\}^K$, in which $\bar{y}^j = 1$ indicates that the label j serves as the complementary label of $\boldsymbol{x}$, and $\bar{y}^j = 0$ otherwise. Correspondingly, we utilize $\hat{Y} = \mathcal{Y} \setminus \bar{y}$ to denote the non-complementary label set of $\boldsymbol{x}$, where its vector representation is $\hat{\boldsymbol{y}} = [\hat{y}^1, \hat{y}^2, \ldots, \hat{y}^K] \in \{0,1\}^K$. The relationship between $\bar{\boldsymbol{y}}$ and $\hat{\boldsymbol{y}}$ can be expressed as $\hat{\boldsymbol{y}} = \mathbb{1} - \bar{\boldsymbol{y}}$, where $\mathbb{1}$ refers to a vector of all ones with K dimensions. The goal of MLCLL aligns with MLL, aiming to learn a multi-labeled classifier $\boldsymbol{f} : \mathcal{X} \mapsto [0,1]^K$ from $\bar{D}$ that can assign relevant labels for unseen instances. As discussed above, MLCLL was first proposed by Gao et al. [7]. Inspired by the challenges of gradient updating in MLCLL, they designed a GDF loss function. The GDF loss is defined as:

$$\bar{L}_{\mathrm{GDF}}(\boldsymbol{f}(\boldsymbol{x}), \bar{\boldsymbol{y}}) = -(\mathbb{1} - \bar{\boldsymbol{y}})\log(\boldsymbol{f}(\boldsymbol{x})) - \bar{\boldsymbol{y}}\log(\mathbb{1} - \boldsymbol{f}(\boldsymbol{x})). \tag{2}$$

While the GDF loss combined with a neural network has contributed to promising performance in MLCLL, it may result in a sharp decision boundary and weak generalization due to the memorization effect of neural networks. To alleviate this issue, in the next section, we introduce a novel framework called NMCB, which incorporates a tailored version of mixup specifically for the MLCLL problem to facilitate a smoother decision boundary and reduce the sensitivity of NMCB to noisy labels.

4 NoiseMix Consistency Boost

In this section, we begin by introducing the overall structure of the proposed method, NMCB, followed by an illustration of the tailored version of mixup for the MLCLL problem. We then describe the label correlation extraction model and present a consistency regularization technique based on these correlations.

4.1 Overview

Fig. 2 depicts the overall illustration of the proposed NMCB method, which consists of two models: a classification model and a label correlation extraction model. In the classification model, an instance $\boldsymbol{x}$ and its corresponding data augmentation $\boldsymbol{x}'$, transformed by the tailored version of mixup, are respectively fed into the classification model $\boldsymbol{f}$ to obtain their predictive probabilities $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{f}(\boldsymbol{x}')$. With the non-complementary labels

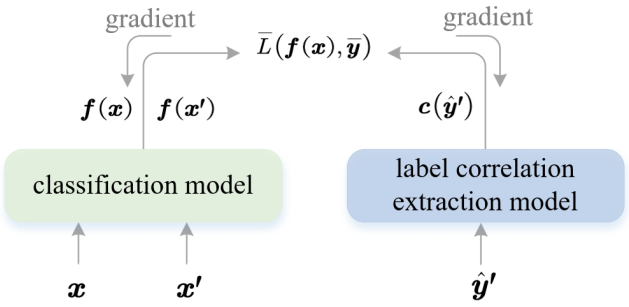


Fig. 2: Illustration of the proposed NMCB method. The instance $\boldsymbol{x}$ and its corresponding data augmentation $\boldsymbol{x}'$ are input into the classification model $\boldsymbol{f}$ individually. The outputs of these, along with the label correlations extracted by the model $\boldsymbol{c}$, are then used to compute the loss function $\bar{L}(\boldsymbol{f}(\boldsymbol{x}), \bar{\boldsymbol{y}})$.

transformed by mixup ($\hat{\boldsymbol{y}}'$), the label correlation extraction model $\boldsymbol{c}$ is employed to automatically capture label correlations among labels in $\hat{\boldsymbol{y}}'$. The loss function $\bar{L}(\boldsymbol{f}(\boldsymbol{x}), \bar{\boldsymbol{y}})$ designed in this paper is then applied to optimize both the classification model and the label correlation extraction model, taking into account $\boldsymbol{f}(\boldsymbol{x})$, $\boldsymbol{f}(\boldsymbol{x}')$, and the extracted label correlations $\boldsymbol{c}(\hat{\boldsymbol{y}}')$. Note that the parameters of these two models are updated simultaneously through back-propagation during the learning process. We proceed to introduce the key components of NMCB in detail in the following sections.

4.2 A Tailored Version of Mixup

As is well-known, neural networks commonly exhibit a memorization effect, which causes the model to gradually fit noisy labels after fitting relevant ones during the learning process [16], [17]. In MLCLL, supervision comes exclusively from complementary labels, and the corresponding non-complementary label set inevitably contains substantial noisy labels. This effect can lead neural network-based methods to fit noisy labels, resulting in a sharp decision boundary and weak generalization. Theoretically, margin-based analysis shows that classifiers with larger margins (i.e., lower margin complexity) enjoy stronger generalization guarantees [21]. Geometrically, larger margins imply that decision boundaries lie farther from training data and are less prone to complex shapes, which tends to produce smoother boundaries [44]. Empirically, mixup-based methods have been shown to encourage smoother boundaries and improve robustness [18], [20]. While these works are not specific to MLCLL, their core insight—that smoother decision boundaries mitigate noise and enhance generalization—is directly applicable to MLCLL. Motivated by this, we are eager to seek a strategy that can alleviate the impact of noisy labels in the MLCLL problem, which facilitates the model to achieve a smoother decision boundary and reduces the model's sensitivity to noisy labels.

A recent line of work in data augmentation has proposed mixup as a powerful tool for facilitating neural network-based methods to smooth decision boundaries and reduce the sensitivity of a model to noisy labels [18], [19], [23]. Mixup is a straightforward yet effective technique that creates new instances through convex combinations of training instances and their corresponding labels [23]. Compared with the features and labels of the original instances, the features and labels of the newly created instances lie between those of two original instances, resulting in a smoother representation. Learning with smoother features encourages the neural network-based model to obtain a smoother decision boundary during the learning phase. In this case, the model exhibits reduced sensitivity to overfitting on noisy labels and enhanced generalization.

It is worth noting that the standard mixup technique in supervised learning operates under the assumption that relevant labels are available for all training instances. However, in MLCLL, only weak supervision in the form of complementary labels is provided, and the derived non-complementary labels may contain noise. Directly applying standard mixup in this setting is infeasible due to the absence of fully supervised information. Hence, we introduce a tailored version of mixup for the proposed method NMCB, which creates a new instance used by non-complementary labels instead of relevant labels. The process of creating a new instance $(\boldsymbol{x}', \hat{\boldsymbol{y}}')$ in this mixup version is illustrated as follows:

$$\begin{aligned} \boldsymbol{x}' &= \lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{x}_j, \\ \hat{\boldsymbol{y}}' &= \lambda \hat{\boldsymbol{y}} + (1-\lambda)\hat{\boldsymbol{y}}_j. \end{aligned} \tag{3}$$

Here, $\boldsymbol{x}$ and $\boldsymbol{x}_j$ are original instances belonging to $\bar{D}$, $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{y}}_j$ are their non-complementary label vectors, respectively. The parameter $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$ represents the mixing level, where the mixup hyper-parameter $\alpha \in (0, \infty)$ regulates the strength of interpolation between feature-target pairs.

4.3 Label Correlation Extraction

Label correlations are prevalent in multi-labeled data and reveal relationships among different labels [33], [45]. For example, in real-world scenarios, "street lights" often accompany "streets" but are unlikely to appear in the "sea". Motivated by this, we recognize that label correlations can help a model infer information about other labels from one, thereby enhancing the model's performance. However, directly extracting label correlations from complementary labels may distort the true relationships among labels due to the presence of noisy labels in MLCLL.

As discussed earlier, the non-complementary label vector transformed by the tailored version of mixup becomes softer compared to the original ones, which helps reduce the impact of noise for learning a model. Therefore, NMCB applies a model $\boldsymbol{c} : [0,1]^K \mapsto \mathbb{R}^K$ to automatically extract label correlations from non-complementary labels transformed by the tailored version of mixup. Specifically, $\boldsymbol{c}$ is a fully connected layer that needs to be learned:

$$\boldsymbol{c}(\hat{\boldsymbol{y}}') = \mathbf{W}\hat{\boldsymbol{y}}', \tag{4}$$

where $\mathbf{W} \in \mathbb{R}^{K \times K}$ denotes the weight matrix of the layer, with the bias term and activation function omitted for simplicity. In this way, the i-th correlation factor $c_i(\hat{\boldsymbol{y}}')$ is a linear combination of all elements in $\hat{\boldsymbol{y}}'$ and hence all possible linear correlations between the i-th label and other labels are taken into consideration. These extracted label correlations serve as the alignment objective for the outputs of the classification model, which encourages this classification model to consider correlations among labels in the training phase.

4.4 Consistency Regularization Depending on Label Correlations

As discussed above, we analyze that label correlations play a crucial role in MLCLL. Hence, we expect to encourage the predictions of the classification model for a label involving the collaboration between its own prediction and the predictions of other

Algorithm 1: NMCB Algorithm

Input:
$\bar{D}$: the complementary-label training set $\{(\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i)\}_{i=1}^n$;
$\boldsymbol{\theta}_1$: the initial parameters of classifier $\boldsymbol{f}$;
$\boldsymbol{\theta}_2$: the initial parameters of label correlation extraction model $\boldsymbol{c}$;
T: the number of epochs;
β: maximum balancing weight;
$\mathcal{A}$: an external stochastic optimization algorithm;
Output:
$\boldsymbol{f}$: learned multi-labeled classifier;

for $t = 1$ *to* T **do**
  Let $\mathcal{L}$ be the risk, $\mathcal{L} = \frac{1}{n}\sum_{i=1}^n \bar{L}(\boldsymbol{f}(\boldsymbol{x}_i), \bar{\boldsymbol{y}}_i) = \frac{1}{n}\sum_{i=1}^n \{\bar{L}_{\mathrm{GDF}}(\boldsymbol{f}(\boldsymbol{x}_i), \bar{\boldsymbol{y}}_i) + \psi(t)\Phi(\boldsymbol{x}_i, \bar{\boldsymbol{y}}_i)\}$;
  Set gradients $-\nabla_{\boldsymbol{\theta}_1}\mathcal{L}$ and $-\nabla_{\boldsymbol{\theta}_2}\mathcal{L}$;
  Update $\boldsymbol{\theta}_1$ by $\mathcal{A}$ with $-\nabla_{\boldsymbol{\theta}_1}\mathcal{L}$;
  Update $\boldsymbol{\theta}_2$ by $\mathcal{A}$ with $-\nabla_{\boldsymbol{\theta}_2}\mathcal{L}$;
end

labels to improve the model's performance. A simple strategy towards this goal is to guide the multi-labeled classification model to consider label correlations throughout the learning process. We accomplish that by minimizing the divergence between the classification model's outputs of augmentations transformed by a tailored version of mixup and the corresponding label correlation vectors.

With the help of consistency-regularized training, the outputs of the classification model are forced to keep consistency with label correlations. Specifically, we implement the consistency regularization term using MSE loss (ℓ_2-norm) to emphasize the alignment of the model's outputs with the extracted label correlations. The consistency regularization term is defined as:

$$\Phi(\boldsymbol{x}, \bar{\boldsymbol{y}}) = \|\boldsymbol{f}(\boldsymbol{x}') - \boldsymbol{c}(\hat{\boldsymbol{y}}')\|_2^2, \tag{5}$$

where $\Phi : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}^+$ denotes the MSE loss, and the newly created data point $(\boldsymbol{x}', \hat{\boldsymbol{y}}')$ is computed by Eq. (3). Here, the complementary labeled data $(\boldsymbol{x}_j, \bar{\boldsymbol{y}}_j)$ is randomly sampled from $\bar{D}\backslash(\boldsymbol{x}, \bar{\boldsymbol{y}})$. In our framework, MSE is selected as the regularization method in the consistency term because of its simplicity and smoothness. To further investigate this design choice, we conducted additional experiments, which are shown in Appendix B.1.

To ensure the ability of the model in handling complementary labels, we combine the MLCLL loss – GDF loss – with the regularization term $\Phi(\boldsymbol{x}, \bar{\boldsymbol{y}})$. The final loss function is expressed as:

$$\bar{L}(\boldsymbol{f}(\boldsymbol{x}), \bar{\boldsymbol{y}}) = \bar{L}_{\mathrm{GDF}}(\boldsymbol{f}(\boldsymbol{x}), \bar{\boldsymbol{y}}) + \beta\Phi(\boldsymbol{x}, \bar{\boldsymbol{y}}), \tag{6}$$

where β represents a trade-off parameter to balance the contributions of these two loss terms. The final loss function can be effectively optimized during the training procedure, which guides the parameter updating of both the classification model and the label correlation extraction model. Moreover, it promises that NMCB possesses the capability to handle complementary labels and align the classification model outputs with the extracted label correlations, which further enhances the performance of NMCB.

In Eq. (6), a trade-off factor β is used to control the strength of regularization. Notably, strong regularization may degrade the performance of the proposed method since the classification model may produce low-quality predictions in the initial stage. This concern is exacerbated in MLCLL due to the presence of noisy labels in non-complementary labels. As predictions of the model progressively improve after some training epochs, strong regularization would become more beneficial. Inspired by this, we adopt a dynamic trade-off parameter to progressively strengthen regularization in $\bar{L}(\boldsymbol{f}(\boldsymbol{x}), \bar{\boldsymbol{y}})$, that is,

$$\bar{L}(\boldsymbol{f}(\boldsymbol{x}), \bar{\boldsymbol{y}}) = \bar{L}_{\mathrm{GDF}}(\boldsymbol{f}(\boldsymbol{x}), \bar{\boldsymbol{y}}) + \psi(t)\Phi(\boldsymbol{x}, \bar{\boldsymbol{y}}), \tag{7}$$

where β is replaced by a dynamic parameter that varies with the epoch number t, i.e.,

$$\psi(t) = \min\{\frac{t}{T'}\beta, \beta\}. \tag{8}$$

This dynamic parameter assigns a small weight to the regularization term in the initial epochs and progressively increases it during the learning process. After the T'-th epoch, the parameter maintains a constant β until the end of training. The entire procedure is outlined in Algorithm 1. Additionally, the theoretical generalization bound of the proposed method, based on uniform stability, is provided in Appendix A for completeness.

TABLE 1: Properties of datasets.

Datasets	$dim(\mathcal{S})$	$\lvert\mathcal{S}\rvert$	$L(\mathcal{S})$	$LCard(\mathcal{S})$
bookmark	2150	38912	208	2.03
mediamill	120	41701	101	4.38
eurlex_dc	100	8636	412	1.29
eurlex_sm	100	13270	201	2.21
delicious	500	14784	983	19.02
tmc2007	981	28596	22	2.16
rcv1-s1	944	5815	101	2.88
rcv1-s2	944	5252	101	2.63
rcv1-s3	944	5410	101	2.61
rcv1-s4	944	5761	101	2.48
rcv1-s5	944	5532	101	2.64
VOC2007	3×448×448	5011	20	1.46

5 Experiments

In this section, we conduct a series of experiments to verify the performance of the proposed NMCB. We employ four MLL criteria: *ranking loss*, *one error*, *coverage*, and *average precision*, to evaluate the effectiveness of the methods. A higher value of *average precision* indicates superior performance, while smaller values for the remaining criteria signify better performance. Our experiments are implemented using PyTorch [46] and NVIDIA RTX 3090 Ti. The code of this paper is available at https://github.com/gaoyi439/NMCB.

5.1 Experimental Settings

Datasets & pre-processing. We employ 12 widely-used MLL datasets for experiments[1]. Following previous work [35], [41], [47], for datasets with more than 100 class labels, we filter out rare labels to keep label spaces under 15 and remove instances without relevant labels. Each instance is associated with one complementary label. Properties of each dataset are described through various statistics, including the number of features $dim(\mathcal{S})$, the number of instances $|\mathcal{S}|$, the number of labels $L(\mathcal{S})$, and the average number of labels per instance $LCard(\mathcal{S})$. The detailed descriptions of datasets are provided in Table 1.

1. Publicly available at https://mulan.sourceforge.net/datasets-mlc.html.

TABLE 2: Experimental results (mean±std) on 12 datasets. The best performance for each dataset is shown in **boldface**, where •/○ indicates whether NMCB is superior/inferior to baselines with pairwise t-test (at the 0.05 significance level).

Method	MLL		PML		CLL	MLCLL		NMCB
	LIFT	CCMN	fpml	PARD	L-UW	MAE	GDF	
Coverage↓								
bookmark	.328±.008•	.398±.029•	.474±.018•	.287±.007•	.219±.007	.318±.008•	.221±.006	**.218±.005**
delicious	.703±.004•	.726±.014•	.726±.009•	.697±.014•	.634±.006•	.615±.006•	.615±.007•	**.605±.007**
mediamill	.501±.023•	.568±.067•	.512±.034•	.622±.117•	.483±.033•	.494±.034•	.463±.023•	**.444±.023**
eurlex_dc	.277±.011•	.374±.006•	.441±.035•	.189±.059•	.252±.014•	.411±.008•	.155±.005	**.153±.005**
eurlex_sm	.421±.012•	.555±.008•	.552±.031•	.354±.014•	.423±.008•	.546±.007•	**.308±.008**	.310±.006
tmc2007	.353±.022•	.571±.069•	.510±.013•	.371±.029•	.538±.009•	.497±.009•	.273±.013	**.259±.013**
rcv1-s1	.469±.031•	.444±.022•	.543±.023•	.451±.064•	.366±.043•	.390±.038•	.374±.048•	**.354±.048**
rcv1-s2	.452±.040•	.412±.027•	.546±.020•	.457±.055•	.382±.036•	.355±.010•	.319±.019	**.318±.019**
rcv1-s3	.446±.038•	.406±.055•	.535±.031•	.426±.099•	.384±.021•	.392±.033•	.360±.056•	**.350±.056**
rcv1-s4	.403±.064•	.387±.060•	.512±.042•	.377±.094•	.356±.047•	.378±.033•	.343±.084•	**.329±.084**
rcv1-s5	.428±.019•	.381±.049•	.522±.035•	.413±.058•	.369±.017•	.376±.036•	.347±.041	**.341±.039**
VOC2007	-	.456±.018•	-	-	.507±.025•	.530±.045•	.377±.007	**.370±.004**
Ranking Loss↓								
bookmark	.310±.007•	.387±.032•	.468±.019•	.265±.007•	.196±.007	.301±.008•	.196±.005	**.194±.003**
delicious	.383±.003•	.457±.013•	.438±.008•	.393±.014•	.319±.005•	.305±.004•	**.292±.005**	.293±.005
mediamill	.202±.015•	.271±.038•	.206±.019•	.320±.090•	.193±.020•	.192±.019•	.174±.013•	**.160±.011**
eurlex_dc	.294±.012•	.398±.007•	.470±.037•	.199±.063•	.267±.015•	.437±.008•	.181±.005•	**.161±.006**
eurlex_sm	.333±.013•	.475±.007•	.472±.034•	.252±.011•	.329±.005•	.461±.006•	**.209±.006**	.212±.006
tmc2007	.209±.046•	.401±.040•	.355±.028•	.227±.031•	.391±.020•	.349±.024•	.149±.032•	**.137±.032**
rcv1-s1	.397±.023•	.357±.017•	.474±.019•	.359±.073•	.291±.028	.313±.027•	.305±.055•	**.275±.055**
rcv1-s2	.373±.026•	.328±.033•	.480±.024•	.368±.062•	.302±.021•	.270±.018•	.255±.019•	**.235±.019**
rcv1-s3	.370±.018•	.327±.048•	.477±.019•	.343±.084•	.318±.038•	.322±.058•	.279±.042	**.279±.042**
rcv1-s4	.347±.041•	.324±.042•	.468±.024•	.319±.078•	.293±.024•	.321±.016•	.283±.067•	**.271±.067**
rcv1-s5	.350±.020•	.302±.027•	.456±.017•	.326±.057•	.292±.034•	.295±.020•	.274±.063	**.267±.058**
VOC2007	-	.382±.014•	-	-	.460±.030•	.467±.034•	.311±.008•	**.303±.004**
One Error ↓								
bookmark	.649±.015•	.767±.050•	.885±.019•	.555±.012•	.504±.008•	.613±.011•	.483±.007•	**.478±.006**
delicious	.533±.014•	.618±.029•	.617±.017•	.542±.021•	.482±.018•	.467±.019•	.426±.012	**.426±.012**
mediamill	.187±.019•	.249±.133•	.188±.019•	.309±.214•	.188±.019•	.188±.019•	.184±.018•	**.174±.016**
eurlex_dc	.759±.024•	.924±.009•	.920±.024•	.509±.056•	.609±.024•	.906±.010•	.459±.015•	**.457±.013**
eurlex_sm	.674±.013•	.852±.008•	.868±.032•	.511±.021•	.552±.011•	.834±.008•	**.400±.010**	.415±.020
tmc2007	.465±.089•	.556±.152•	.695±.049•	.486±.077•	.788±.030•	.717±.039•	.359±.094•	**.329±.094**
rcv1-s1	.855±.088•	.667±.045•	.875±.014•	.662±.111•	.718±.024•	.698±.039•	.685±.081•	**.652±.081**
rcv1-s2	.858±.080	.675±.068	.873±.027•	**.626±.085**	.751±.027•	.669±.085	.681±.062•	.671±.062
rcv1-s3	.838±.150•	.662±.112	.879±.035•	**.587±.089**○	.768±.064•	.721±.136•	.715±.109	.695±.109
rcv1-s4	.918±.076•	.656±.086	.894±.019•	**.634±.153**	.703±.056•	.716±.071•	.674±.080•	.648±.080
rcv1-s5	.810±.119•	.724±.075•	.865±.020•	**.627±.103**	.741±.058•	.684±.072•	.681±.101•	.668±.109
VOC2007	-	.596±.000	-	-	.864±.071•	.845±.074•	.636±.000•	**.595±.000**
Average Precision↑								
bookmark	.480±.010•	.379±.030•	.267±.018•	.548±.007•	.604±.005•	.494±.008•	.619±.006•	**.623±.004**
delicious	.511±.004•	.453±.006•	.457±.006•	.504±.006•	.554±.006•	.567±.006•	.586±.004	**.586±.004**
mediamill	.710±.009•	.643±.039•	.709±.012•	.588±.064•	.717±.012•	.715±.012•	.732±.007•	**.748±.006**
eurlex_dc	.429±.016•	.260±.005•	.240±.028•	.614±.055•	.525±.017•	.267±.009•	.656±.010	**.658±.010**
eurlex_sm	.425±.012•	.284±.005•	.285±.026•	.543±.014•	.475±.008•	.288±.003•	**.623±.009**	.615±.012
tmc2007	.535±.063•	.367±.041•	.353±.035•	.524±.049•	.303±.024•	.351±.030•	.652±.064•	**.672±.064**
rcv1-s1	.319±.032•	.417±.025•	.274±.008•	.422±.046•	.428±.019•	.424±.029•	.451±.054•	**.471±.054**
rcv1-s2	.321±.049•	.440±.045•	.276±.025•	.447±.048•	.412±.022•	.469±.044•	.478±.036•	**.488±.036**
rcv1-s3	.356±.057•	.441±.055•	.270±.026•	**.465±.052**	.399±.047•	.418±.088•	.460±.057	.460±.057
rcv1-s4	.329±.031•	.448±.051•	.264±.012•	.452±.112•	.445±.023•	.419±.031•	.485±.080	**.489±.080**
rcv1-s5	.372±.049•	.437±.030•	.283±.013•	.460±.061•	.425±.042•	.452±.043•	.463±.070•	**.471±.070**
VOC2007	-	.389±.011•	-	-	.241±.031•	.264±.037•	.404±.002•	**.431±.001**

Baselines. We compare NMCB with seven other methods and provide a brief introduction to each, grouped by their respective fields:

- **MLL method:** LIFT [5] and CCMN [48] were proposed for MLL, which address MLCLL by treating all non-complementary labels of instances as possible labels.
- **PML method:** We include two PML methods, fpml [37] and PARD [47], as comparison methods, where $(\mathcal{Y} \setminus \bar{y})$ is regarded as candidate labels for learning.
- **CLL method:** L-UW [14] is a CLL method designed for multi-class learning, which is adapted for MLCLL by using the Sigmoid layer and BCE loss to replace the Softmax layer and cross-entropy loss, respectively.
- **MLCLL method:** The comparison method, GDF [7], was designed for the MCLL problem. Following Gao et al., we modify the *Mean Absolute Error* (MAE) loss to make it applicable for MCLL.

Setup. We utilize SGD with a momentum of 0.9 for optimization. The batch size and number of training epochs are set to 256 and 200, respectively. Weight decay is 10^{-3} and the learning rate is selected from $\{10^{-1}, 10^{-2}, 10^{-3}\}$, where the learning rate is multiplied by 0.1 at the 100-th and 150-th epochs [49]. The hyper-

TABLE 3: Ablation study on six datasets. Results are described in the format of mean±std, where the best performance for each dataset is shown in **boldface**.

Datasets	bookmark	mediamill	tmc2007	rcv1-s1	rcv1-s2	rcv1-s3	rcv1-s4	rcv1-s5
				One Error↓				
NMCB	**.478±.006**	**.174±.016**	**.329±.094**	**.652±.081**	**.671±.062**	**.695±.109**	**.648±.080**	**.668±.109**
NMCB w/o label correlation	.488±.006	.183±.016	.335±.094	.672±.081	.681±.062	.712±.109	.660±.08	.669±.109
NMCB w/o mixup & label correlation	.483±.007	.184±.018	.359±.094	.685±.081	.681±.062	.715±.109	.674±.080	.681±.101
				Coverage↓				
NMCB	**.218±.005**	**.444±.023**	**.259±.013**	**.354±.048**	**.318±.019**	**.350±.056**	**.329±.084**	.341±.039
NMCB w/o label correlation	.229±.004	.454±.023	.264±.013	.374±.048	.320±.019	.359±.056	.333±.084	.341±.039
NMCB w/o mixup & label correlation	.221±.006	.463±.023	.273±.013	.374±.048	.319±.019	.360±.056	.343±.084	.347±.041
				Ranking Loss↓				
NMCB	.194±.003	**.160±.011**	**.137±.032**	**.275±.055**	**.235±.019**	.279±.042	**.271±.067**	.267±.058
NMCB w/o label correlation	.194±.003	.168±.011	.147±.032	.295±.055	.249±.019	.288±.042	.285±.067	.267±.058
NMCB w/o mixup & label correlation	.196±.005	.174±.013	.149±.032	.305±.055	.255±.019	.279±.042	.283±.067	.274±.063
				Average Precision↑				
NMCB	**.623±.004**	**.748±.006**	**.672±.064**	**.471±.054**	**.488±.036**	.460±.057	**.489±.080**	**.471±.070**
NMCB w/o label correlation	.620±.004	.730±.006	.660±.064	.450±.054	.475±.036	.460±.057	.485±.080	.461±.071
NMCB w/o mixup & label correlation	.619±.006	.732±.007	.652±.064	.451±.054	.478±.036	.460±.057	.485±.080	.463±.070

TABLE 4: Results of using proposed strategies on comparison methods. Results are described in the format of mean±std, where the best performance for each dataset is shown in **boldface**.

Datasets	bookmark	delicious	mediamill	eurlex_dc	eurlex_sm
			One Error↓		
NMCB	**.478±.006**	**.426±.012**	**.174±.016**	**.457±.013**	**.415±.020**
NMCB w/ L-UW	.781±.007	.481±.007	.192±.017	.916±.008	.858±.009
NMCB w/ MAE	.696±.010	.466±.019	.190±.017	.910±.011	.834±.008
L-UW w/ mixup	.806±.006	.513±.018	.301±.014	.924±.011	.912±.007
MAE w/ mixup	.714±.009	.496±.018	.282±.015	.920±.011	.910±.007
			Coverage↓		
NMCB	**.218±.005**	**.605±.007**	**.444±.023**	**.153±.005**	**.310±.006**
NMCB w/ L-UW	.372±.005	.635±.006	.496±.027	.440±.007	.563±.008
NMCB w/ MAE	.372±.008	.616±.006	.491±.031	.416±.007	.546±.007
L-UW w/ mixup	.416±.004	.667±.006	.578±.025	.445±.007	.576±.009
MAE w/ mixup	.392±.005	.643±.006	.558±.026	.431±.009	.574±.009
			Ranking Loss↓		
NMCB	**.194±.003**	**.293±.005**	**.160±.011**	**.161±.006**	**.212±.006**
NMCB w/ L-UW	.359±.005	.318±.005	.200±.018	.469±.007	.479±.006
NMCB w/ MAE	.358±.008	.305±.004	.196±.020	.442±.008	.461±.006
L-UW w/ mixup	.405±.005	.345±.006	.243±.010	.474±.008	.507±.007
MAE w/ mixup	.380±.005	.326±.005	.224±.011	.460±.009	.505±.007
			Average Precision↑		
NMCB	**.623±.004**	**.586±.004**	**.748±.006**	**.658±.010**	**.615±.012**
NMCB w/ L-UW	.373±.006	.555±.006	.712±.010	.245±.008	.274±.002
NMCB w/ MAE	.424±.007	.567±.006	.714±.011	.264±.009	.288±.003
L-UW w/ mixup	.340±.004	.531±.007	.663±.007	.237±.009	.245±.006
MAE w/ mixup	.404±.006	.545±.006	.683±.007	.249±.010	.247±.005

parameters in Eq. (8) are set as $T' = 100$ and $\beta = 1$, with a mixing level of $\alpha = 0.9$. The label correlation extraction model is a linear model. As the VOC2007 dataset consists of color images with a size of $3 \times 448 \times 448$, we employ an 18-layer ResNet as the classification model. For other datasets, a linear model is adopted. The methods are evaluated over 5 trials on the VOC2007 dataset, while the remaining datasets employ ten-fold cross-validation. During the learning process, the training data includes only complementary labels and the test data associated with relevant labels is used for evaluation. Mean and *standard deviation* (std) of four criteria are reported, where $\downarrow$ / $\uparrow$ indicates that a smaller/higher criterion value signifies better method performance.

5.2 Empirical Results

Table 2 presents the results of four criteria for various methods across 12 datasets, where results of LIFT, fpml and PARD are marked as "-" on the VOC2007 dataset since they cannot directly handle raw image data. As depicted in Table 2, NMCB outperforms most methods across the 12 datasets. Specifically, we observe performance enhancements over the best baseline on tmc2007, with improvements of 0.03 and 0.02 on the *one error* and *average precision*, respectively. This indicates the effective-

TABLE 5: Results (mean±std) with different trade-off parameter β. The best performance for each dataset is highlighted in **boldface**, where $\bullet/\circ$ indicates whether NMCB is superior/inferior to other results with Wilcoxon signed-rank test (at the 0.05 significant level). "Fixed" means that the value of the trade-off parameter β remains constant from the beginning to the end of the training, while "Dynamic" denotes that β follows the dynamic trade-off parameter strategy, shown in Eq. (8).

β	Fixed					Dynamic
	0.1	0 .3	0 .5	0.8	1	1
			One Error↓			
bookmark	.483±.006	.486±.006•	.481±.006	.487±.006•	.485±.005	**.478±.006**
delicious	.466±.012•	.444±.011	.438±.013•	.464±.012•	.449±.013•	**.426±.012**
mediamill	.210±.016	.202±.016	.206±.013	.207±.016	.207±.016	**.174±.016**
eurlex_dc	.476±.016•	.478±.013	.473±.018	.464±.013	.462±.014	**.457±.013**
eurlex_sm	**.410±.018**	.420±.028	.415±.020	.417±.027	.434±.018•	.415±.020
tmc2007	.334±.094	.348±.094	.347±.094	**.311±.094**○	.321±.094	.329±.094
			Coverage↓			
bookmark	.213±.004	.213±.004	.217±.004	.213±.005	.213±.005	.218±.005
delicious	.625±.007•	.618±.007•	.608±.007	.626±.007•	.621±.007•	**.605±.007**
mediamill	.458±.023•	**.429±.023**○	.443±.023	.446±.023	.450±.023	.444±.023
eurlex_dc	.158±.004	.155±.005•	.160±.004	.163±.005•	.158±.005	**.153±.005**
eurlex_sm	.315±.009	.317±.011•	.315±.008	**.309±.008**	.326±.008	.310±.006
tmc2007	.315±.013	.321±.013•	.320±.013•	.303±.013•	.308±.013•	**.259±.013**
			Ranking Loss↓			
bookmark	.191±.003	.191±.003	.195±.003•	.191±.003	.198±.003	.194±.003
delicious	.316±.005	.306±.005	.295±.005	.317±.005	.310±.005	**.293±.005**
mediamill	.172±.011	**.154±.011**	.162±.011	.164±.011•	.166±.010•	.160±.011
eurlex_dc	.166±.004	.163±.006•	.168±.004	.171±.005•	.165±.005	**.161±.006**
eurlex_sm	.214±.007	.215±.012	.215±.008	.222±.008•	.226±.007	**.212±.006**
tmc2007	.163±.032•	.168±.032•	.167±.032•	.152±.032•	.156±.033•	**.137±.032**
			Average Precision↑			
bookmark	.620±.004	.619±.004	.619±.004•	.618±.004•	.615±.004•	**.623±.004**
delicious	.561±.004•	.571±.004•	.579±.004•	.560±.004•	.567±.004•	**.586±.004**
mediamill	.733±.006	**.750±.006**○	.743±.006	.741±.006	.739±.006	.748±.006
eurlex_dc	.646±.011	.646±.011•	.644±.012•	.644±.009	.652±.010	**.658±.010**
eurlex_sm	.615±.012	.611±.019	.611±.014	.602±.015•	.599±.012	**.615±.012**
tmc2007	.639±.064	.628±.064•	.630±.064•	.660±.064•	.652±.065•	**.672±.064**

ness of our method, NMCB, in solving the MLCLL problem. Furthermore, NMCB surpasses LIFT and CCMN on four criteria across all datasets, which demonstrates its suitability over MLL methods for solving the MLCLL problem. Compared with PML methods, the *average precision* of NMCB is 0.418 higher than that of fpml on the eurlex_dc dataset, which indicates the effectiveness of NMCB in MLCLL scenarios with dense noisy labels.

As evident in Table 2, NMCB outperforms L-UW across all datasets, which suggests that CLL methods may not adequately tackle the challenges posed by the MLCLL problem. This limitation arises from their reliance on the assumption of one relevant label per instance, while the actual number of relevant labels per instance remains unknown in MLL. This lack of information hinders CLL methods from capturing relationships among multiple relevant labels and complementary labels in the problem of MLCLL. Additionally, our method achieves comparable performance with state-of-the-art MLCLL methods, which proves the effectiveness of a tailored version of mixup to reduce the sensitivity of the model on noisy labels and achieve smoother decision boundaries. Moreover, this also demonstrates that the strategy of using consistency regularization to emphasize the alignment of the model's outputs with label correlations positively contributes to improving the performance of NMCB.

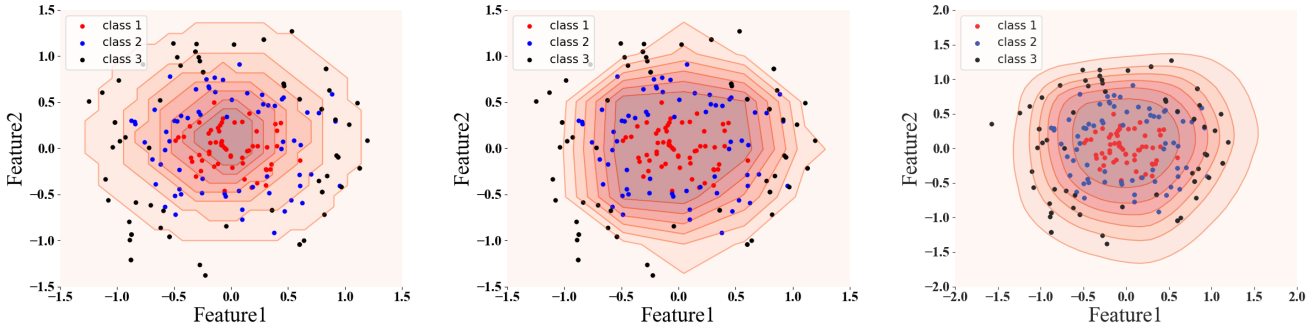
5.3 Ablation Studies

The effect of proposed strategies. To validate the contributions of the two main strategies of NMCB, namely the mixup strategy to alleviate the problem of density noisy labels in MLCLL and the label correlation extraction model to automatically explore relationships among labels, we compare NMCB with two variants: (1) *NMCB w/o label correlation*: This variant removes the label correlation extraction model from NMCB, which uses the loss $\bar{L}(\boldsymbol{f}(\boldsymbol{x}), \bar{\boldsymbol{y}}) = \bar{L}_{\mathrm{GDF}}(\boldsymbol{f}(\boldsymbol{x}), \bar{\boldsymbol{y}}) + \psi(t)\|\boldsymbol{f}(\boldsymbol{x}') - \hat{\boldsymbol{y}}'\|_2^2$ to learn from complementary labeled data. (2) *NMCB w/o mixup & label correlation*: This variant removes both the mixup and label correlation strategies proposed by us, which only adopts the GDF loss to train a classifier (i.e., this variant will become GDF). These variants allow us to isolate and evaluate the individual impacts of the mixup strategy and the label correlation extraction model on the overall performance of NMCB.

From Table 3, our proposed method outperforms the two variants in most cases. Specifically, NMCB achieves comparable or superior performance compared to variant (1) across all settings. In variant (1), using the non-complementary label vector $\hat{\boldsymbol{y}}'$ in mixup to replace the label correlation vector $\boldsymbol{c}(\hat{\boldsymbol{y}}')$ leads to performance differences. This indicates that label correlations are effective in making the alignment objective in the regularization term clearer, and encourage the classification model to consider label correlations during the learning process. Moreover, the results of variant (2) on four criteria across all datasets are inferior to those of variant (1), which demonstrates that the mixup strategy effectively alleviates the problem of noisy labels in the MLCLL scenario and further enhances the model's performance.

TABLE 6: Results (mean±std) with various mixing levels α. The best performance for each dataset is shown in **boldface**.

α	0.2	0.4	0.6	0.9	1
One Error↓					
bookmark	.481±.006	.480±.006	.480±.006	**.478±.006**	.479±.006
delicious	.427±.013	.426±.013	.434±.012	**.426±.012**	.434±.012
mediamill	.189±.016	.189±.016	.187±.016	**.174±.016**	.174±.016
eurlex_dc	.459±.014	.458±.016	.458±.013	**.457±.013**	.457±.015
eurlex_sm	.417±.017	.415±.019	**.405±.019**	.415±.020	.425±.016
tmc2007	.322±.094	.320±.094	**.317±.094**	.329±.094	.338±.094
Coverage↓					
bookmark	.221±.005	.221±.005	.221±.005	**.218±.005**	.221±.005
delicious	.614±.007	.614±.007	.613±.007	**.605±.007**	.612±.007
mediamill	.484±.023	.484±.023	.430±.023	**.444±.023**	.445±.023
eurlex_dc	.154±.005	.153±.005	**.152±.005**	.153±.005	.155±.007
eurlex_sm	.310±.008	.310±.009	.308±.009	.310±.006	**.299±.010**
tmc2007	.276±.013	.274±.013	.272±.013	**.259±.013**	.261±.013
Ranking Loss↓					
bookmark	.195±.003	.195±.003	.195±.003	**.194±.003**	.195±.003
delicious	.303±.005	.303±.005	.302±.005	**.293±.005**	.300±.005
mediamill	.178±.011	.178±.011	.150±.011	**.160±.011**	.160±.011
eurlex_dc	.162±.005	.161±.005	**.160±.005**	.161±.006	.163±.007
eurlex_sm	.213±.007	.212±.008	**.210±.007**	.212±.006	.229±.008
tmc2007	.143±.032	.142±.032	.140±.032	**.137±.032**	.137±.032
Average Precision↑					
bookmark	.621±.004	.621±.004	.621±.004	**.623±.004**	.621±.004
delicious	.577±.004	.577±.004	.578±.004	**.586±.004**	.579±.004
mediamill	.737±.006	.737±.006	.752±.006	**.748±.006**	.748±.006
eurlex_dc	.657±.010	.657±.012	.658±.010	**.658±.010**	.657±.011
eurlex_sm	.613±.011	.613±.013	.621±.013	**.615±.012**	.608±.012
tmc2007	.661±.064	.664±.064	.668±.064	**.672±.064**	.670±.064



(a) CCMN (b) GDF (c) NMCB

Fig. 3: Decision Boundaries for various classifiers trained with different methods based on a toy dataset. This toy dataset consists of three different classes (see Appendix B.2 for details).

Extended comparison. To further validate the effectiveness of the proposed method NMCB, we design four variants by decomposing and reassembling its core components into existing MLCLL methods. Specifically, we consider the following settings: (a) *NMCB w/ L-UW*: Replacing the GDF loss in NMCB with L-UW. (b) *NMCB w/ MAE*: Replacing GDF with MAE (i.e., the revised MAE loss introduced in the baselines). (c) *L-UW w/ mixup*: Applying only the tailored mixup strategy to L-UW. (d) *MAE w/ mixup*: Applying only the tailored mixup strategy to MAE. As shown in Table 4, variants (a) and (b) outperform (c) and (d), respectively, which illustrates that the proposed consistency regularization term contributes positively to performance improvements. Moreover, NMCB consistently achieves the best results across all settings, which indicates that the GDF loss is particularly well-aligned with our framework. These findings highlight the importance of the coordinated integration of mixup, label correlation extraction, and consistency regularization within NMCB, which work synergistically to effectively address the challenges of MLCLL.

5.4 Effect of NMCB with Different Parameters

Different trade-off parameter. Here, we investigate the impact of varying the trade-off parameter on NMCB performance, and the results of NMCB with different β values are reported in Table 5. In the Table 5, "Fixed" means that the value of the

TABLE 7: Results (mean±std) with varying layer depths in the extraction model.

Datasets	Label correlation extraction model		
	w/ 1-layer	w/ 2-layer	w/ 3-layer
One Error↓			
bookmark	.478±.006	.478±.006	.478±.005
delicious	.426±.012	.426±.013	.426±.012
mediamill	.174±.016	.174±.016	.174±.016
eurlex_dc	.457±.013	.457±.015	.456±.013
eurlex_sm	.415±.020	.406±.020	.413±.025
Coverage↓			
bookmark	.218±.005	.218±.004	.218±.005
delicious	.605±.007	.605±.007	.605±.007
mediamill	.444±.023	.444±.023	.444±.023
eurlex_dc	.153±.005	.154±.005	.155±.007
eurlex_sm	.310±.006	.309±.008	.309±.011
Ranking Loss↓			
bookmark	.194±.003	.194±.003	.195±.003
delicious	.293±.005	.293±.005	.293±.005
mediamill	.160±.011	.160±.011	.160±.011
eurlex_dc	.161±.006	.162±.005	.163±.007
eurlex_sm	.212±.006	.210±.007	.211±.01
Average Precision↑			
bookmark	.623±.004	.623±.004	.623±.004
delicious	.586±.004	.586±.004	.586±.004
mediamill	.748±.006	.748±.006	.748±.006
eurlex_dc	.658±.010	.657±.011	.657±.011
eurlex_sm	.615±.012	.619±.014	.615±.019

trade-off parameter β remains constant from the beginning to the end of the training, while "Dynamic" denotes that β follows the dynamic trade-off parameter strategy (shown in Eq. (8)). For the "Fixed" experimental part, we provide five different values of β chosen from the set $\{0.1, 0.3, 0.5, 0.8, 1\}$. Observing the results, we find that the best performance over all four criteria is achieved with dynamic trade-off parameter β on almost six datasets. This suggests that progressively strengthening the regularization implemented by the dynamic trade-off parameter strategy is effective in improving model performance. From the results of the fixed β, whether the strength of regularization is kept consistently small or large during the learning process, the benefits for model learning are not as pronounced. While the dynamic strategy does not always yield statistically significant improvements, it demonstrates more robust performance overall by eliminating the need for manual tuning of the trade-off parameter β.

Mixing level α. We subsequently delve into an analysis of the impact of varying mixing levels, denoted by the parameter α, on our proposed method NMCB across four criteria. Here, we adopt the dynamic trade-off parameter for the regularization term in experiments (shown as Eq. (7)). Our investigation contains six diverse datasets, with mixing levels set at 0.2, 0.4, 0.6, and 0.9, respectively. The experimental results, as detailed in Table 6, reveal that criteria such as *one error*, *coverage*, *ranking loss*, and *average precision* display slightly superior performance at a mixing level of 0.9 compared to other α values. Despite this marginal discrepancy, it is noteworthy that the differences in results across varying α values are not substantial. This observation demonstrates the robustness of our proposed method, which indicates its ability to maintain consistent performance even in the face of parameter variations. Consequently, we adopt a mixing level of $\alpha = 0.9$ as the standard configuration for our experiments, as it showcases commendable performance.

Layer depth in label correlation extraction model. To investigate the impact of model depth on label correlation extraction, we compare the original architecture used in Eq. (4) with deeper variants incorporating two and three fully connected layers. The two-layer variant consists of a linear layer followed by a ReLU activation with 1000 units, whereas the three-layer variant adds another linear-ReLU block. All versions are trained with the same experimental settings and a linear model for prediction. The results, reported in Table 7, indicate that increasing model depth does not consistently improve performance. In most cases, the one-layer model performs comparably to the deeper models on all datasets across various metrics. These findings empirically support our choice of using a linear model for label correlation extraction in the proposed NMCB framework. The linear extractor is simple and computationally efficient, while remaining effective in capturing the pairwise correlations essential for MLL.

5.5 Exploration of Decision Boundaries

To further explore the sensitivity and robustness of different methods to noisy labels, we design a classification task with three possible labels to closely examine and visualize the decision boundaries produced by each method. For this task, we generate a toy dataset with 2-dimensional features, where the training data is annotated with complementary labels, and the test data is associated with the relevant labels. To ensure a fair and direct comparison, we adopt the same settings for both our method (NMCB) and the baselines, CCMN and GDF. Detailed experimental setups and parameter configurations are provided in Appendix B.2.

As illustrated in Fig. 3, the decision boundaries of NMCB are smoother than those generated by other baseline methods, demonstrating that our method is less sensitive to noisy labels. The improved smoothness in decision boundaries suggests that NMCB effectively mitigates the disruptive impact of noisy labels on the learning process. By contrast, the less smooth boundaries observed in the previous methods indicate suboptimal performance when handling noisy labels, which compromises their generalization capabilities. These results emphasize the robustness of our proposed method, which not only demonstrates enhanced resilience to label noise but also delivers superior generalization performance compared to both CCMN and GDF. Specifically, our method surpasses GDF in achieving smoother and more accurate decision boundaries, an advantage attributed to our innovative strategy of consistency regularization with label correlations on data augment transformed by a tailored version of mixup, as outlined in Eq. (5). This approach enforces stronger consistency across labels, promoting robustness by encouraging the model to learn shared label structures, thereby improving classification performance in the presence of noise.

6 Conclusion

Due to the memorization effect of neural networks and the presence of noisy labels in MLCLL, previous methods may fit noisy labels and lead to a sharp decision boundary. In this paper, we propose a novel framework called NMCB to solve the MLCLL problem. To alleviate the impact of noise, NMCB pioneers a tailored version of mixup for MLCLL to aid in achieving a smoother decision boundary, which helps reduce sensitivity to

noisy labels and improve generalization performance. Since non-complementary labels transformed by the tailored version of mixup exhibit smoother characteristics, NMCB employs a model to automatically extract label correlations from these labels to avoid distortion in the label relationships of MLCLL caused by noisy labels. Accordingly, NMCB introduces a consistency regularization term that aligns the model's output with the extracted label correlations, further improving performance. We also establish a generalization bound for our method theoretically. Empirical studies demonstrate the effectiveness of the proposed method.

While the proposed NMCB framework demonstrates its effectiveness across various domains, the mixup strategy is inherently better suited to continuous input spaces (e.g., images), where convex combinations preserve semantic meaning. For discrete domains like natural language processing, directly applying mixup to raw discrete inputs (e.g., word sequences) may lead to semantically invalid data. In our work, although some datasets originate from the textual domain, all textual data are preprocessed into numerical vector representations, which makes our method feasible. Nevertheless, this reliance on feature-level representations may limit NMCB's use in tasks requiring raw discrete input processing. Future work could explore mixup-compatible augmentation strategies tailored for discrete domains to broaden its applicability.

REFERENCES

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

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

[3] Y. Gao, M. Xu, and M.-L. Zhang, "Complementary to multiple labels: A correlation-aware correction approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 9179–9191, 2024.

[4] Y.-F. Zhang and M.-L. Zhang, "Generalization analysis for label-specific representation learning," in *Advances in Neural Information Processing Systems 38*, Vancouver, Canada, 2024, pp. 104 904–104 933.

[5] M.-L. Zhang and L. Wu, "Lift: Multi-label learning with label-specific features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 107–120, 2015.

[6] Y.-F. Zhang and M.-L. Zhang, "Generalization analysis for multi-label learning," in *Proceedings of 41st International Conference on Machine Learning*, Vienna, Austria, 2024, pp. 60 220–60 243.

[7] Y. Gao, M. Xu, and M.-L. Zhang, "Unbiased risk estimator to multi-labeled complementary label learning," in *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, Macao, China, 2023, pp. 3732–3740.

[8] Y. Gao, J.-Y. Zhu, M. Xu, and M.-L. Zhang, "Multi-label learning with multiple complementary labels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 9, pp. 8013–8024, 2025.

[9] T. Ishida, G. Niu, W.-H. Hu, and M. Sugiyama, "Learning from complementary labels," in *Advances in Neural Information Processing Systems 30*, Long Beach, CA, 2017, pp. 5639–5649.

[10] T. Ishida, G. Niu, A. K. Menon, and M. Sugiyama, "Complementary-label learning for arbitrary losses and models," in *Proceedings of the 36th International Conference on Machine Learning*, Long Beach, CA, 2019, pp. 2971–2980.

[11] X.-Y. Yu, T.-L. Liu, M.-M. Gong, and D.-C. Tao, "Learning with biased complementary labels," in *Proceedings of the 15th European Conference on Computer Vision*, Munich, Germany, 2018, pp. 69–85.

[12] Y.-W. Xu, M.-M. Gong, J.-X. Chen, T.-L. Liu, K. Zhang, and K. Batmanghelich, "Generative-discriminative complementary learning," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, NY, 2020, pp. 6526–6533.

[13] H.-R. Jiang, Z.-H. Sun, and Y.-J. Tian, "Comco: Complementary supervised contrastive learning for complementary label learning," *Neural Networks*, vol. 169, pp. 44–56, 2024.

[14] Y. Gao and M.-L. Zhang, "Discriminative complementary-label learning with weighted loss," in *Proceedings of the 38th International Conference on Machine Learning*, Virtual Event, 2021, pp. 3587–3597.

[15] L. Feng, T. Kaneko, B. Han, G. Niu, B. An, and M. Sugiyama, "Learning with multiple complementary labels," in *Proceedings of the 37th International Conference on Machine Learning*, Virtual Event, 2020, pp. 3072–3081.

[16] Y. Shi, N. Xu, H. Yuan, and X. Geng, "Unreliable partial label learning with recursive separation," in *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, Macao, China, 2023, pp. 4208–4216.

[17] Y.-B. Bai, E. Yang, B. Han, Y.-H. Yang, J.-T. Li, Y.-N. Mao, G. Niu, and T.-L. Liu, "Understanding and improving early stopping for learning with noisy labels," in *Advances in Neural Information Processing Systems 34*, Virtual Event, 2021, pp. 24 392–24 403.

[18] S. H. Lim, N. B. Erichson, F. Utrera, W.-N. Xu, and M. W. Mahoney, "Noisy feature mixup," in *Proceedings of the 10th International Conference on Learning Representations*, Virtual Event, 2022.

[19] L. Carratino, M. Cissé, R. Jenatton, and J.-P. Vert, "On mixup regularization," *Journal of machine learning research*, vol. 23, pp. 1–31, 2022.

[20] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, Long Beach, CA, 2019, pp. 6438–6447.

[21] P. Bartlett and J. Shawe-Taylor, "Generalization performance of support vector machines and other pattern classifiers," *Advances in Kernel Methods—Support Vector Learning*, pp. 43–54, 1999.

[22] W. S. Lee, P. L. Bartlett, and R. C. Williamson, "Lower bounds on the VC dimension of smoothly parameterized function classes," *Neural Computation*, vol. 7, no. 5, pp. 1040–1053, 1995.

[23] H.-Y. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada, 2018.

[24] M.-L. Zhang, Y.-K. Li, X.-Y. Liu, and X. Geng, "Binary relevance for multi-label learning: an overview," *Frontiers of Computer Science*, vol. 12, no. 2, pp. 191–202, 2018.

[25] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Machine Learning*, vol. 73, no. 2, pp. 133–153, 2008.

[26] Y.-C. Li, Y. Song, and J.-B. Luo, "Improving pairwise ranking for multi-label image classification," in *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 2017, pp. 1837–1845.

[27] S. Burkhardt and S. Kramer, "Online multi-label dependency topic models for text classification," *Machine Learning*, vol. 107, no. 5, pp. 859–886, 2018.

[28] W. Gerych, T. Hartvigsen, L. Buquicchio, E. Agu, and E. A. Rundensteiner, "Recurrent bayesian classifier chains for exact multi-label classification," in *Advances in Neural Information Processing Systems 34*, Virtual Event, 2021, pp. 15 981–15 992.

[29] W.-T. Zhao, S.-F. Kong, J.-W. Bai, D. Fink, and C. P. Gomes, "HOT-VAE: learning high-order label correlation for multi-label classification via attention-based variational autoencoders," in *Proceedings of 35th AAAI Conference on Artificial Intelligence*, Virtual Event, 2021, pp. 15 016–15 024.

[30] N. Xu, Y.-P. Liu, and X. Geng, "Partial multi-label learning with label distribution," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, NY, 2020, pp. 6510–6517.

[31] L.-C. Wang, Z.-M. Ding, S.-J. Han, J.-J. Han, C. Choi, and Y. Fu, "Generative correlation discovery network for multi-label learning," in *Proceedings of 2019 IEEE International Conference on Data Mining*, Beijing, China, 2019, pp. 588–597.

[32] G.-X. Xun, K. Jha, J.-H. Sun, and A.-D. Zhang, "Correlation networks for extreme multi-label text classification," in *Proceedings of 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Virtual Event, 2020, pp. 1074–1082.

[33] L. Sun, S. Feng, J. Liu, G. Lyu, and C. Lang, "Global-local label correlation for partial multi-label learning," *IEEE Transactions on Multimedia*, vol. 24, no. 99, pp. 581–593, 2021.

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

[35] M.-K. Xie and S.-J. Huang, "Partial multi-label learning," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, New Orleans, LA, 2018, pp. 4302–4309.

[36] G. Lyu, S.-H. Feng, and Y.-D. Li, "Noisy label tolerance: A new perspective of partial multi-label learning," *Information Sciences*, vol. 543, pp. 454–466, 2021.

[37] G.-X. Yu, X. Chen, C. Domeniconi, J. Wang, Z. Li, Z.-L. Zhang, and X.-D. Wu, "Feature-induced partial multi-label learning," in *Proceedings of 2018 IEEE International Conference on Data Mining*, Singapore, 2018, pp. 1398–1403.

[38] L.-J. Sun, S.-H. Feng, T. Wang, C.-Y. Lang, and Y. Jin, "Partial multi-label learning by low-rank and sparse decomposition," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, Honolulu, HI, 2019, pp. 5016–5023.

[39] J.-P. Fang and M.-L. Zhang, "Partial multi-label learning via credible label elicitation," in *Proceedings of 33rd AAAI Conference on Artificial Intelligence*, Honolulu, HI, 2019, pp. 3518–3525.

[40] H.-B. Wang, W.-W. Liu, Y. Zhao, C. Zhang, T. Hu, and G. Chen, "Discriminative and correlative partial multi-label learning," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, Macao, China, 2019, pp. 3691–3697.

[41] M.-K. Xie and S.-J. Huang, "Partial multi-label learning with noisy label identification," in *Proceedings of 34th AAAI Conference on Artificial Intelligence*, New York, NY, 2020, pp. 6454–6461.

[42] C. Li, X. Li, L. Feng, and J. Ouyang, "Who is your right mixup partner in positive and unlabeled learning," in *Proceedings of the 10th International Conference on Learning Representations*, Virtual Event, 2022.

[43] W.-I. Lin, G. Niu, H.-T. Lin, and M. Sugiyama, "Enhancing label sharing efficiency in complementary-label learning with label augmentation," *CoRR*, vol. abs/2305.08344, 2023.

[44] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[45] S.-J. Huang and Z.-H. Zhou, "Multi-label learning by exploiting label correlations locally," in *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, J. Hoffmann and B. Selman, Eds., Toronto, Canada, 2012, pp. 949–955.

[46] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z.-M. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J.-J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, Vancouver, Canada, 2019, pp. 8024–8035.

[47] J.-Y. Hang and M.-L. Zhang, "Partial multi-label learning with probabilistic graphical disambiguation," in *Advances in Neural Information Processing Systems 36*, New Orleans, LA, 2023.

[48] M.-K. Xie and S.-J. Huang, "CCMN: A general framework for learning with class-conditional multi-label noise," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 154–166, 2023.

[49] Z.-R. Wu, Y.-J. Xiong, S. X. Yu, and D.-H. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018, pp. 3733–3742.

Yi Gao received the PhD degree from Southeast University, China, in 2025. She obtained the BSc and MSc degrees in computer science from Northwest University, China, in 2017 and 2020 respectively. Currently, she is an assistant researcher at the School of Computer Science and Engineering, Southeast University, China. Her main research interests include machine learning and data mining, with a focus on learning from complementary labels.

Yuan-Yuan Meng received the BSc degree in computer science from Southeast University, China, in 2019. She is currently pursuing the MSc degree in Southeast University. Her main research interests include machine learning and data mining, with a focus on learning from complementary labels.

Miao Xu is a senior lecturer in the School of Electrical Engineering and Computer Science at the University of Queensland, Australia. She was awarded the Australian Research Council Discovery Early Career Researcher Award (DECRA) in 2023. Dr Xu specializes in machine learning and data mining, particularly focusing on the challenges of learning from imperfect information. Dr Xu earned a PhD from Nanjing University, where research efforts led to notable recognitions including the CAAI Outstanding Doctoral Dissertation Award.

Min-Ling Zhang received the BSc, MSc, and PhD degrees in computer science from Nanjing University, China, in 2001, 2004 and 2007, respectively. Currently, he is a Professor at the School of Computer Science and Engineering, Southeast University, China. His main research interests include machine learning and data mining. In recent years, Dr. Zhang has served as the General Co-Chairs of ACML'18, Program Co-Chairs of CCML'25, PAKDD'19, CCF-ICAI'19, ACML'17, CCFAI'17, PRICAI'16, Senior PC member or Area Chair of KDD 2021-2024, AAAI 2022-2025, IJCAI 2017-2024, ICML 2024, ICLR 2024, etc. He is also on the editorial board of IEEE Transactions on Pattern Analysis and Machine Intelligence, Science China Information Sciences, ACM Transactions on Intelligent Systems and Technology, Frontiers of Computer Science, Machine Intelligence Research, etc. Dr. Zhang is the Steering Committee Member of ACML and PAKDD, Vice-Chair of the CAAI (Chinese Association of Artificial Intelligence) Machine Learning Society. He is a Distinguished Member of CCF, CAAI, and Senior Member of AAAI, ACM, IEEE.

APPENDIX A
THEORETICAL ANALYSIS: GENERALIZATION BOUND

Generalization refers to a model's ability to perform well on unseen instances. Here, we establish a generalization bound for the proposed method based on *uniform stability* to investigate the generalization of NMCB. Let the expected risk of MLCLL in $\bar{L}$ over $\bar{p}(\boldsymbol{x}, \bar{y})$ be denoted as $\bar{R}(\boldsymbol{f}) = \mathbb{E}_{\bar{p}(\boldsymbol{x},\bar{y})}[\bar{L}(\boldsymbol{f}(\boldsymbol{x}), \bar{\boldsymbol{y}})]$, which usually is approximated by its empirical risk $\bar{R}_n(\boldsymbol{f}) = \frac{1}{n}\sum_{i=1}^{n} \bar{L}(\boldsymbol{f}(\boldsymbol{x}_i), \bar{\boldsymbol{y}}_i)$. Here, $\bar{D}^{i,\bar{\boldsymbol{z}}'}$ represents the dataset obtained by replacing the i-th instance $\bar{\boldsymbol{z}}_i = (\boldsymbol{x}_i, \bar{y}_i)$ in $\bar{D}$ with $\bar{\boldsymbol{z}}' = (\boldsymbol{x}', \bar{y}')$. We start investigating a generalization bound in Theorem 1 based on Definition 1.

Definition 1. *(Uniform Stability) For $\bar{D}$ and $\bar{\boldsymbol{z}} = (\boldsymbol{x}, \bar{y}), \bar{\boldsymbol{z}}' = (\boldsymbol{x}', \bar{y}') \in \mathcal{X} \times \mathcal{Y}$, the learning method is uniform stability with respect to $\bar{L}(\boldsymbol{x}, \bar{\boldsymbol{y}})$ if there exists $\xi \geq 0$ such that the following conditions hold:*

$$|\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}}) - \bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}'}}, \bar{\boldsymbol{z}})| \leq \xi,$$

where $\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}}) = \bar{L}(\boldsymbol{f}_{\bar{D}}(\boldsymbol{x}), \bar{\boldsymbol{y}})$ and $\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}'}}, \bar{\boldsymbol{z}}) = \bar{L}(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}'}}(\boldsymbol{x}), \bar{\boldsymbol{y}})$, in which $\boldsymbol{f}_{\bar{D}}$ and $\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}'}}$ refer to multi-labeled classifiers trained from $\bar{D}$ and $\bar{D}^{i,\bar{\boldsymbol{z}}'}$, respectively.

This definition is valid since $\bar{L}_{\mathrm{GDF}}$ has an upper bound according to [7], and $0 \leq \Phi(\boldsymbol{x}, \bar{\boldsymbol{y}}) \leq K$. Building upon this definition, we derive a generalization bound for our method in Theorem 1. Before stating the theorem, we first justify the assumption of the upper bound constant $M = \sup_{\boldsymbol{z} \in \mathcal{X}\times\mathcal{Y}} \bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}})$, which ensures the validity of Theorem 1. Specifically, the model outputs $\boldsymbol{f}(\cdot)$ are passed through a Sigmoid layer and constrained within $[\varepsilon, 1-\varepsilon]$ (e.g., $\varepsilon = 10^{-6}$) for numerical stability. This bounds each term in the GDF loss by $\log(1/\varepsilon)$, and the total GDF loss across K classes satisfies:

$$\bar{L}_{\mathrm{GDF}} \leq K \cdot \log(1/\varepsilon).$$

As $\boldsymbol{f}(\boldsymbol{x}'), \boldsymbol{c}(\hat{\boldsymbol{y}}') \in [0,1]^K$, we have $\Phi(\boldsymbol{x}, \bar{\boldsymbol{y}}) \leq K$. Therefore, the total loss satisfies $M \leq K \cdot \log(1/\varepsilon) + K$, which is independent of n. This upper bound ensures that the constant M is finite, and it generally holds in practice when using standard neural network architectures with output constraints (e.g., Sigmoid activations) and numerically stabilized loss functions.

Theorem 1. *Suppose $M = \sup_{\boldsymbol{z} \in \mathcal{X}\times\mathcal{Y}} \bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}})$ and the learning method is uniform stability with respect to $\bar{L}(\boldsymbol{x}, \bar{\boldsymbol{y}})$. For any $\delta \in (0,1)$, with a probability at least $1-\delta$, the following bound holds:*

$$\bar{R}(\boldsymbol{f}_{\bar{D}}) - \bar{R}_n(\boldsymbol{f}_{\bar{D}}) \leq \xi + (2n\xi + M)\sqrt{\frac{\ln(1/\delta)}{2n}}.$$

Proof. Let $\Upsilon(\bar{D}) = \bar{R}(\boldsymbol{f}_{\bar{D}}) - \bar{R}_n(\boldsymbol{f}_{\bar{D}})$. For any $i \in [n] = \{1, 2, \ldots, n\}$, we have

$$\begin{aligned}
\mathbb{E}_{\bar{D}}[\Upsilon(\bar{D})] &= \mathbb{E}_{\bar{D}}[\bar{R}(\boldsymbol{f}_{\bar{D}}) - \bar{R}_n(\boldsymbol{f}_{\bar{D}})] \\
&= \mathbb{E}_{\bar{D}}[\bar{R}(\boldsymbol{f}_{\bar{D}})] - \mathbb{E}_{\bar{D}}[\bar{R}_n(\boldsymbol{f}_{\bar{D}})] \\
&= \mathbb{E}_{\bar{D},\bar{\boldsymbol{z}}}[\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}})] - \frac{1}{n}\sum_{j=1}^{n} \mathbb{E}_{\bar{D}}[\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}}_j)] \\
&= \mathbb{E}_{\bar{D},\bar{\boldsymbol{z}}_i'}[\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}}_i')] - \mathbb{E}_{\bar{D}}[\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}}_i)] \\
&= \mathbb{E}_{\bar{D},\bar{\boldsymbol{z}}_i'}[\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}}_i')] - \mathbb{E}_{\bar{D},\bar{\boldsymbol{z}}_i'}[\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}}, \bar{\boldsymbol{z}}_i')] \quad (\bar{\boldsymbol{z}}_i \text{ is replaced by } \bar{\boldsymbol{z}}_i') \\
&= \mathbb{E}_{\bar{D},\bar{\boldsymbol{z}}_i'}[\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}}_i') - \bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}}, \bar{\boldsymbol{z}}_i')] \leq \xi. \quad \text{(According to Definition 1)}
\end{aligned}$$

Given $\bar{\boldsymbol{z}}_i' \in \mathcal{X} \times \mathcal{Y}$, the following equation satisfies

$$\begin{aligned}
\left|\Upsilon(\bar{D}) - \Upsilon(\bar{D}^{i,\bar{\boldsymbol{z}}_i'})\right| &= \left|\bar{R}(\boldsymbol{f}_{\bar{D}}) - \bar{R}_n(\boldsymbol{f}_{\bar{D}}) - \bar{R}(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}}) + \bar{R}_n(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}})\right| \\
&\leq \left|\bar{R}(\boldsymbol{f}_{\bar{D}}) - \bar{R}(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}})\right| + \left|\bar{R}_n(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}}) - \bar{R}_n(\boldsymbol{f}_{\bar{D}})\right|.
\end{aligned}$$

Due to the learning method has uniform stability, the above inequality can be further solved. For $\left|\bar{R}(\boldsymbol{f}_{\bar{D}}) - \bar{R}(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}})\right|$, we can obtain

$$\begin{aligned}
\left|\bar{R}(\boldsymbol{f}_{\bar{D}}) - \bar{R}(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}})\right| &= \left|\mathbb{E}_{\bar{\boldsymbol{z}}\sim\bar{p}(\boldsymbol{x},\bar{y})}\left[\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}})\right] - \mathbb{E}_{\bar{\boldsymbol{z}}\sim\bar{p}(\boldsymbol{x},\bar{y})}\left[\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}}, \bar{\boldsymbol{z}})\right]\right| \\
&= \left|\mathbb{E}_{\bar{\boldsymbol{z}}\sim\bar{p}(\boldsymbol{x},\bar{y})}\left[\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}}) - \bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}}, \bar{\boldsymbol{z}})\right]\right| \leq \xi.
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
\left|\bar{R}_n(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}}) - \bar{R}_n(\boldsymbol{f}_{\bar{D}})\right| &= \left|\frac{1}{n}\left\{\sum_{j=1,j\neq i}^{n}\left|\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}}_j) - \bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}}, \bar{\boldsymbol{z}}_j)\right| + \left|\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}}_i) - \bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}}, \bar{\boldsymbol{z}}_i')\right|\right\}\right| \\
&\leq \frac{\left|\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}}_i) - \bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}}, \bar{\boldsymbol{z}}_i')\right|}{n} + \sum_{j=1,j\neq i}^{n}\frac{\left|\bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}}, \bar{\boldsymbol{z}}_j) - \bar{\mathcal{L}}(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}}, \bar{\boldsymbol{z}}_j)\right|}{n}
\end{aligned}$$

$$\leq \frac{M}{n} + \xi.$$

Based on these two inequalities, we have

$$\left|\Upsilon(\bar{D}) - \Upsilon(\bar{D}^{i,\bar{\boldsymbol{z}}_i'})\right| \leq 2\xi + \frac{M}{n}.$$

By applying McDiarmid's inequality to $\Upsilon(\bar{D})$, for any $\epsilon > 0$, we obtain

$$\begin{aligned} P\left(\bar{R}_n(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}}) - \bar{R}_n(\boldsymbol{f}_{\bar{D}}) \geq \xi + \epsilon\right) &= P\left(\Upsilon(\bar{D}) \geq \xi + \epsilon\right) \\ &\leq P\left(\Upsilon(\bar{D}) \geq \mathbb{E}[\Upsilon(\bar{D})] + \epsilon\right) \\ &\leq \exp\left(\frac{-2n\epsilon^2}{(2n\xi + M)^2}\right), \end{aligned}$$

let $\delta = \exp\left(\frac{-2n\epsilon^2}{(2n\xi+M)^2}\right)$, then $\epsilon = (2n\xi + M)\sqrt{\frac{\ln(1/\delta)}{2n}}$. Finally, for any $\delta > 0$, with the probability at least $1 - \delta$, we have

$$\bar{R}_n(\boldsymbol{f}_{\bar{D}^{i,\bar{\boldsymbol{z}}_i'}}) - \bar{R}_n(\boldsymbol{f}_{\bar{D}}) \leq \xi + (2n\xi + M)\sqrt{\frac{\ln(1/\delta)}{2n}}.$$

□

Theorem 1 shows that the proposed method possesses a generalization bound based on uniform stability with replacing instances, and the convergence rate is $\mathcal{O}(\sqrt{n})$.

APPENDIX B
ADDITIONAL EXPERIMENTS

B.1 Effects of Different Regularization Methods

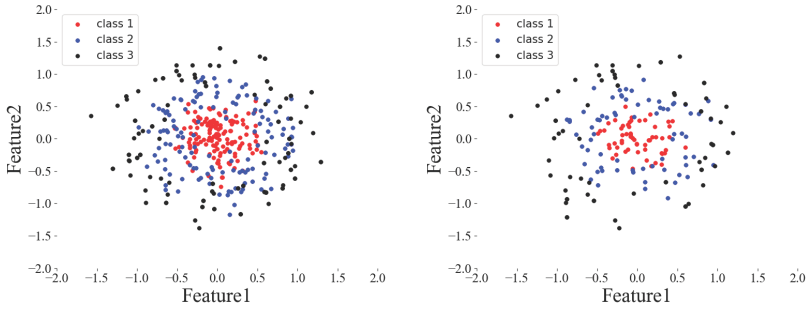
Due to its simplicity and smoothness, MSE is adopted as the regularization method in the consistency term (Eq. (5)) to align model outputs for augmented instances with the extracted label correlations in our framework. To further examine this choice, we conducted additional experiments with two alternatives in Eq. (5): MAE loss and *Kullback–Leibler* (KL) divergence. The results in Table 8 show that "NMCB w/ MAE" and "NMCB w/ KL" generally underperform the MSE-based variant. The weaker performance of MAE may stem from its sparser gradients, which can slow convergence, while KL divergence is sensitive to small probabilities and inherently asymmetric, potentially causing instability. Overall, these findings validate MSE as a balanced and robust regularization method for MLCLL.

TABLE 8: Results (mean±std) with different regularization methods.

Datasets	bookmark	delicious	mediamill	eurlex_dc	eurlex_sm
One Error↓					
NMCB	**.478±.006**	**.426±.012**	**.174±.016**	.457±.013	.415±.020
NMCB w/ MAE	.487±.005	.438±.013	.204±.016	**.450±.014**	.436±.022
NMCB w/ KL	.481±.005	.432±.013	.203±.015	.453±.013	**.412±.019**
Coverage↓					
NMCB	**.218±.005**	**.605±.007**	**.444±.023**	**.153±.005**	**.310±.006**
NMCB w/ MAE	.228±.005	.613±.007	.447±.024	**.153±.004**	.314±.010
NMCB w/ KL	.227±.004	.616±.007	.452±.023	.155±.004	.316±.007
Ranking Loss↓					
NMCB	.194±.003	**.293±.005**	**.160±.011**	**.161±.006**	**.212±.006**
NMCB w/ MAE	.195±.003	.296±.005	.161±.011	.167±.004	.213±.010
NMCB w/ KL	**.192±.002**	.295±.005	.169±.010	.162±.004	.217±.007
Average Precision↑					
NMCB	**.623±.004**	**.586±.004**	**.748±.006**	**.658±.010**	.615±.012
NMCB w/ MAE	.615±.004	.581±.004	.747±.007	.650±.010	**.619±.017**
NMCB w/ KL	.622±.004	.580±.004	.745±.006	.650±.010	.615±.011

B.2 Illustration of the Effect on Toy Dataset

We employ a 2-dimensional dataset consisting of three concentric circles to explore the effect of different methods on noisy labels. Each circle's data points share the same label. We generate a total of 600 instances, which are divided into two parts: training data and test data. Specifically, 400 instances associated with complementary labels are used for training, while 200 instances equipped with relevant labels are used for testing. During data generation, we set scale factors between the inner circle, middle circle, and outer circle to 0.2 and 0.65 respectively. At the same time, we add Gaussian noise with zero mean and standard deviation of 0.2 to instances. Fig. 4 shows the training data and test data. For all methods, we regard a fully connected neural network composed by four layers with the ReLU activation functions as the predictive model. SGD with momentum 0.9 is used for optimization. Weight decay and learning rate are set as 10^{-3} and 10^{-1}, respectively.



(a) Training dataset (b) Test dataset

Fig. 4: The toy dataset we used to explore decision boundaries.