

# Multi-class Label Noise Learning via Loss Decomposition and Centroid Estimation

Yongliang Ding<sup>1,2</sup>, Tao Zhou<sup>1,2\*</sup>, Chuang Zhang<sup>1,2</sup>, Yijing Luo<sup>1,2</sup>, Juan Tang<sup>3\*</sup> and Chen Gong<sup>1,2\*</sup>

<sup>1</sup>PCA Lab, the Key Laboratory of Intelligent Perceptron and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, China

<sup>2</sup>Jiangsu Key Lab of Image and Video Understanding for Social Security

<sup>3</sup>School of Computer Science and Cyber Engineering, Guangzhou University, China  
{ylding; c.zhang; yjluo; chen.gong}@njjust.edu.cn, taozhou.ai@gmail.com, tanjn16@gzhu.edu.cn

## Abstract

In real-world scenarios, many large-scale datasets often contain inaccurate labels, *i.e.*, noisy labels, which may confuse model training and lead to performance degradation. To overcome this issue, Label Noise Learning (LNL) has recently attracted much attention, and various methods have been proposed to design an unbiased risk estimator to the noise-free dataset to combat such label noise. Among them, a trend of works based on Loss Decomposition and Centroid Estimation (LDCE) has shown very promising performance. However, existing LNL methods based on LDCE are only designed for binary classification, and they are not directly extendable to multi-class situations. In this paper, we propose a novel multi-class robust learning method for LDCE, which is termed “MC-LDCE”. Specifically, we decompose the commonly adopted loss (*e.g.*, mean squared loss) function into a label-dependent part and a label-independent part, in which only the former is influenced by label noise. Further, by defining a new form of data centroid, we transform the recovery problem of a label-dependent part to a centroid estimation problem. Finally, by critically examining the mathematical expectation of clean data centroid given the observed noisy set, the centroid can be estimated which helps to build an unbiased risk estimator for multi-class learning. The proposed MC-LDCE method is general and applicable to different types (*i.e.*, linear and nonlinear) of classification models. The experimental results on five public datasets demonstrate the superiority of the proposed MC-LDCE against other representative LNL methods in tackling multi-class label noise problem.

**Keywords:** Multi-class Classification; Label Noise; Loss Decomposition; Centroid Estimation.

## 1 Introduction

With the availability of large quantities of well-annotated datasets, recent learning methods such as deep learning have been proven successful in various practical tasks. However, it is very expensive to collect large-scale well-annotated data in some fields, such as medical image analysis, speech translation, natural language processing, and so on. Currently, several strategies with low cost have been developed to collect labeled data for many tasks, such as automatic web crawlers and crowdsourcing. However, these strategies inevitably introduce many incorrect labels, due to the limitation of technologies and human expertise, leading to dramatic performance degradation of learning models [19, 39]. Therefore, developing effective Label Noise Learning (LNL) algorithms is highly demanded in various real-world applications.

Up to now, different LNL methods have been proposed to deal with the label noise problem [1] over the last few years, and they can be roughly divided into three categories. The first category is sample selection [2, 3, 4] which relies on the memorization effect of neural networks to select probable correctly-labeled examples. To be specific, neural networks tend to overfit the small-loss examples which are considered as clean data in the early learning stage, then gradually overfit the large-loss examples which are likely to be contaminated by label noise. Therefore, many methods based on choosing small-loss examples are proposed to improve LNL performance. The second category is label correction [5, 6, 7] which attempts to identify and correct the potentially incorrect labels through joint optimization of label purification and network weights. The third category is loss correction [8, 9, 10] which modifies the loss functions, which are further minimized, to make them robust to noisy labels.

Among the above three categories of methods, loss cor-

\*Corresponding author.

rection has shown very promising performance due to a solid mathematical foundation, and a popular way to convert the conventional losses to the robust ones is based on loss decomposition and centroid estimation (LDCE), such as Labeled Instance Centroid Smoothing (LICS) [11],  $\mu$ SGD [12], and Centroid Estimation with Guaranteed Efficiency (CEGE) [13]. These methods aim to estimate the real data centroid under a clean set using the observed noisy data so that a noise-robust loss function can be obtained. However, the above works can only deal with binary classification tasks and can hardly be applied to multi-class cases. The reasons are two-fold. First, they only focus on decomposing the binary classification loss such as hinge loss and perceptron loss. As a result, it is difficult to apply these models to multi-class classification tasks. Second, they need to use the facts that  $y^2 = 1$  for the label  $y = +1$  or  $-1$  and the positive label  $+1$  and negative label  $-1$  only differ in the sign. Unfortunately, these facts do not hold under multi-class cases anymore. In this condition, developing LNL methods based on LDCE that are suitable for multi-class classification is highly demanded.

To this end, in this paper we propose a new **Multi-Class LNL** method via loss decomposition and **Centroid Estimation** (termed **MC-LDCE**) to deal with LNL problems. Specifically, we propose to decompose the multi-class classification loss (e.g., mean squared loss) into a label-independent part and a label-dependent part, so that the multi-class label noise only affects the second part. Then by defining a new form of data centroid, we observe that the label-dependent part is strongly related to such centroid which critically governs the model robustness. Further, by investigating the mathematical expectation of centroid under the noisy dataset as well as introducing an elementary row transformation matrix, such data centroid can be estimated which leads to an unbiased risk estimator to the noise-free case for multi-class learning. Besides, our MC-LDCE is quite general and independent of the classification models, which does not need auxiliary clean data unlike some existing methods such as [14] and [15]. The experimental results on typical benchmarks and real-world noisy datasets show that MC-LDCE outperforms the representative LNL methods under different types of multi-class label noise.

## 2 Related Work

LNL is an important branch of weakly-supervised learning [20, 30, 31, 40] which has attracted intensive research over the past decades. We review three major types of existing LNL methods, including sample selection, label correction, and loss correction.

**Sample selection.** The methods of this type try to select the correctly-labeled examples according to different criteria. For example, Jiang *et al.* [16] proposed MentorNet to teach another student network to select the examples with

probably correct labels during training. However, such a selection method cannot overcome the inferiority of accumulated error caused by sample-selection bias. To overcome such drawbacks, Han *et al.* [4] proposed Co-teaching to train two networks simultaneously and update itself with the data selected by its peer network. As for Co-teaching+ [17], it improves Co-teaching by only selecting the small-loss instances with different predictions from two networks. To further explore the information inherited by data, Wei *et al.* [18] proposed to use a joint loss to select small-loss examples, so that more data with the consensus of two networks can be selected.

**Label correction.** Label correction is a quite intuitive solution that identifies the possible incorrectly-labeled data and then corrects their labels for reliable training [5]. However, such clean data identification and correction can be imprecise. Therefore, Samel *et al.* [21] presented a new active deep denoising approach that first builds a deep neural network noise model and then applies an active learning algorithm to identify the optimal denoising function. Besides, Sheng *et al.* [22] effectively correct the labels when the data are collected from the crowdsourcing platform. The works of [6, 23] proposed a self-training scheme that alternatively conducts label correction and trains a deep neural network on noisily-labeled data. However, these methods use original noisy labels for learning properly, so the performance is still likely to be degraded by the noisy data. Therefore, [7] takes a similar way to [6], but it completely replaces all labels with pseudo-labels for model training. Similarly, Song *et al.* [24] proposed a method of which the key idea is to selectively refurbish and exploit unclean samples that can be corrected with high precision, which results in an augmentation of examples available for training.

**Loss correction.** This type of LNL methods aim to correct the traditional loss functions to make them robust to label noise. Natarajan *et al.* [8] provided the general form of a noise corrected loss  $\hat{l}$ , making the minimization of  $\hat{l}$  is equivalent to the minimization of the noise-free loss  $l$  on clean data given the noise rate. To a further step, if the noise rate is not available, Ghosh *et al.* [9] provided a condition to be satisfied when  $l$  is robust to label noise. Van *et al.* [10] provided several examples of such robust non-convex losses, while Masnadi *et al.* [25] showed that the linear or unhinged loss itself is noise-corrected loss. Besides, Jindal *et al.* [26] augmented the standard network by adding a softmax layer for estimating the confusion matrix. Jin *et al.* [27] proposed a maximum likelihood estimation based objective function for robust classification.

## 3 Preliminaries

In this section, we first introduce some notations which will be used in this paper. In traditional supervised learning, we define the  $\mathcal{X} \in \mathbb{R}^d$  and  $\mathcal{Y} = \{0, 1\}^c$  as the input

feature space and the output label space, where  $d$  denotes the dimension of features and  $c$  denotes the number of classes. For a matrix  $\mathbf{M}$ , we define  $\mathbf{M}^\dagger$  as its pseudo-inverse matrix and  $\mathbf{M}^\top$  as its transpose. The superscript “ $\sim$ ” stands for the variable being noisy or estimated and the “ $\hat{\cdot}$ ” indicates that the variable is an empirical quantity.

Let  $\mathcal{D}$  be the underlying noise-free joint distribution of a pair of random variables  $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ . The clean training set  $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$  of  $(X, Y)$  containing  $n$  i.i.d. data points can be drawn identically and independently from  $\mathcal{D}$ , where  $\{\mathbf{x}_i\}_{i=1}^n$  are the feature representations of  $n$  examples and  $\{\mathbf{y}_i\}_{i=1}^n$  are one-hot label vectors which are correct. However, for the supervised learning with noisy labels, we only have access to the noisy distribution  $\tilde{\mathcal{D}}$  of random variables  $(X, \tilde{Y}) \in \mathcal{X} \times \mathcal{Y}$ , and the noisy training set  $\tilde{S} = \{(\mathbf{x}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^n$  containing  $n$  i.i.d. data points which is drawn from a noisy distribution  $\tilde{\mathcal{D}}$ . In our learning problem under noisy labels, the target is to find a suitable multi-class decision function  $h \in \mathcal{H} : \mathcal{X} \rightarrow \mathcal{Y}$  with parameter matrix  $\mathbf{W} \in \mathbb{R}^{d \times c}$  training on  $\tilde{S}$ , where  $\mathcal{H}$  represents the hypothesis space, such that  $h$  can precisely predict the label  $\mathbf{y}$  of any test example  $\mathbf{x} \in \mathcal{X}$ .

To further understand the corruption process under label noise, we introduce the noise transition matrix  $\mathbf{T} \in [0, 1]^{c \times c}$ , of which the main purpose is to model the transition from latent clean label  $Y$  to the observed noisy label  $\tilde{Y}$ . In this case, the element  $\mathbf{T}_{ij} = p(\tilde{Y} = \mathbf{e}_j | Y = \mathbf{e}_i)$  stands for the probability of clean label  $\mathbf{e}_i$  wrongly annotated as  $\mathbf{e}_j$  where  $\mathbf{e}_i$  denotes the one-hot vector with the  $i$ -th element being 1 and the others being 0.

## 4 Proposed Method

In this section, we will detail our proposed MC-LDCE algorithm.

### 4.1 Multi-class Loss Decomposition

We define  $l : \mathbb{R} \times \mathcal{Y} \rightarrow \mathbb{R}$  as the loss function that penalizes the difference between the model output  $h(\mathbf{x})$  and the ground-truth label  $\mathbf{y}$  under traditional supervised learning. Thus, the empirical risk on a clean set  $S$  can be formulated as

$$(4.1) \quad \hat{\mathcal{R}}(h, S) = \frac{1}{n} \sum_{i=1}^n \ell(h(\mathbf{x}_i), \mathbf{y}_i),$$

where  $h$  is the shorthand for  $h(\mathbf{x})$  in this paper if no confusion is incurred. Similar to Eq. (4.1), we can define the empirical risk on corrupted data set  $\tilde{S}$  as Eq. (4.2).

$$(4.2) \quad \tilde{\mathcal{R}}(h, \tilde{S}) = \frac{1}{n} \sum_{i=1}^n \ell(h(\mathbf{x}_i), \tilde{\mathbf{y}}_i).$$

Because of the unavailability of the ground-truth  $\{\mathbf{y}_i\}_{i=1}^n$ , the  $\tilde{\mathcal{R}}(h, \tilde{S})$  can be deviated from the real  $\hat{\mathcal{R}}(h, S)$ .

It is expected to find an unbiased estimator  $\tilde{\hat{\mathcal{R}}}(h, \tilde{S})$  of  $\hat{\mathcal{R}}(h, S)$  on the noisy set  $\tilde{S}$ , so that the negative impact caused by the noisy label can be eliminated.

As mentioned earlier, our method is based on LDCE to deal with multi-class LNL, while previous works [12, 13] based on LDCE are designed to solve the binary LNL. By decomposing the mean squared loss and expressing the decision function as  $h(\mathbf{x}; \mathbf{W}) = \langle \mathbf{W}, \mathbf{x} \rangle$ . Eq. (4.1) can be reformulated as:

$$(4.3) \quad \begin{aligned} \hat{\mathcal{R}}(h, S) &= \frac{1}{n} \sum_{i=1}^n \|\mathbf{y}_i - \mathbf{W}^\top \mathbf{x}_i\|_2^2 \\ &= \frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i^\top \mathbf{y}_i - 2\mathbf{y}_i^\top \mathbf{W}^\top \mathbf{x}_i + \mathbf{x}_i^\top \mathbf{W} \mathbf{W}^\top \mathbf{x}_i) \\ &= \frac{1}{n} \sum_{i=1}^n (\mathbf{y}_i^\top \mathbf{y}_i + \mathbf{x}_i^\top \mathbf{W} \mathbf{W}^\top \mathbf{x}_i) - \frac{2}{n} \sum_{i=1}^n \mathbf{y}_i^\top \mathbf{W}^\top \mathbf{x}_i. \end{aligned}$$

It should be noted that when the label vector  $\mathbf{y}_i$  follows the form of one-hot encoding,  $\mathbf{y}_i^\top \mathbf{y}_i = 1$  always holds. Besides, from the knowledge of linear algebra, the following equation holds, which is

$$(4.4) \quad \mathbf{y}_i^\top \mathbf{W}^\top \mathbf{x}_i = \text{trace}(\mathbf{y}_i^\top \mathbf{W}^\top \mathbf{x}_i) = \text{trace}(\mathbf{W}^\top \mathbf{x}_i \mathbf{y}_i^\top).$$

Therefore, according to Eq. (4.4), Eq. (4.3) can be derived as

$$(4.5) \quad \begin{aligned} \hat{\mathcal{R}}(h, S) &= \frac{1}{n} \sum_{i=1}^n \left(1 + \mathbf{x}_i^\top \mathbf{W} \mathbf{W}^\top \mathbf{x}_i\right) - \frac{2}{n} \sum_{i=1}^n \text{trace}(\mathbf{W}^\top \mathbf{x}_i \mathbf{y}_i^\top) \\ &= \left(1 + \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{W} \mathbf{W}^\top \mathbf{x}_i\right) - \frac{2}{n} \text{trace}(\mathbf{W}^\top \sum_{i=1}^n \mathbf{x}_i \mathbf{y}_i^\top) \\ &= \left(1 + \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{W} \mathbf{W}^\top \mathbf{x}_i\right) - 2\text{trace}(\mathbf{W}^\top \hat{\mu}(S)), \end{aligned}$$

where the empirical centroid  $\hat{\mu}(S)$  of the clean data set  $S$  is defined as

$$(4.6) \quad \hat{\mu}(S) = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{y}_i^\top.$$

Note that the dataset centroid  $\hat{\mu}(S)$  defined here is different from of that defined in previous works [12, 13] which target binary classification, and this is essential for our method to handling multi-class classification. Corresponding

to Eq (4.6), the mathematical expectation of the centroid on the entire distribution  $\mathcal{D}$  is defined as

$$(4.7) \quad \mu(\mathcal{D}) = \mathbb{E}_{(X,Y) \sim \mathcal{D}}[XY^\top],$$

where  $\mathbb{E}[\cdot]$  calculates the mathematical expectation. From Eq. (4.5), we see that only the second term is related to the label value  $\mathbf{y}_i$ . Therefore, if we want to find an unbiased  $\tilde{\mathcal{R}}(h, \tilde{S})$  to  $\hat{\mathcal{R}}(h, S)$  to deal with noisy labels, we need to accurately estimate the dataset centroid  $\hat{\mu}(S)$  based on  $\tilde{S}$ , and the obtained estimator for  $\hat{\mu}(S)$  is dubbed  $\tilde{\mu}(S)$ . Finally, the proposed MC-LDCE model can be built by combining the empirical risk Eq. (4.5) with some techniques for preventing overfittings, such as the linear model with  $\ell_2$  regularizer, and the neural network with a dropout operation.

From Eq. (4.5), we see the key to recover the empirical risk on clean set is to accurately estimate the centroid (*i.e.*, Eq. (4.6)), and this will be detailed in the following section.

## 4.2 Centroid Estimation

We estimate  $\hat{\mu}(S)$  of the clean set  $S$  via the centroid  $\hat{\mu}(\tilde{S})$  of the noisy set  $\tilde{S}$ . To this end, we investigate the mathematical expectation of the centroid on the noisy set, which can be formulated by

$$(4.8) \quad \mathbb{E}_{\tilde{Y}}[X\tilde{Y}^\top|(X, Y)] = \sum_{i=1}^c \pi_i \mathbb{E}_{\tilde{Y}}[X\tilde{Y}^\top|(X, Y = \mathbf{e}_i)],$$

where  $\mathbf{e}_i$  represents a one-hot vector of which only the  $i$ -th element is 1, so  $Y = \mathbf{e}_i$  denotes that the example belongs to the  $i$ -th class. Besides,  $\pi_i = P(Y = \mathbf{e}_i)$  stands for the prior probability of the  $i$ -th class. To compute the value of Eq. (4.8), we need the following definition:

**DEFINITION 4.1.** Suppose there are two one-hot label vectors, where  $\mathbf{y}_i$  has the  $i$ -th element being 1, and  $\mathbf{y}_j$  has the  $j$ -th element being 1 where  $i \neq j$ . Then the two vectors can be converted using an *imputation matrix*  $\mathbf{K}_{i \rightarrow j}$ , which is

$$(4.9) \quad \mathbf{y}_j = \mathbf{K}_{i \rightarrow j} \mathbf{y}_i,$$

where  $\mathbf{K}_{i \rightarrow j}$  can be obtained by swapping the  $i$ -th row and the  $j$ -th row of an identity matrix  $\mathbf{I}$ .

Therefore, for the  $j$ -th class, we introduce the *imputation matrix*  $\mathbf{K}_{i \rightarrow j}$  defined in Definition 4.1, and then have

$$(4.10) \quad \mathbb{E}_{\tilde{Y}}[X\tilde{Y}^\top|(X, Y = \mathbf{e}_i)] = \sum_{j=1}^c \mathbf{T}_{ij} XY^\top \mathbf{K}_{i \rightarrow j}^\top,$$

where  $\mathbf{T}$  is the noise transition matrix defined in Section 3. The estimation for this matrix can be completed by employing some off-the-shelf tools such as the method in [28, 29].

---

## Algorithm 1 The overall algorithm of MC-LDCE.

---

- 1: **Input:** Noisy training dataset  $\tilde{S} = \{(\mathbf{x}_i, \tilde{\mathbf{y}}_i)\}_{i=1}^n$ ;
  - 2: Estimate the transition matrix  $\mathbf{T}$  via VolMinNet [32];
  - 3: Compute all class priors  $\pi_1, \dots, \pi_c$  via Eq. (4.14);
  - 4: Compute  $\mathbf{M}$  via Eq. (4.11);
  - 5: Compute the estimated centroid of  $S$  via Eq. (4.12);
  - 6: Compute the unbiased risk estimator  $\tilde{\mathcal{R}}(h, \tilde{S})$  via Eq. (4.13);
  - 7: Use any off-the-shelf solver to optimize the model (*e.g.*, linear model or CNN) by employing the  $\tilde{\mathcal{R}}(h, \tilde{S})$  as the loss function;
  - 8: **Output:** Optimal parameters  $\mathbf{W}$ .
- 

In this work, we use the state-of-art method VolMinNet [32] to estimate  $\mathbf{T}$ .

Based on Eq. (4.10), Eq. (4.8) can be further derived as

$$(4.11) \quad \begin{aligned} \mathbb{E}_{\tilde{Y}}[X\tilde{Y}^\top|(X, Y)] &= \sum_{i=1}^c \pi_i \sum_{j=1}^c \mathbf{T}_{ij} XY^\top \mathbf{K}_{i \rightarrow j}^\top \\ &= XY^\top \underbrace{\left[ \sum_{i=1}^c \pi_i \sum_{j=1}^c \mathbf{T}_{ij} \mathbf{K}_{i \rightarrow j}^\top \right]}_{\mathbf{M}}. \end{aligned}$$

Here we denote  $\mathbf{M} = \sum_{i=1}^c \pi_i \sum_{j=1}^c \mathbf{T}_{ij} \mathbf{K}_{i \rightarrow j}^\top$ . Thus, the unbiased estimator  $\tilde{\mu}(S)$  can be formulated as

$$(4.12) \quad \tilde{\mu}(S) = \hat{\mu}(\tilde{S}) \mathbf{M}^\dagger,$$

where the  $\mathbf{M}^\dagger$  stands for the pseudo inverse matrix of  $\mathbf{M}$ .

Finally, the unbiased risk estimator  $\tilde{\mathcal{R}}(h, \tilde{S})$  to  $\hat{\mathcal{R}}(h, S)$  under noisy set  $\tilde{S}$  can be obtained by substituting Eq. (4.12) to Eq. (4.5), which can be represented as

$$(4.13) \quad \tilde{\mathcal{R}}(h, \tilde{S}) = 1 + \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{W} \mathbf{W}^\top \mathbf{x}_i - 2 \text{trace}(\mathbf{W} \hat{\mu}(\tilde{S}) \mathbf{M}^\dagger).$$

## 4.3 Class Prior Estimation

Note that in Eq. (4.11), it needs to obtain the class prior  $\pi_1, \dots, \pi_c$ , thus we present how to estimate them based on the noise transition matrix  $\mathbf{T}$  in this subsection, and we will describe the corruption process from clean labels to noisy labels. The element  $\mathbf{T}_{ij} = P(\tilde{Y} = \mathbf{e}_j | Y = \mathbf{e}_i)$  in the matrix represents the label flip rate from the  $i$ -th class to the  $j$ -th class as defined before. Obviously,  $\sum_{j=1}^c \mathbf{T}_{ij} = 1$ . The class priors are defined as  $\pi_1 = P(Y = \mathbf{e}_1), \pi_2 = P(Y = \mathbf{e}_2), \dots, \pi_c = P(Y = \mathbf{e}_c)$  which can be easily obtained by

Table 1: The characteristic of *CIFAR-10*, *MNIST*, *FASHION-MNIST*, *SVHN* and *Animal-10N*.

| Dataset              | # train | # test | size    |
|----------------------|---------|--------|---------|
| <i>MNIST</i>         | 60,000  | 10,000 | 28×28   |
| <i>FASHION-MNIST</i> | 60,000  | 10,000 | 28×28   |
| <i>CIFAR-10</i>      | 50,000  | 10,000 | 32×32×3 |
| <i>SVHN</i>          | 73,257  | 26,032 | 32×32×3 |
| <i>Animal-10N</i>    | 50,000  | 5,000  | 64×64×3 |

solving the following equation

$$(4.14) \quad \begin{cases} P(\tilde{Y} = \mathbf{e}_1) = \mathbf{T}_{11}\pi_1 + \mathbf{T}_{21}\pi_2 + \cdots + \mathbf{T}_{c1}\pi_c \\ \vdots \\ P(\tilde{Y} = \mathbf{e}_i) = \mathbf{T}_{1i}\pi_1 + \mathbf{T}_{2i}\pi_2 + \cdots + \mathbf{T}_{ci}\pi_c \\ \vdots \\ P(\tilde{Y} = \mathbf{e}_c) = \mathbf{T}_{1c}\pi_1 + \mathbf{T}_{2c}\pi_2 + \cdots + \mathbf{T}_{cc}\pi_c \end{cases},$$

where the left-hand side of every equation can be directly estimated based on the noisy  $\tilde{S}$ .

#### 4.4 Summary of the Proposed Method

From Subsection 4.1 to Subsection 4.3, we see that the proposed MC-LDCE approach decomposes the multi-class classification loss (e.g., mean squared loss) into a label-independent part and a label-dependent part, and then directly estimates the label-dependent part via centroid estimation, which makes it can solve the multi-class LNL problem. It is worth noting that MC-LDCE is a simple yet effective LNL algorithm, which is flexible and compatible with different types of classification models  $h(\mathbf{x})$  (e.g., deep nonlinear models and linear models). The detailed steps of our method are summarized in **Algorithm 1**.

## 5 Experiments

In this section, we first provide the experimental settings, including the characteristics of datasets, selected backbone, and some implementation details. Then, we present the experimental results on both simulated and real-world noisy datasets with deep classification models. Further, we validate the robustness of our MC-LDCE when a linear classification model is deployed.

### 5.1 Experiments with Deep Classification Models

In this part, we equip our MC-LDCE with deep classification models and compare it with several representative deep robust methods.

#### 5.1.1 Basic Setup

**Simulated Noisy Datasets.** We verify the effectiveness of our approach on four manually corrupted datasets (i.e., *CIFAR-10*, *MNIST*, *FASHION-MNIST*, and *SVHN*). The statistics of the used datasets are summarized in Ta-

Table 2: Network architectures of the adopted 6-layer CNN and 2-layer MLP.

| 6-layers CNN             | 2-layers MLP         |
|--------------------------|----------------------|
| 3×3 conv, 128 LReLU      |                      |
| 3×3 conv, 128 LReLU      |                      |
| 3×3 conv, 128 LReLU      |                      |
| 2×2 max-pool, stride 2   | dense 784 → 256      |
| dropout, p = 0.25        |                      |
| 3×3 conv, 512 LReLU      | 256 LReLU            |
| 3×3 conv, 256 LReLU      |                      |
| 3×3 conv, 128 LReLU      |                      |
| avg-pool                 |                      |
| 1-1 dense 128 → #classes | dense 256 → #classes |

ble 1. Specifically, *FASHION-MNIST* and *MNIST* consist of 60,000 images for training and 10,000 images for testing, with the number of classes and the scale of each image being 10 and 28×28, respectively. While *CIFAR-10* and *SVHN* are also considered 10-class datasets, with the the scale of each image being 32×32×3. *CIFAR-10* contains 50,000 training images and 10,000 test images. *SVHN* contains 73,257 training images and 26,032 test images. Note that all the original datasets are clean. Following the common setting in [4, 17, 18], we corrupted the training sets manually by using Sym-flipping and Pair-flipping noise transition matrices [1], with the noise rate being {20%, 60%} and {20%, 40%}, respectively. To be specific, the Sym-flipping structure models the scenario where the class of clean label can uniformly flip into other classes, and the Pair-flipping structure models the situation where the class of a clean label can flip into its adjunct class instead of a far-away class.

**Real-world Noisy Dataset.** *Animal-10N* is introduced by [24] recently, which contains five pairs of confusing animals. The images are crawled from several online search engines including *Bing* and *Google* using the predefined labels as the search keyword. All label noise on *Animal-10N* is introduced by human mistakes, and the overall noise rate on the training dataset is around 8% while the test dataset is clean. This dataset contains 50,000 RGB images used for training and 5,000 RGB images for testing, and the resolution of each image is 64×64.

**Compared Baselines.** We compare our MC-LDCE with several popular robust learning methods, including:

- Co-teaching+ [17] trains two deep neural networks simultaneously and lets them teach each other given every mini-batch.
- JoCoR [18] trains two networks and utilizes co-regularization to reduce the diversity of the two networks and combat the noisy labels.
- SIGUA [33] adopts gradient descent on “good” data while using a learning-rate-reduced gradient ascent on “bad” data.
- Generalized Cross-Entropy (GCE) [34] changes the loss function to make the trained neural network more

Table 3: Average test accuracy and the corresponding standard deviation on *CIFAR-10*, *MNIST*, *FASHION-MNIST*, and *SVHN* over the last ten epochs. The best results are marked in **bold**.

| Dataset              | Noise Type and Rate | GCE [34]   | Co-teaching+[17] | JoCoR [18]        | SIGUA [33]        | MC-LDCE           |
|----------------------|---------------------|------------|------------------|-------------------|-------------------|-------------------|
| <i>CIFAR-10</i>      | Symmetry-20%        | 80.73±0.04 | 78.75±0.04       | 84.85±0.03        | 81.49±0.01        | <b>85.18±0.18</b> |
|                      | Symmetry-60%        | 57.40±0.08 | 48.78±0.37       | 69.07±0.07        | 67.38±0.02        | <b>70.34±0.08</b> |
|                      | Pairflip-20%        | 79.03±0.03 | 74.99±0.08       | 83.63±0.06        | 80.59±0.01        | <b>85.46±0.18</b> |
|                      | Pairflip-40%        | 60.01±0.05 | 51.73±0.07       | 62.95±0.09        | 71.37±0.04        | <b>78.22±0.28</b> |
| <i>MNIST</i>         | Symmetry-20%        | 95.88±0.01 | 96.79±0.03       | 96.32±0.03        | 93.64±0.01        | <b>97.40±0.05</b> |
|                      | Symmetry-60%        | 93.95±0.01 | 93.83±0.07       | 94.10±0.06        | 84.39±0.05        | <b>94.84±0.02</b> |
|                      | Pairflip-20%        | 95.93±0.01 | 97.08±0.04       | 95.27±0.02        | 86.87±0.31        | <b>97.28±0.02</b> |
|                      | Pairflip-40%        | 95.16±0.01 | 91.57±0.09       | <b>95.52±0.07</b> | 73.72±0.25        | 93.06±0.19        |
| <i>FASHION-MNIST</i> | Symmetry-20%        | 86.22±0.01 | 87.48±0.05       | 87.42±0.04        | 82.63±0.88        | <b>87.70±0.04</b> |
|                      | Symmetry-60%        | 84.27±0.01 | 76.64±0.03       | 83.92±0.08        | 75.12±0.22        | <b>84.70±0.04</b> |
|                      | Pairflip-20%        | 86.38±0.02 | 86.89±0.09       | 87.62±0.03        | 77.94±0.51        | <b>87.83±0.18</b> |
|                      | Pairflip-40%        | 85.18±0.02 | 69.10±0.06       | 82.90±0.06        | 70.55±6.17        | <b>85.38±0.36</b> |
| <i>SVHN</i>          | Symmetry-20%        | 81.29±0.01 | 93.02±0.05       | 78.40±0.02        | 92.19±0.01        | <b>94.44±0.04</b> |
|                      | Symmetry-60%        | 56.24±0.01 | 72.13±0.27       | 36.49±0.05        | <b>82.85±0.01</b> | 77.76±0.15        |
|                      | Pairflip-20%        | 92.96±0.01 | 92.55±0.03       | 77.15±0.02        | 90.81±0.01        | <b>93.35±0.04</b> |
|                      | Pairflip-40%        | 83.95±0.01 | 72.49±0.09       | 54.96±0.03        | <b>88.97±0.01</b> | 77.74±0.18        |

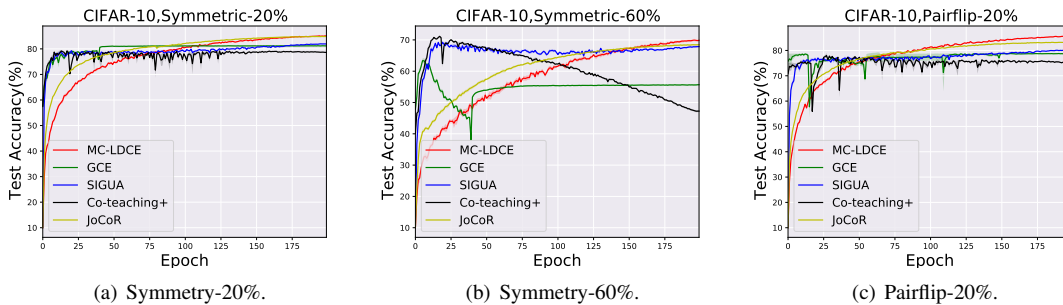


Figure 1: Test accuracy curves on *CIFAR-10* with different noise rates for all the compared methods. Colored curves show the mean accuracy of five trials, and shaded bars denote the standard deviations of the accuracies over five trials.

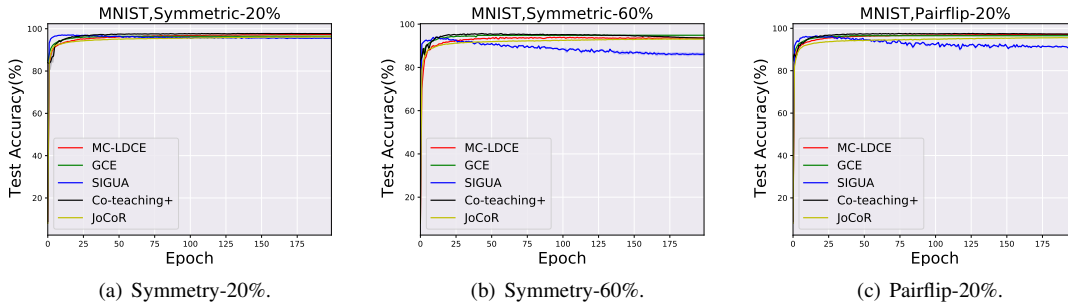


Figure 2: Test accuracy curves on *MNIST* with different noise rates for all the compared methods. Colored curves show the mean accuracy of five trials, and shaded bars denote the standard deviations of the accuracies over five trials.

robust in noisy label situations.

It is worth noting that we do not compare with other unbiased loss correction methods, such as [8, 11, 12], since they are not applicable to multi-class cases.

**Network Architectures.** We adopt a 6-layer Convolutional Neural Network (CNN) as the backbone for *CIFAR-10*, *SVHN* and *Animal-10N*, and 2-layer Fully Connected Neural Network (MLP) for *FASHION-MNIST* and *MNIST*, which are both widely used in the related literature [4, 18,

33, 35]. The detailed network architectures are summarized in Table 2.

**Implementation Details.** For a fair comparison, all experiments are conducted on an RTX2080-Ti GPU. The backbone network architectures are the same for all the methods, and we implement the compared baselines using their default parameters suggested by the original papers. For our MC-LDCE, the model is trained over 200 epochs, and we adopt the Adam algorithm to optimize our model with a

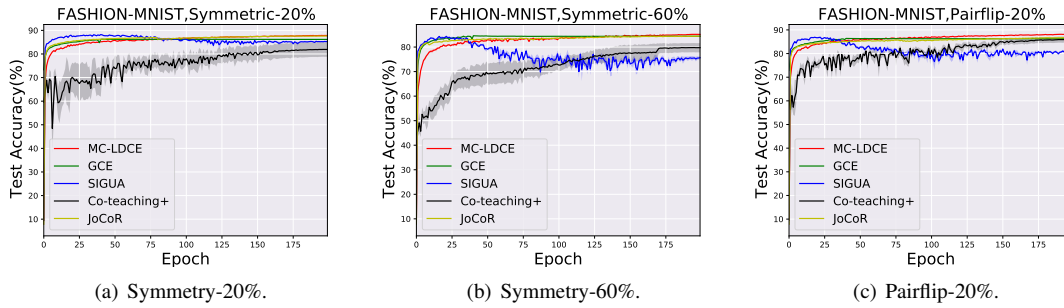


Figure 3: Test accuracy curves on *FASHION-MNIST* with different noise rates for all the compared methods. Colored curves show the mean accuracy of five trials, and shaded bars denote the standard deviations of the accuracies over five trials.

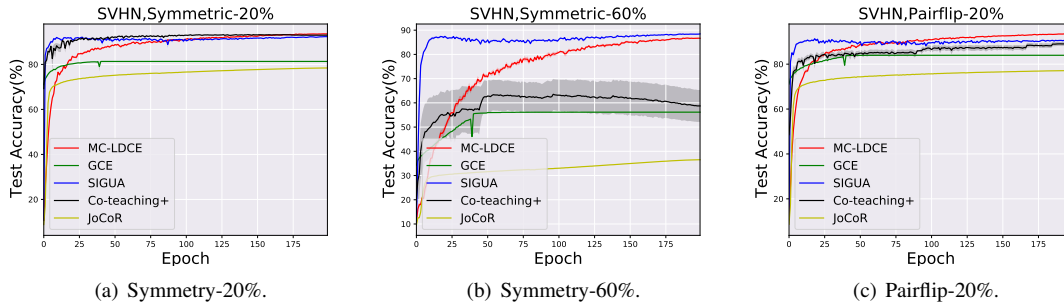


Figure 4: Test accuracy curves on *SVHN* with different noise rates for all the compared methods. Colored curves show the mean accuracy of five trials, and shaded bars denote the standard deviations of the accuracies over five trials.

momentum of 0.9. The initial learning rate is set to 0.001, and will be linearly decreased after 80 epochs. The batchsize is set to 128.

### 5.1.2 Experimental Results

**Results on *CIFAR-10*.** Figure 1 plots the test accuracy vs. number of epochs on *CIFAR-10*. In the easiest Symmetry-20% case, the test accuracy of all compared methods increases steadily over the increase of epochs, which demonstrates their robustness. However, when meeting with a harder case, *i.e.*, 60% symmetric noise, Co-teaching+ and GCE first reach a very high level and then decrease gradually, which is because of the memorization effect of neural networks. To be specific, when the training proceeds, the neural network will tend to overfit the noisy examples which will lead to a decline in test accuracy. While the accuracy of our method increases steadily and finally exceeds all the others, verifying the robustness of our MC-LDCE for extremely corrupted datasets (more than 50% data are corrupted). As for pairflip noise, we can see that our MC-LDCE outperforms the competitors with a large margin. For example, MC-LDCE exceeds the second best method with 1.83% and 6.85% in Pairflip-20% case and Pairflip-40% case, respectively. Thus, the proposed MC-LDCE is effective in dealing with both symmetric and pairflip label noise.

**Results on *MNIST*.** For *MNIST*, we evaluate the proposed method with synthetic label noise, *i.e.*, symmetric label noise with the noise rate in {20%, 60%} and pairflip label noise with the noise rate in {20%, 40%}. We run five individual trials for all the compared methods under each noise level. Figure 2 (a) and (b) plot the test accuracy curves on *MNIST* with 20% and 60% noise rates under symmetric label noise. Figure 2 (c) shows the test accuracy curves with the noise rate of 20% under the pairflip label noise. Table 3 provides us the test accuracies and the corresponding standard deviations of all compared methods. From the results, we can see that the accuracy of our MC-LDCE increases steadily over the increase of epochs, and our method outperforms other compared baselines finally, which indicates the effectiveness of our MC-LDCE in dealing with noisy labels.

**Results on *FASHION-MNIST*.** Figure 1 shows the test accuracy vs. number of epochs on *FASHION-MNIST*. Similarly, the accuracy of our MC-LDCE grows steadily over the increase of epochs and outperforms the other compared baseline gradually. As shown in Table 3, our method on *FASHION-MNIST* consistently outperforms all the compared methods on all label noise cases, which demonstrate the superiority of the proposed method.

**Results on *SVHN*.** The comparison results on *SVHN* with different types of noise and different noise rates are shown in Figure 4 and Table 3. From the comparison results,

Table 4: Average test accuracy on *Animal-10N*. The best results are marked in **bold**.

| Method            | Accuracy (%)       |
|-------------------|--------------------|
| GCE [34]          | 68.7 ± 0.04        |
| Co-teaching+ [17] | 69.7 ± 0.11        |
| JoCoR [18]        | 75.7 ± 0.12        |
| SIGUA [33]        | 74.0 ± 0.21        |
| MC-LDCE           | <b>76.6 ± 0.23</b> |

it can be seen that our MC-LDCE grows stably with the epoch increasing and gradually outperforms the compared methods with a large margin, especially for JoCoR, GCE, and Co-teaching+. In addition, as shown in Table 3, our MC-LDCE can consistently achieve the best or the second best performance among all the compared methods. It is noted that the additional experimental results on four simulated noisy datasets can be found in the **supplementary material**.

**Results on *Animal-10N*.** Similar to the experimental settings on *CIFAR-10*<sup>1</sup>, we run five individual trials for all compared methods on *Animal-10N*. Note that we do not apply any data augmentation or pre-processing procedures. Table 4 shows the average test accuracies and corresponding standard deviations of all compared methods on *Animal-10N*, where we can see that our MC-LDCE achieves the highest classification accuracy among all comparators. Therefore, the proposed MC-LDCE is effective in handling real-world label noise.

## 5.2 Experimental Results with Linear Model

In this part, we equip our MC-LDCE with a linear classification model and compare it with several statistical learning-based robust methods. The compared methods include: 1) Unbiased Logistic Estimator (ULE) [8], 2)  $\mu$  Stochastic Gradient Descent ( $\mu$ SGD) [12], 3) Spectral Cluster Discovery (SCD) [36], 4) Rank Pruning (RP) [37], and 5) Label Noise handling via Side Information (LNSI) [38]. Note that the first two approaches are originally designed for binary classification tasks, so we use the one-vs-all strategy to apply them to multi-class cases. Details of all the compared methods can be found in the **supplementary materials**. In the experiments, we evaluate the proposed method on corrupted *CIFAR-10*. To be specific, we randomly pick up 30,000 images from *CIFAR-10* across different classes and corrupt them with different levels of symmetric noise. The classification accuracies of all the compared approaches under different noise levels are shown in Table 5. It is worth noting that the proposed MC-LDCE consistently outperforms all the competitors under various noise levels, which

<sup>1</sup>The only difference lies in the input dimension of the last fully connected layer of the network architecture, where 1024 is used for *Animal-10N*, while 512 is for *CIFAR-10*, as the sizes of their images are different.

Table 5: Average test accuracy on *CIFAR-10* using a linear classification model. The best results are marked in **bold**.

| Method         | 20%               | 40%               | 60%               |
|----------------|-------------------|-------------------|-------------------|
| ULE [8]        | 74.8±0.050        | 61.7±0.075        | 41.5±0.068        |
| $\mu$ SGD [12] | 74.1±0.009        | 72.6±0.012        | 71.6±0.001        |
| RP [37]        | 77.9±0.015        | 64.6±0.009        | 47.4±0.006        |
| LNSI [38]      | 84.7±0.006        | 83.8±0.006        | 77.4±0.006        |
| SCD [36]       | 86.5±0.007        | 84.5±0.006        | 77.6±0.020        |
| MC-LDCE        | <b>87.1±0.359</b> | <b>85.1±0.441</b> | <b>79.7±0.884</b> |

again demonstrates the superiority of MC-LDCE in dealing with label noise.

## 6 Conclusion

In this paper, we propose a novel multi-class LNL method termed MC-LDCE to deal with the label noise problem. In the proposed MC-LDCE, we decompose the multi-class classification loss (*e.g.*, mean squared loss) into label-independent and label-dependent parts, and directly estimate the label-dependent part via centroid estimation. Our MC-LDCE is the first method based on LDCE to deal with multi-class LNL problems. Furthermore, as our MC-LDCE is independent of the classification model, we conduct intensive experiments by using deep and linear models on both simulated and real-world noisy datasets. Experimental results demonstrate that our MC-LDCE outperforms other representative LNL methods.

**Acknowledgement.** This research is supported by NSF of China (Nos: 61973162, 62172228), NSF of Jiangsu Province (No: BZ2021013), and the Fundamental Research Funds for the Central Universities (Nos: 30920032202, 30921013114), the GuangDong Basic and Applied Basic Research Foundation (No: 2020A1515110554), and the Science and Technology Program of Guangzhou, China (No: 202002030138).

## References

- [1] Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406*, 2020.
- [2] M Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *NeurIPS*, 23:1189–1197, 2010.
- [3] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. Self-paced curriculum learning. In *AAAI*, pages 2694–2700, 2015.
- [4] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pages 8536–8546, 2018.
- [5] Carla E Brodley and Mark A Friedl. Identifying mislabeled training data. In *JAIR*, 11:131–167, 1999.



- [6] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *ICLR Workshop*, 2014.
- [7] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, pages 5552–5560, 2018.
- [8] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *NeurIPS*, 26:1196–1204, 2013.
- [9] Aritra Ghosh, Naresh Manwani, and PS Sastry. Making risk minimization tolerant to label noise. *Neurocomputing*, 160:93–107, 2015.
- [10] Brendan Van Rooyen, Aditya Krishna Menon, and Robert C Williamson. Learning with symmetric label noise: The importance of being unhinged. In *NeurIPS*, pages 10–18, 2015.
- [11] Wei Gao, Lu Wang, Zhi-Hua Zhou, et al. Risk minimization in the presence of label noise. In *AAAI*, volume 30, 2016.
- [12] Giorgio Patrini, Frank Nielsen, Richard Nock, and Marcello Carioni. Loss factorization, weakly supervised learning and label noise robustness. In *ICML*, pages 708–717. PMLR, 2016.
- [13] Chen Gong, Jian Yang, Jane J You, and Masashi Sugiyama. Centroid estimation with guaranteed efficiency: A general framework for weakly supervised learning. In *TPAMI*, 2020.
- [14] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, pages 4334–4343, 2018.
- [15] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, pages 839–847, 2017.
- [16] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pages 2304–2313. PMLR, 2018.
- [17] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *ICML*, pages 7164–7173, 2019.
- [18] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, pages 13726–13735, 2020.
- [19] Chuang Zhang, Qizhou Wang, Tengfei Liu, Xun Lu, Jin Hong, Bo Han and Chen Gong. Fraud Detection under Multi-Sourced Extremely Noisy Annotations. In *CIKM*, pages 2497–2506, 2021.
- [20] Chen Gong, Qizhou Wang, Tongliang Liu, Bo Han, Jane You, Jian Yang and Masashi Sugiyama. Instance-Dependent Positive and Unlabeled Learning with Labeling Bias Estimation. In *TPAMI*, 2021.
- [21] Karan Samel and Xu Miao. Active deep learning to tune down the noise in labels. In *SIGKDD*, pages 685–694, 2018.
- [22] Victor S Sheng, Jing Zhang, Bin Gu, and Xindong Wu. Majority voting and pairing with multiple noisy labeling. In *TKDE*, pages 31(7):1355–1368, 2017.
- [23] Mengying Hu, Hu Han, Shiguang Shan, and Xilin Chen. Weakly supervised image classification through noise regularization. In *CVPR*, pages 11517–11525, 2019.
- [24] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *ICML*, pages 5907–5915, 2019.
- [25] Hamed Masnadi-Shirazi and Nuno Vasconcelos. On the design of loss functions for classification: theory, robustness to outliers, and savageboost. *NeurIPS*, pages 1049–1056, 2008.
- [26] Ishan Jindal, Matthew Nokleby, and Xuewen Chen. Learning deep networks from noisy labels with dropout regularization. In *ICDM*, pages 967–972. IEEE, 2016.
- [27] Junwei Jin, Yanting Li, and CL Philip Chen. Pattern classification with corrupted labeling via robust broad learning system. In *TKDE*, 2021.
- [28] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. In *TPAMI*, pages 38(3):447–461, 2015.
- [29] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? In *NeurIPS*, pages 32:6838–6849, 2019.
- [30] Chuang Zhang, Dexin Ren, Tongliang Liu, Jian Yang, Chen Gong. Positive and Unlabeled Learning with Label Disambiguation. In *IJCAI*, pages 4250–4256, 2019.
- [31] Chen Gong, Hong Shi, Tongliang Liu, Chuang Zhang, Jian Yang, Dacheng Tao. Loss Decomposition and Centroid Estimation for Positive and Unlabeled Learning. In *TPAMI* pages 918–932, 2019.
- [32] Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. Provably end-to-end label-noise learning without anchor points. In *ICML*, pages 6403–6413, 2021.
- [33] 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*, pages 4006–4016, 2020.
- [34] Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, 2018.
- [35] Eran Malach and Shai Shalev-Shwartz. Decoupling “when to update” from “how to update”. In *NeurIPS*, 2017.
- [36] Yijing Luo, Bo Han, and Chen Gong. A bi-level formulation for label noise learning with spectral cluster discovery. In *IJCAI*, pages 2605–2611, 2020.
- [37] Curtis G Northcutt, Tailin Wu, and Isaac L Chuang. Learning with confident examples: Rank pruning for robust classification with noisy labels. *arXiv preprint arXiv:1705.01936*, 2017.
- [38] Yang Wei, Chen Gong, Shuo Chen, Tongliang Liu, Jian Yang, and Dacheng Tao. Harnessing side information for classification under label noise. In *TNNLS*, 31(9):3178–3192, 2019.
- [39] Qizhou Wang, Bo Han, Tongliang Liu, Gang Niu, Jian Yang and Chen Gong. Tackling Instance-Dependent Label Noise via a Universal Probabilistic Model. In *AAAI*, pages 10183–10191, 2021.
- [40] Chuang Zhang, Chen Gong, Tengfei Liu, Xun Lu, Weiqiang Wang and Jian Yang. Online Positive and Unlabeled Learning. In *IJCAI*, pages 1926–1932, 2020.

# Multi-class Label Noise Learning via Loss Decomposition and Centroid Estimation (Supplementary Material)

In this supplementary material, we first provide some details of all the compared statistical learning-based robust methods. Then, we present more experimental results on synthetic noisy datasets.

## 1 Details of Compared Linear Model Baselines

In this section, we provide more details of all the compared statistical learning-based robust methods in the linear model part.

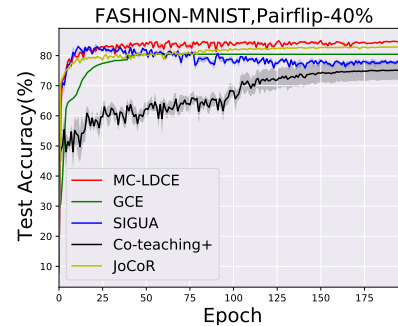
- **Unbiased Logistic Estimator (ULE)** [1]: It provides two approaches to suitably modify any given surrogate loss function when learning with noisy labels.
- **$\mu$  Stochastic Gradient Descent ( $\mu$ SGD)** [2]: It proves that the empirical risk of most well-known loss functions factors into a linear term aggregating all labels with a term that is label-free and can further be expressed by sums of the same loss.
- **Spectral Cluster Discovery (SCD)** [3]: It simultaneously learns a robust classifier (Learning stage) by discovering the low-rank approximation to the ground-truth label matrix and learning an ideal affinity graph (Clustering stage).
- **Rank Pruning (RP)** [4]: It aims at solving the problem of estimating the noise rates, i.e. the fraction of wrong positive and negative labels.
- **Label Noise handling via Side Information (LNSI)** [5]: It treats the example features as side information and formulates the noisy label removal problem as a matrix recovery problem.

For a fair comparison, the prior knowledge such as noise rate is provided for all methods, and LNSI is implemented on a 10-NN graph with Gaussian kernel width  $\sigma_k=0.5$ .

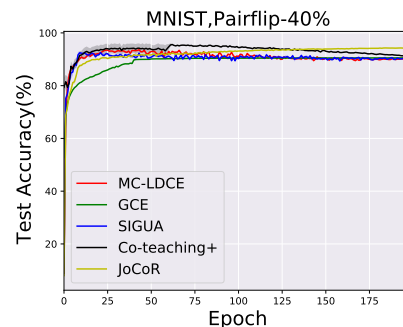
## 2 More Experimental Results

In this section, we provide more experiments on four datasets with label noise rate 40% under pairflip label noise. Figure 1 and Figure 2 plot the test accuracy vs. number of

epochs on *CIFAR-10*, *MNIST*, *FASHION-MNIST*, and *SVHN*, respectively. We can see that our MC-LDCE achieves the best performance on *FASHION-MNIST* and *CIFAR-10*, the second best performance on *SVHN*, and comparable performance on *MNIST*, which clearly demonstrate the effectiveness of our MC-LDCE in dealing with pairflip label noise.



(a) FASHION-MNIST.

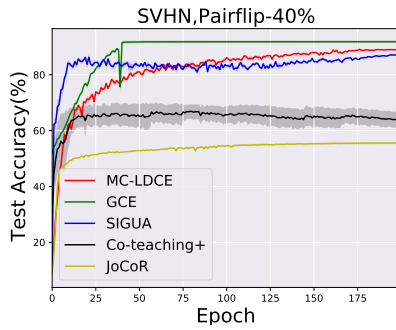


(b) MNIST.

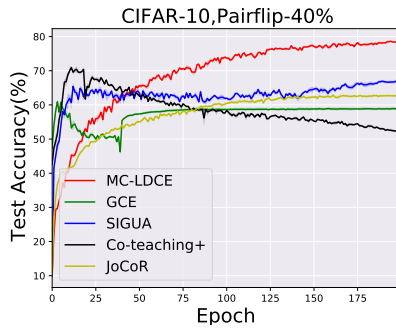
Figure 1: Test accuracy curves on *FASHION-MNIST* and *MNIST* with noise rate 0.4 under pairflip label noise. The colored curves show the mean accuracy of 5 trials, and the shaded bars denote standard deviation.

## References

- [1] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Ad-*



(a) SVHN.



(b) CIFAR-10.

Figure 2: Test accuracy curves on *SVHN* and *CIFAR-10* with noise rate 0.4 under pairflip label noise. The colored curves show the mean accuracy of 5 trials, and the shaded bars denote standard deviation.

*vances in Neural Information Processing Systems*, 26:1196–1204, 2013.

[2] Giorgio Patrini, Frank Nielsen, Richard Nock, and Marcello Carioni. Loss factorization, weakly supervised learning and label noise robustness. In *International Conference on Machine Learning*, pages 708–717. PMLR, 2016.

[3] Yijing Luo, Bo Han, and Chen Gong. A bi-level formulation for label noise learning with spectral cluster discovery. In *International Joint Conference on Artificial Intelligence*, pages 2605–2611, 2020.

[4] Curtis G Northcutt, Tailin Wu, and Isaac L Chuang. Learning with confident examples: Rank pruning for robust classification with noisy labels. *arXiv preprint arXiv:1705.01936*, 2017.

[5] Yang Wei, Chen Gong, Shuo Chen, Tongliang Liu, Jian Yang, and Dacheng Tao. Harnessing side information for classification under label noise. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3178–3192, 2019.