Class-Dependent Label-Noise Learning with Cycle-Consistency Regularization

Anonymous Author(s)
Affiliation
Address
email

Abstract

In label-noise learning, estimating the transition matrix plays an important role in building statistically consistent classifier. Current state-of-the-art consistent estimator for the transition matrix has been developed under the newly proposed *sufficiently scattered* assumption, through incorporating the minimum *volume* constraint of the transition matrix T into label-noise learning. To compute the *volume* of T, it heavily relies on the estimated *noisy class posterior*. However, the estimation error of the *noisy class posterior* could usually be large as deep learning methods tend to easily overfit the noisy labels. Then, directly minimizing the *volume* of such obtained T could lead the transition matrix to be poorly estimated. Therefore, how to reduce the side-effects of the inaccurate *noisy class posterior* has become the bottleneck of such method. In this paper, we creatively propose to estimate the transition matrix under the forward-backward cycle-consistency regularization, of which we have greatly reduced the dependency of estimating the transition matrix T on the *noisy class posterior*. We show that the cycle-consistency regularization helps to minimize the *volume* of the transition matrix T indirectly without exploiting the estimated *noisy class posterior*, which could further encourage the estimated transition matrix T to converge to its optimal solution. Extensive experimental results consistently justify the effectiveness of the proposed method, on reducing the estimation error of the transition matrix and greatly boosting the classification performance.

1 Introduction

Deep learning based algorithms rely heavily on large-scale annotated training data. However, it is often extremely costly and sometimes even infeasible to accurately annotate such massive dataset [13]. An usual way to address this issue is to collect large-scale training data from the crowd-sourcing platform [34] or online query search engine [3], which could inevitably yield noisy data. Training deep model with noisy data can significantly degrade the performance [13, 1, 28]. Therefore, how to mitigate the side-effects of the noisy data automatically becomes very important, and it has drawn increasing attention recently.

Recent studies show that estimating the transition matrix plays an important role in building statistically consistent classifiers for label-noise learning, since the transition matrix can well model the noise generation process. These methods guarantee that classifier learned from the noisy data could approach to the optimal classifier defined on the clean risk asymptotically, as the size of the noisy data increases [13]. The basic principle is that the noisy class-posterior $P(\bar{\mathbf{Y}}|X)$ can be inferred by the clean class-posterior $P(\mathbf{Y}|X = \mathbf{x})$ and the transition matrix $T(\mathbf{x})$, where $T_{ij}(\mathbf{x}) = P(\bar{Y} = j|Y = i, X = \mathbf{x})$, i.e., $P(\bar{\mathbf{Y}}|X = \mathbf{x}) = T(\mathbf{x})P(\mathbf{Y}|X = \mathbf{x})$. This paper focuses

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

on the common class-dependent and instance-independent label noise, i.e., $T(\mathbf{x}) = T$. Usually, the transition matrix T is unidentifiable and hard to learn without additional assumptions [35, 5, 31].

In the literature, many methods try to estimate the transition matrix under the *anchor point* assumption, which assumes that there contain some instances belonging to a specific class almost surely [31, 16, 22]. However, the main challenge is that the *anchor point* assumption can not be always satisfied [24, 37]. Therefore, there gradually appear some methods aiming to develop statistically consistent classifier without anchor points [31, 13]. Among them, state-of-the-art method (VolMinNet [13]) proposed by far the mildest *sufficiently scattered* assumption, where *anchor point* assumption is its special case [13]. They proved that the optimal T is the one with minimum *volume* enclosing the noisy class posterior of training examples. Hence, they proposed to incorporate the minimum *volume* constraint of T into the label-noise learning. To compute the *volume* of T, it heavily relies on the estimated *noisy class posterior*. However, the estimation error of the *noisy class posterior* could usually be large as the deep learning methods tend to easily overfit the noisy labels, especially under the limited training samples [35]. This could lead the transition matrix to be poorly estimated when directly minimizing the *volume* of such obtained T. Therefore, how to reduce the side-effects of the inaccurate *noisy class posterior* has become the bottleneck of such methods.

To address this issue, we propose the forward-backward cycle-consistency regularized algorithm to consistently estimate the transition matrix T, of which we could greatly reduce the dependency of estimating T on the *noisy class posterior*, and further build a statistically consistent classifier. Theoretically, we have proved that minimizing the *volume* of the transition matrix T is equal to maximizing the *volume* of corresponding clean class posterior. More importantly, maximizing the *volume* of the clean class posterior could avoid the side-effects of the estimated inaccurate noisy class-posterior probability on estimating T .

To be specific, based on the the principle that $P(\bar{\mathbf{Y}}|X = \mathbf{x}) = TP(\mathbf{Y}|X = \mathbf{x})$, $P(\bar{\mathbf{Y}}|X = \mathbf{x}) \in \mathbb{R}^{C\times N}$ is the noisy class posterior, $T \in \mathbb{R}^{C\times C}$, and $P(\mathbf{Y}|X = \mathbf{x}) \in \mathbb{R}^{C\times N}$ is the clean class posterior, where C is the number of classes, and N is the number of training examples. Note that, $P(\bar{\mathbf{Y}}|X = \mathbf{x})$ can be estimated by exploiting the noisy data, while T and $P(\mathbf{Y}|X = \mathbf{x})$ are unobserved. For simplicity, we re-write the above equation as $A = TB$ for the following derivation, where A and B represent $P(\bar{\mathbf{Y}}|X = \mathbf{x})$ and $P(\mathbf{Y}|X = \mathbf{x})$, respectively. We adopt the commonly used determinant operation to measure the *volume* of T, i.e., det(T) . As we known, only the square matrix has determinant, thus we compute det($AA^\top$)=det($TBB^\top T$)=det($TT^\top$)det($BB^\top$). As $P(\bar{\mathbf{Y}}|X = \mathbf{x})$, i.e., A, can be estimated by the given noisy label and its determinant det($AA^\top$) is constant, then minimizing det(T) equals to maximizing det($BB^\top$). Therefore, we have successfully transformed the minimum *volume* constraint of T into the maximum *volume* constraint of the clean class posterior $P(\mathbf{Y}|X = \mathbf{x})$.

Obviously, maximizing the *volume* of the clean class posterior $P(\mathbf{Y}|X = \mathbf{x})$ does not rely on the estimated inaccurate noisy class-posterior probability $P(\bar{\mathbf{Y}}|X = \mathbf{x})$. Specifically, we project the given noisy labels, which corresponds to the largest *volume*, onto the clean class posterior space, denoted as backward clean class posterior $\hat{P}(\mathbf{Y}|X = \mathbf{x})$ which intuitively corresponds to large *volume*, through one *diagonally dominant column statistic* matrix T^b. Afterwards, we guide the clean class posterior $P(\mathbf{Y}|X = \mathbf{x})$ learned by the neural network to match to $\hat{P}(\mathbf{Y}|X = \mathbf{x})$. Therefore, the *volume* of $P(\mathbf{Y}|X = \mathbf{x})$ could be maximized to encourage the transition matrix T to converge to its optimal solution. Besides, we further build the cycle-consistency regularization between the forward and backward transition matrices T and T^b, to learn better clean class-posterior probability.

Our main contributions summarize as follows: 1) We are the first to propose to estimate the transition matrix T under the forward-backward cycle-consistency regularization, of which we could greatly reduce the dependency of estimating the transition matrix T on the *noisy class posterior*; 2) We show that such cycle-consistency regularization could help to minimize the *volume* of the transition matrix T without exploiting the estimated noisy class posterior, which further encourages the estimated transition matrix T to converge to the optimal solution; 3) Experimental results on four datasets (two synthetic and two real-world datasets) with different label-noise settings consistently justify the effectiveness of the proposed method, on reducing the estimation error of the transition matrix and greatly improving the classification performance.

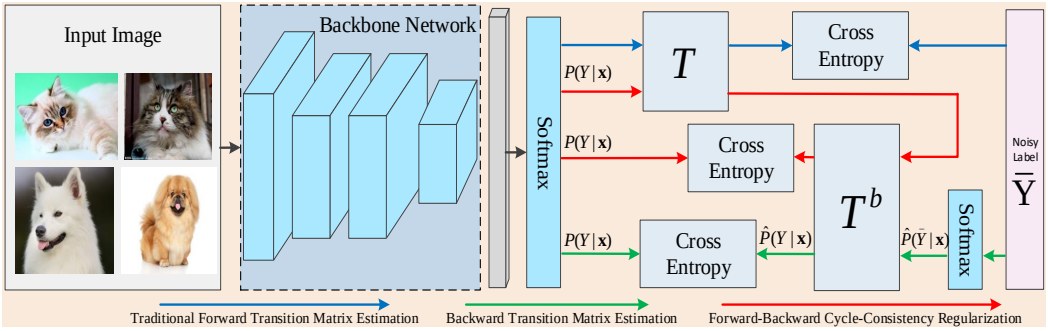

Figure 1: Overview of our method. We optimize the proposed method in an end-to-end manner with three objectives simultaneously. The blue line represents the traditional forward transition matrix T estimation, the green line means the backward transition matrix T^b estimation, and the red line means the cycle-consistency regularization between the forward and backward transition matrices.

2 Estimating Transition Matrix for Label-Noise Learning

Problem Setting. Let D be the distribution of a pair of random variables $(\mathbf{X}, \mathbf{Y}) \in (\mathcal{X} \times \mathcal{Y})$. $\mathbf{X}$ and $\mathbf{Y}$ denote the variable of instances and the corresponding labels, $\mathcal{X} \in \mathbb{R}^{d \times N}$ represents the instance feature space, where d is the feature dimension and N is the total number of instances, $\mathcal{Y} \in \{1, \cdots, C\}$ denotes the label space, and C is the size of the classes. However, in some real-world scenarios, it is often expensive or even infeasible to accurately draw large-scale training data independently from the clean distribution D ideally. This is also the key problem that the label-noise learning methods aim to address.

Let $\bar{D}$ be the distribution of a pair of random variables $(\mathbf{X}, \bar{\mathbf{Y}}) \in (\mathcal{X} \times \mathcal{Y})$, and $\bar{\mathbf{Y}}$ is the corresponding variable of the noisy labels. In label-noise learning, we denote the training dataset as $\bar{\mathcal{D}} = \{(\mathbf{x}_i, \bar{y}_i)\}_{i=1}^{N}$ which is independently drawn from the noisy distribution $\bar{D}$. The goal of the label-noise learning is to learn a robust classifier on the noisy dataset $\bar{\mathcal{D}}$, which aims to approximate to the optimal classifier obtained on the clean dataset.

Transition Matrix plays an important role in building statistically consistent classifier through the "loss correction" strategy. It explicitly models the label-noise generation process from clean data distribution to the noisy data distribution, i.e., $T_{ij}(\mathbf{x}) = P(\bar{Y} = j | Y = i, X = \mathbf{x})$, and $T_{ij}(\mathbf{x}) \in \mathbb{R}^{C \times C}$. It represents the probability that an instance $\mathbf{x}$ having the clean label $Y = i$ is mislabeled as the noisy label $\bar{Y} = j$. Intuitively, the noisy class-posterior probability $P(\bar{\mathbf{Y}}|\mathbf{x})$ can be inferred by the transition matrix $T_{ij}(\mathbf{x})$ and the clean class-posterior probability $P(\mathbf{Y}|\mathbf{x})$, i.e., $P(\bar{\mathbf{Y}}|\mathbf{x}) = T_{ij}(\mathbf{x}) P(\mathbf{Y}|\mathbf{x})$. Note that, in the above equation, only $P(\bar{\mathbf{Y}}|\mathbf{x})$ can be estimated by the noisy data. Therefore, the transition matrix $T(\mathbf{x})$ is generally unidentifiable and very hard to estimate without additional assumptions. In this paper, we focus on the common class-dependent and instance independent label noise, i.e, $T(\mathbf{x}) = T, T \in \mathbb{R}^{C \times C}$.

Current state-of-the-art method (VolMinNet [13]) proposes to estimate the class-dependent transition matrix T under the *sufficiently scattered* assumption, where the traditional *anchor point* assumption is its special case. They proved that the optimal T is the one with minimum *volume* enclosing the noisy class posterior of the training examples to build consistent classifier. However, in this work, we resort to address its equivalence problem, i.e., maximizing the *volume* of corresponding clean class posterior probability $P(\mathbf{Y}|X)$, which could avoid the side-effect of the estimated inaccurate noisy class posterior on estimating T. Therefore, to maximize the *volume* of $P(\mathbf{Y}|X)$, we propose the forward-backward cycle-consistency regularization, and further to consistently estimate T. Specifically, as shown in Fig 1, our proposed method simultaneously optimize three objectives: the cross entropy loss between the given noisy label $\bar{y}$ and the estimated noisy class posterior $P(\bar{\mathbf{Y}}|X) = TP(\mathbf{Y}|X)$; the cross entropy loss between the backward estimated clean class posterior $\hat{P}(\mathbf{Y}|X) = T^b \hat{P}(\bar{\mathbf{Y}}|X)$ and the predicted clean class posterior $P(\mathbf{Y}|X)$ by the neural network; the cycle-consistency regularization to minimize the approximation error between $P(\mathbf{Y}|X)$ and $T^b(TP(\mathbf{Y}|X))$. The backward and cycle-consistency regularization help to minimize the volume of T indirectly without exploiting the noisy class posterior, and further encourage the estimated T to converge to its optimal solution.

2.1 Forward Transition Matrix Estimation

Given the input instance $\mathbf{x}$, the probability of observing a noisy label $\bar{Y}$ can be inferred as,

$$\begin{aligned} P(\bar{Y}=j|\mathbf{x}) &= \sum_{i=1}^{C} P(\bar{Y}=j|Y=i,\mathbf{x})P(Y=i|\mathbf{x}) = \sum_{i=1}^{C} P(\bar{Y}=j|Y=i)P(Y=i|\mathbf{x}) \\ &= \sum_{i=1}^{C} T_{ij}P(Y=i|\mathbf{x}). \end{aligned} \tag{1}$$

Obviously, we can simplify Eq. 1 as $P(\bar{\mathbf{Y}}|\mathbf{x}) = TP(\mathbf{Y}|\mathbf{x})$ under the class dependent and instance independent assumption, where $P(\bar{\mathbf{Y}}|\mathbf{x}) = [P(\bar{Y}=1|\mathbf{x}), ..., P(\bar{Y}=C|\mathbf{x})]^\top$ and $P(\mathbf{Y}|\mathbf{x}) = [P(Y=1|\mathbf{x}), ..., P(Y=C|\mathbf{x})]^\top$ are the noisy class-posterior probability and the clean class-posterior probability, respectively. To learn the clean class-posterior $P(\mathbf{Y}|X)$, we design a neural network $f(\mathbf{x};\mathbf{w})$ parameterized with $\mathbf{w}$, which project the input instance $\mathbf{x}$ onto C classes with probability output. The transition matrix T should be trainable *diagonally dominant column stochastic* matrix, i.e., $T \in [0,1]^{C\times C}, \sum_{i=1}^{C} T_{ij} = 1$ and $T_{ii} > T_{ji}$ for any $i \neq j$. Naturally, we could use $Tf(\mathbf{x};\mathbf{w})$ to learn the noisy class-posterior probability $P(\bar{\mathbf{Y}}|X)$. VolMinNet [13] motivates us that, if $Tf(\mathbf{x};\mathbf{w})$ models $P(\bar{\mathbf{Y}}|X)$ perfectly and the transition matrix T has the minimum volume, T will converge to the optimal transition matrix and the classifier $f(\mathbf{x};\mathbf{w})$ will converge to the optimal $P(\mathbf{Y}|X)$. Here, we first solve the approximation problem between $Tf(\mathbf{x};\mathbf{w})$ and $P(\bar{\mathbf{Y}}|X)$, then we introduce the cycle-consistency regularization to indirectly reduce the volume of T, without exploiting the inaccurate noisy class-posterior probability.

Given the training dataset $\bar{\mathcal{D}} = \{(\mathbf{x}_i, \bar{y}_i)\}_{i=1}^{N}$, we adopt the empirical risk to jointly optimize the transition matrix T and the consistent classifier $f(\mathbf{x};\mathbf{w})$ for label-noise learning. Specifically, we minimize the approximation error between the inferred noisy class-posterior probability $Tf(\mathbf{x};\mathbf{w})$ and the given noisy label $\bar{y}$ as follows,

$$\min_{\mathbf{w},T} \mathcal{L}_1(\mathbf{w},T) = -\frac{1}{N}\sum_{i=1}^{N} \bar{y}_i \log(T \cdot f(\mathbf{x}_i;\mathbf{w})), \tag{2}$$

where N is the number of training examples. Note that, in Eq. 2, both the transition matrix T and the output of classifier $f(\mathbf{x}_i;\mathbf{w})$ are unobservable, which makes the transition matrix T unidentifiable without any assumptions. To optimize Eq. 2, we adopt the same initialization strategy as VolMinNet [13] for constructing T, to make it differentiable. Further, we can optimize the objective in an end-to-end manner.

2.2 Cycle-Consistency Regularization

As shown in Eq. 1, traditional methods always try to estimate the transition matrix T through minimizing the cross-entropy loss between the noisy class-posterior probability $P(\bar{\mathbf{Y}}|X=\mathbf{x})$ and the given noisy label $\hat{y}$, under specific constraints. According to this basic principle, we can also derive that the clean class-posterior probability can be expressed as $P(\mathbf{Y}|X=\mathbf{x}) = T^{-1}P(\bar{\mathbf{Y}}|X=\mathbf{x})$. Therefore, we also propose to estimate the backward transition matrix T^b simultaneously, to act as T^{-1}. Specifically, we denote $\hat{P}(\mathbf{Y}|X=\mathbf{x}) = T^b\hat{P}(\bar{\mathbf{Y}}|X=\mathbf{x})$, where $\hat{P}(\bar{\mathbf{Y}}|X=\mathbf{x})$ is the input noisy class-posterior probability, and $\hat{P}(\mathbf{Y}|X=\mathbf{x})$ is the reversely estimated clean class-posterior probability. Naturally, we could use $\hat{P}(\mathbf{Y}|X=\mathbf{x})$ to learn the clean class posterior $P(\mathbf{Y}|X)$. Note that, since T^b should also be *diagonally dominant column stochastic* matrix, i.e., $T^b \in [0,1]^{C\times C}, \sum_{i=1}^{C} T^b_{ij} = 1$ and $T^b_{ii} > T^b_{ji}$ for any $i \neq j$, the backward transition matrix T^b will be different from T^{-1}.

More specifically, we first transform the one-hot noisy label into the input noisy class-posterior probability by using the Softmax function, denoting it as $\hat{P}(\bar{\mathbf{Y}}|X=\mathbf{x}_i) = SoftMax(\hat{y}_i)$. Then, we use T^b to estimate the backward clean class-posterior probability $T^b\hat{P}(\bar{\mathbf{Y}}|X=\mathbf{x})$. Since the true clean class posterior is unobservable, here we also use the output of the designed neural network

$f(\mathbf{x}_i;\mathbf{w})$ as its approximation. Finally, we minimize the following cross-entropy loss between $\hat{P}(\mathbf{Y}|X=\mathbf{x})$ and $f(\mathbf{x}_i;\mathbf{w})$, to optimize the parameters of $\mathbf{w}$ and T^b simultaneously,

$$\min_{\mathbf{w},T^b} \mathcal{L}_2(\mathbf{w},T^b) = -\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i;\mathbf{w})\log(T^b \cdot SoftMax(\bar{y}_i)). \tag{3}$$

At present, we have obtained the transition matrices from both the forward and backward ways, i.e., T and T^b, respectively. Intuitively, we could build the "indirect" cycle-consistency through minimizing the approximation error between $P(\mathbf{Y}|X=\mathbf{x})$ and $T^b(TP(\mathbf{Y}|X=\mathbf{x}))$, where "indirect" means that we have made full use of the invertible relationship between these two matrices T and T^b indirectly through the original and circularly computed clean class-posterior. Specifically, the cycle-consistency regularization can be expressed as,

$$\min_{\mathbf{w}} \mathcal{L}_3(\mathbf{w};T,T^b) = -\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i;\mathbf{w})\log(T^b(T \cdot f(\mathbf{x}_i;\mathbf{w}))). \tag{4}$$

Note that, in Eq. 4, we just optimize the parameter $\mathbf{w}$ in the network classifier $f(\mathbf{x}_i;\mathbf{w})$, while keep T and T^b be constant. This is because that our ultimate goal is to learn good clean class posterior, if the transition matrices are well learned, the estimated clean class posterior should satisfy this constraint.

Based on the above descriptions, the overall objective function can be expressed as,

$$\mathcal{L} = \mathcal{L}_1(\mathbf{w},T) + \mathcal{L}_2(\mathbf{w},T^b) + \lambda\mathcal{L}_3(\mathbf{w};T,T^b), \tag{5}$$

where λ is the trade-off hyper-parameter to balance the two transition matrix learning objectives with the cycle-consistency regularization term. We optimize all the model parameters simultaneously in an end-to-end manner.

2.3 Theoretical Analysis

In this section, we justify that the proposed method helps to minimize the *volume* of the transition matrix T without relying on the noisy class posterior, which could further encourage the estimated T to converge to its optimal solution.

As theoretically proved in the VolMinNet [13], the optimal transition matrix T is the one with minimum *volume* enclosing the noisy class posterior of the training examples under *sufficiently scattered* assumption. In Section 1, our work theoretically proves that minimizing the *volume* of T is equal to maximize the *volume* of the clean class posterior. Importantly, this transform could reduce the side-effects of the estimated inaccurate noisy class-posterior probability on T estimation. Specifically, our proposed framework as shown in Eq. 5, simultaneously optimize the forward and backward transition matrices T and T^b. Especially during T^b estimation, the input is the given one-hot noisy label which corresponds to the maximum *volume* in practice. When they pass through the diagonally dominant matrix T^b, the estimated clean class-posterior probability $\hat{P}(\mathbf{Y}|X)$ should intuitively has large *volume*, then minimizing the cross entropy between $\hat{P}(\mathbf{Y}|X)$ and $P(\mathbf{Y}|X)$ could help to maximize the *volume* of $P(\mathbf{Y}|X)$. Simultaneously, optimizing Eq. 2 will minimize the *volume* of the transition matrix T. Besides, the cycle-consistency constraint on T and T^b as illustrated in Eq. 4 further helps to learn better clean class posterior.

3 Experiments

In this section, we introduce the experiment setup, including datasets, noise types, and implementation details. We compare our proposed method with the state-of-the-arts on two synthetic and two real noisy datasets, followed by an ablation study to analyze the experimental results and some useful hyper-parameters.

3.1 Experiment Setup

Datasets. Extensive experiments are conducted on two manually corrupted datasets with different noisy types (i.e., CIFAR-10 [9], CIFAR-100 [9]) and two real-world noisy datasets (i.e., Clothing1M [32] and Food-101N [10]), to demonstrate the effectiveness of the proposed method. Both

Table 1: Comparison with state-of-the-art methods on CIFAR-10 and CIFAR-100 datasets. The mean and standard deviation computed over five runs are presented. "Sym-xx%" means the noise rate is xx% and noise type is "Symmetry".

Method	Cifar-10			CIFAR-100		
	Sym-20%	Sym-40%	Sym-60%	Sym-20%	Sym-40%	Sym-60%
CE (baseline)	84.58± 0.18	80.78± 0.38	68.31± 0.33	51.93 ±0.39	40.11± 0.86	25.81± 0.74
GCE [38]	89.31 ±0.07	86.61± 0.23	79.40± 0.41	66.07± 0.24	59.03± 0.21	45.68± 0.39
PeerLoss [17]	88.78 ±0.18	84.87± 0.15	75.28± 0.31	57.34± 0.34	43.39± 0.33	28.66± 0.67
Co-teaching [6]	85.76 ±0.26	83.12± 0.31	70.89± 1.06	56.83± 0.28	43.38± 0.51	28.04± 0.69
Co-teaching++ [36]	86.39±0.33	83.80±0.30	72.51±0.46	57.64±0.34	44.28±0.83	29.60±1.16
T-Revision [31]	88.01±0.16	84.52±0.11	71.53±0.82	62.66±0.53	55.25±0.36	39.94±1.28
VolMinNet [13]	89.69±0.19	85.46±0.19	73.55±0.78	64.70±0.60	56.25±0.45	41.06±0.45
DualT [35]	89.88±0.13	86.23±0.64	72.21±1.67	65.75±0.38	56.80±0.18	42.56±0.55
T-For	89.53±0.11	85.38±0.13	73.01±0.54	64.23±0.64	56.02±0.39	40.89±0.37
$T + T^b$	89.64±0.16	85.47±0.32	73.39±0.40	64.95±0.91	56.36±0.51	41.94±0.43
Ours	**90.44**±0.19	**87.30**±0.25	**81.01**±0.25	**67.74**±0.17	**61.71**±0.20	**49.30**±0.82

CIFAR-10 and CIFAR-100 contain 60K images of size 32×32, of which 50K images constitute the training set and 10K images for testing set, while CIFAR-10 contains 10 classes, and CIFAR-100 contains 100 classes. Clothing1M is a real-world noisy dataset that contains 1M images with about 38.46% noisy labels for training and 10K images with clean labels for testing. Food-101N is also a real-world noisy dataset which contains 310K images with about 19.66% noisy labels for training and 55K images with clean labels for testing.

Noisy type. For CIFAR-10 and CIFAR-100, we manually corrupted the training set according to the ground-truth transition matrices. Specifically, we conducted experiments using three commonly used noisy types: (1) Symmetry flipping [20]; (2) Asymmetry flipping [12]; (3) Pair flipping [8];

3.2 Implementation Details

For fair comparisons, all our experiments are performed on NVIDIA GeForce RTX 3090, and implemented on the same PyTorch platform. For CIFAR-10 and CIFAR-100, the backbone we used is ResNet-34. We train the classification network $f(\mathbf{x}_i; \mathbf{w})$, the transition matrices T and T^b by SGD strategy, with batchsize of 128, momentum 0.9, weight decay 10^{-3}, and learning rate 10^{-2}. For CIFAR-10, the algorithm run 60 epochs and the learning rate is divided by 10 after the 30-th epoch. For CIFAR-100, the algorithm run 80 epochs and the learning rate is divided by 10 after 30-th and 60-th epoch. For Clothing 1M and Food-101N, the backbone we used is ResNet-50 which is pre-trained on ImageNet. We train the classification network $f(\mathbf{x}_i; \mathbf{w})$, the transition matrices T and T^b also with SGD strategy, with batchsize of 32, momentum 0.9, weight decay 10^{-3}, and learning rate 2×10^{-3}. The algorithm run 80 epochs and the learning rate is divided by 10 every 30 epochs. Before training, we warm up on all noisy data with early stopping technique, where we have trained 10, 10, 1 and 1 epochs on the CIFAR-10, CIFAR-100, Clothing 1M and Food 101N datasets, respectively.

3.3 Comparison with state-of-the-art methods

We compare the proposed method with the following representative works: 1) CE, which trains the classification network with the standard cross-entropy loss on the original noisy dataset; 2) GCE [38]; 3) PeerLoss [17]; 4) Co-teaching [6]; 5) Co-teaching++ [36]; 6) T-Revision [31]; 7) Dual-T [35]; VolMinNet [13] which estimates the transition matrix under *sufficiently scattered* assumption.

Results on the synthetic noisy datasets. Tables 1, 2 and 3 report the experiment results under three different synthetic noisy types on cifar10 and cifar100 datasets, respectively. Each table compares the proposed method with 8 representative works on one synthetic noisy type. As illustrated in Eq. 5, our overall objective contains three variants as described in the tables: 1)"T-For" means that we optimize the classifier just use $\mathcal{L}_1(\mathbf{w}, T)$; 2)"$T + T^b$" means that we optimize the classifier use $\mathcal{L}_1(\mathbf{w}, T) + \mathcal{L}_2(\mathbf{w}, T^b)$; 3) "Ours" means that we utilize the overall objective as shown in Eq. 5 to optimize the classifier.
Compared with the representative methods, the proposed method achieves the best performance on

Table 2: Comparison with state-of-the-art methods on CIFAR-10 and CIFAR-100 datasets. The mean and standard deviation computed over five runs are presented. "Asym-xx%" means the noise rate is xx% and noise type is "Asymmetry".

Method	Cifar-10			CIFAR-100		
	Asym-20%	Asym-40%	Asym-60%	Asym-20%	Asym-40%	Asym-60%
CE (baseline)	84.71±0.21	81.26±0.04	68.40±1.16	52.16±0.37	40.21±0.23	26.56±0.64
GCE [38]	89.54±0.21	85.95±0.40	79.55±0.51	65.66±0.73	57.34±0.35	45.46±0.16
PeerLoss [17]	88.98±0.15	85.61±0.59	77.03±0.49	57.51±0.05	43.95±0.35	30.02±0.39
Co-teaching [6]	85.90±0.38	83.09±0.44	71.69±0.50	57.21±0.37	43.76±0.46	30.18±0.71
Co-teaching++ [36]	87.13±0.07	84.86±0.36	73.50±0.47	58.79±0.35	45.26±0.41	32.02±1.22
T-Revision [31]	87.99±0.32	85.17±0.07	72.93±0.27	63.94±0.19	57.19±1.28	42.36±1.09
VolMinNet [13]	89.62±0.15	86.12±0.16	74.80±0.15	65.91±0.25	58.35±0.35	42.16±0.94
DualT [35]	89.36±0.44	86.59±0.30	78.89±0.99	65.76±0.56	56.90±0.39	44.61±1.20
T-For	89.46±0.21	85.74±0.11	74.54±0.12	65.30±0.01	56.31±0.42	42.21±0.58
$T + T^b$	89.62±0.24	86.25±0.03	74.80±0.21	65.52±0.28	57.10±0.20	42.72±0.29
Ours	**90.55**±0.03	**87.29**±0.05	**82.58**±0.24	**68.34**±0.24	**62.64**±0.49	**50.29**±0.24

Table 3: Comparison with state-of-the-art methods on CIFAR-10 and CIFAR-100 datasets. The mean and standard deviation computed over five runs are presented. "Pair-xx%" means the noise rate is xx% and noise type is "Pair".

Method	Cifar-10			CIFAR-100		
	Pair-20%	Pair-40%	Pair-45%	Pair-20%	Pair-40%	Pair-45%
CE (baseline)	86.26±0.41	80.24±0.61	70.73±1.44	52.85±0.35	39.66±0.68	28.06±0.28
GCE [38]	89.77±0.17	84.11±0.15	77.54±0.81	65.37±0.29	50.91±0.64	42.38±0.12
PeerLoss [17]	90.11±0.03	85.52±0.40	76.75±2.98	60.48±0.08	44.55±0.29	39.51±0.19
Co-teaching [6]	86.81±0.31	82.29±0.23	74.34±0.74	59.54±0.14	44.97±0.49	39.59±0.58
Co-teaching++ [36]	87.44±0.33	83.59±0.33	73.96±0.35	63.98±0.55	46.76±0.73	42.60±0.72
T-Revision [31]	90.90±0.11	87.03±0.37	75.86±1.16	66.37±0.35	44.64±0.73	38.84±0.47
VolMinNet [13]	90.99±0.27	88.97±0.22	74.77±2.15	69.70±0.30	44.63±0.64	39.23±0.17
DualT [35]	90.18±0.39	88.81±0.07	77.22±2.08	70.07±0.22	53.15±0.48	42.58±0.54
T-For	90.25±0.40	88.40±0.35	74.08±1.88	69.27±0.14	44.65±0.37	39.10±0.26
$T + T^b$	90.67±0.27	89.35±0.49	78.62±1.40	69.50±0.53	44.79±0.65	39.16±0.58
Ours	**91.67**±0.27	**91.36**±0.13	**91.08**±0.08	**71.63**±0.39	**70.87**±0.14	**69.18**±1.30

both datasets with three synthetic noise types under three different noise ratios. The evaluation results of the three tables can be summarized as follows:

- Compared with representative methods, the proposed method has significant improvements over these methods, which specifically outperforms the baseline method "CE" and the representative work VolMinNet [13] by a large margin of 16.80%and 7.24% on average.
- As the noise rate increases, the superiority of this method gradually emerges. As shown in these three tables: for symmetric noise, the average advantage of our method over other methods is 4.47%, 6.65% and 10.54% at noise rate of 20%, 40% and 60%, respectively; For asymmetric noise, the average advantage of our method is 4.46%, 6.61% and 10.28% at noise rate of 20%, 40% and 60%, respectively; For pair noise, the average advantage of our method is 4.61%, 15.31% and 22.88% at noise rate of 20%, 40% and 45%, respectively. This indicates that our method can handle difficult cases with high noise rates much better.
- By comparing the three variants of our method, it clearly shows that integrating the backward transition matrix and cycle-consistency regularization can greatly help to improve the baseline performances step by step.

Results on the real-world datasets.. Tables 4 and 5 compare the proposed method with representative works on Clothing1M and Food-101N datasets, respectively. We can clearly see that our methods outperforms the baseline method "CE" and VolMinNet [13] by a margin of 1.79% and 0.91% on Clothing1M dataset, and 2.27% and 0.67% on Food-101N dataset. What's more, since the proposed method can be used as a plug-and-play module, we integrate this module into the representative work DivideMix [11], and denoted as "DivideMix+Ours". To further illustrate its effectiveness, we also combine DivideMix with VolMinNet as another baseline, denoted as "DivideMix+VolMinNet". We can clearly see that our method achieves superior performances over all these methods.

Table 4: Classification accuracy (%) on the Clothing1M dataset. (*) indicates that the implementation of the compared method is based on the authors' code.

Methods	CE (Baseline)	GCE [38]	SL [26]	Co-teaching [6]	JointOpt [23]	L_{DMI} [33]
Accuracy	68.94	69.75	71.02	69.21	72.16	72.46
Methods	PTD-R-V [30]	ERL [15]	ForwardT [20]	JoCor [27]	CORES [5]	CAL [39]
Accuracy	71.67	72.87	69.84	70.30	73.24	74.17
Methods	MEIDTM [4]	VolMinNet* [13]	Ours	DivideMix* [11]	DivideMix+VolMinNet	DivideMix+Ours
Accuracy	73.05	69.82	70.73	74.67	74.83	**75.12**

Table 5: Classification accuracy (%) on the Food-101N dataset. (*) indicates that the implementation of the compared method is based on the authors' code.

Methods	CE (Baseline)	CleanNet$_{WHard}$ [10]	CleanNet$_{WSoft}$ [10]	DeepSelf [7]	NoiseResist [14]
Accuracy	81.44	83.47	83.95	85.11	84.70
Methods	VolMinNet* [13]	Ours	DivideMix* [11]	DivideMix+VolMinNet	DivideMix+Ours
Accuracy	83.04	83.71	84.39	85.07	**86.13**

3.4 Ablation Study

As shown in Figure 2 (a) to (l), we also show the estimation error of the transition matrix T of our method during the model training, with different noise types and different noise rates, on the CIFAR-10 and CIFAR-100 datasets, to evaluate the estimated transition matrix T. The estimation error is measured by the l_1 norm between the ground-truth transition matrix and the estimated transition matrix. It shows that the proposed method always outperforms previous algorithms on T estimation. To explore the effect of hyper-parameter λ on model performance, we conducted experiments with different λ under different noise types and noise rates on CIFAR-10 and cifar-100 datasets, as shown in Figure 2 (m) to (p). Each experiment run five times. We summarize Figure 2 (m) to (p) as follows: 1) The smaller the T estimation error is, the higher the test accuracy will be; 2) When $\lambda = 0.3$, the estimation error is the smallest and we get the best performances. Based on this observation, we set λ as 0.3 in all our experiments.

4 Related Work

Based on the statistical consistency of the learned classifier, we roughly divide existing label-noise learning methods into two categories: algorithms with statistically inconsistent classifiers (i.e., heuristic algorithms), and the algorithms with statistically consistent classifiers.

The statistically inconsistent algorithms usually explore some heuristics to reduce the side-effect of the noisy labels. The representative works include: 1) the data cleaning methods which specially design some strategies to select reliable examples [18, 5]; 2) the "label correction" methods which aim to improve the label quality during model training [23, 21]; 3) the semi-supervised learning methods which treat the unreliable examples as unlabeled data and then adopt some self-supervised training technics for robust feature representation learning [11]; 4) some other classic machine learning techniques [19, 20], such as soft label based methods [29], early stop tricks to avoid over-fitting [2]. Although these methods empirically work well without explicitly modeling the label noise distribution, we can not theoretically guarantee the consistency of the learned classifiers [6].

The statistically consistent algorithms aim to learn consistent classifier with the noisy data, which will asymptotically converge to the optimal classifier obtained on its corresponding clean risk[35]. These methods are primarily designed based on the "loss correction" strategy. Among them, the transition matrix $T(\mathbf{x})$ has been used to modify the loss function to build consistent classifiers. Specifically, the anchor point assumption has been widely adopted to estimate the transition matrix [16, 31], and some methods directly assume that the anchor points have already been given. However, this assumption sometimes could be too strong in real application. When no anchor point is given in the dataset, these algorithms can not be guaranteed statistically consistent. Then, to remove this strong dependence on anchor points, some methods propose the slack variable trick [25], which just use the data points with high noisy class-posterior probabilities to estimate the transition matrix.

Afterwards, a series of assumptions [4, 30, 13] have been proposed to efficiently estimate the transition matrix. For example, Dual-T estimator [35] introduces the intermediate class to factorize the original

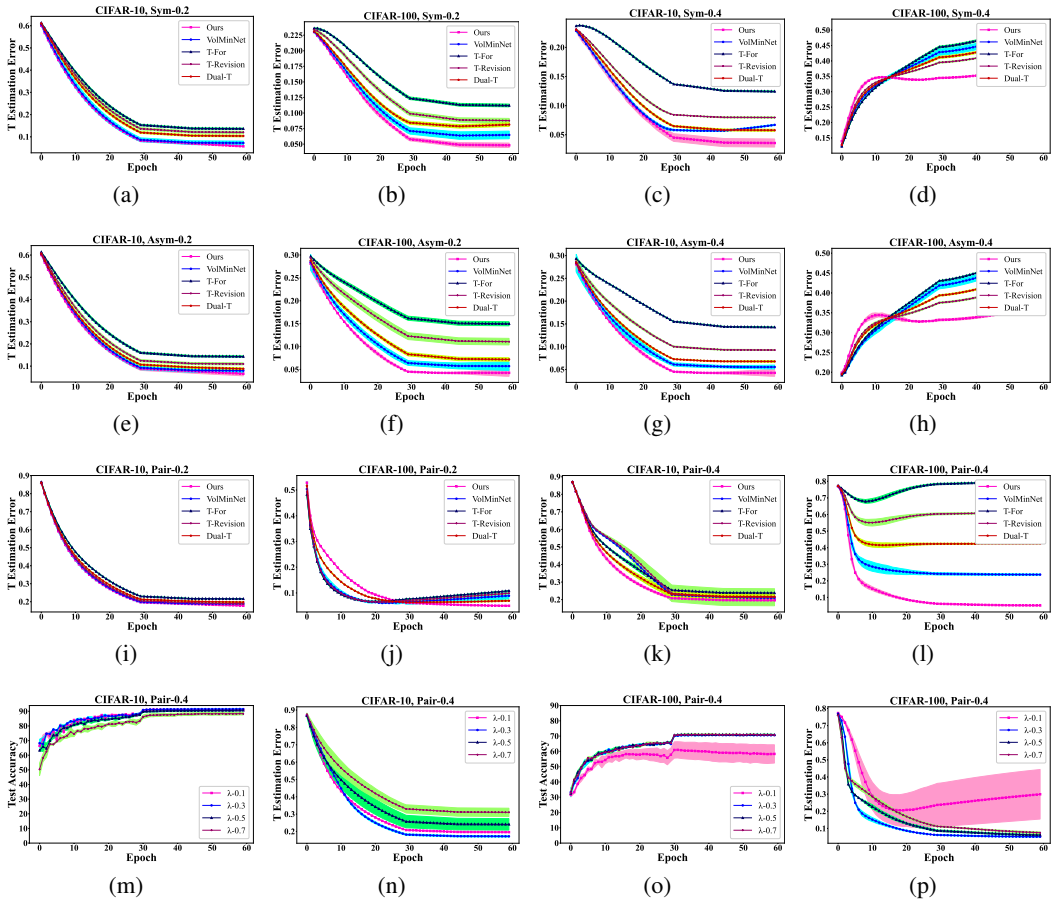

Figure 2: (a) to (l) compare the estimation error of T between our method and other transition matrix based methods with different noise types and noise ratios on two datasets. (m) to (p) show classification accuracy and T estimation error with various values of λ on two datasets.

transition matrix into the product of two easy-to-estimate transition matrices. VolMinNet [13] try to consistently estimate the transition matrix under the *sufficiently scattered* assumption, which empirically incorporates the minimum volume constraint of T into the label-noise learning. However, these methods relies heavily on the estimated inaccurate noisy class-posterior probability, which could lead the transition matrix to be poorly estimated.

5 Conclusion and Limitation

Estimating the transition matrix T plays an important role in building statistically consistent classifier for label-noise learning. To address the bottleneck of how to reduce the side-effects of the inaccurate noisy class posterior on transition matrix estimation, this paper creatively proposes to estimate T under the forward-backward cycle-consistency regularization, of which we could greatly reduce the dependency of estimating T on the noisy class posterior. We also show that the proposed method helps to minimize the *volume* of T without directly exploiting the estimated noisy class posterior, which further encourages the estimated transition matrix T to converge to its optimal solution. Extensive experimental results consistently justify the effectiveness of the proposed method, from both of the superior classification accuracy and the estimation error of the learned transition matrix.

Limitation. One major limitation in this study is that our method is constrained to work well under the *sufficiently scattered* assumption. When this assumption is violated in some cases, the statistical consistency of the estimated transition matrix and classifier would not be guaranteed. In the future, we will make deep theoretical analysis on the backward transition matrix and estimated clean class posterior probability.

References

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

[2] Yingbin Bai, Erkun Yang, Bo Han, Yanhua Yang, Jiatong Li, Yinian Mao, Gang Niu, and Tongliang Liu. Understanding and improving early stopping for learning with noisy labels. *arXiv preprint arXiv:2106.15853*, 2021.

[3] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.

[4] De Cheng, Tongliang Liu, Yixiong Ning, Nannan Wang, Bo Han, Gang Niu, Xinbo Gao, and Sugiyama. Masashi. Instance-dependent label-noise learning with manifold-regularized transition matrix estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

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

[6] 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. *arXiv preprint arXiv:1804.06872*, 2018.

[7] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5138–5147, 2019.

[8] 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 *International Conference on Machine Learning*, pages 2304–2313. PMLR, 2018.

[9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[10] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5447–5456, 2018.

[11] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

[12] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5051–5059, 2019.

[13] Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. Provably end-to-end label-noise learning without anchor points. *arXiv preprint arXiv:2102.02400*, 2021.

[14] Chang Liu, Han Yu, Boyang Li, Zhiqi Shen, Zhanning Gao, Peiran Ren, Xuansong Xie, Lizhen Cui, and Chunyan Miao. Noise-resistant deep metric learning with ranking-based instance selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6811–6820, 2021.

[15] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *arXiv preprint arXiv:2007.00151*, 2020.

[16] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

[17] Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International Conference on Machine Learning*, pages 6226–6236. PMLR, 2020.

[18] Yueming Lyu and Ivor W Tsang. Curriculum loss: Robust learning and generalization against label corruption. *arXiv preprint arXiv:1905.10045*, 2019.

[19] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pages 3355–3364. PMLR, 2018.

[20] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952, 2017.

[21] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[22] Clayton Scott. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *Artificial Intelligence and Statistics*, pages 838–846. PMLR, 2015.

[23] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560, 2018.

[24] Robert A Vandermeulen and Clayton D Scott. An operator theoretic approach to nonparametric mixture models. *The Annals of Statistics*, 47(5):2704–2733, 2019.

[25] Xiaobo Wang, Shuo Wang, Jun Wang, Hailin Shi, and Tao Mei. Co-mining: Deep face recognition with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9358–9367, 2019.

[26] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019.

[27] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13726–13735, 2020.

[28] Songhua Wu, Xiaobo Xia, Tongliang Liu, Bo Han, Mingming Gong, Nannan Wang, Haifeng Liu, and Gang Niu. Class2simi: A new perspective on learning with label noise. 2020.

[29] Yichen Wu, Jun Shu, Qi Xie, Qian Zhao, and Deyu Meng. Learning to purify noisy labels via meta soft label corrector. *AAAI*, 2021.

[30] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 33, 2020.

[31] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? *Advances in Neural Information Processing Systems*, 32:6838–6849, 2019.

[32] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015.

[33] Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *NeurIPS*, pages 6222–6233, 2019.

[34] Yan Yan, Rómer Rosales, Glenn Fung, Ramanathan Subramanian, and Jennifer Dy. Learning from multiple annotators with varying expertise. *Machine learning*, 95(3):291–327, 2014.

[35] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. *arXiv preprint arXiv:2006.07805*, 2020.

[36] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173. PMLR, 2019.

[37] Xiyu Yu, Tongliang Liu, Mingming Gong, Kayhan Batmanghelich, and Dacheng Tao. An efficient and provable approach for mixture proportion estimation using linear independence assumption. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4480–4489, 2018.

[38] Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *32nd Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[39] Zhaowei Zhu, Tongliang Liu, and Yang Liu. A second-order approach to learning with instance-dependent label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10113–10123, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

Optionally include extra information (complete proofs, additional experiments and plots) in the appendix. This section will often be part of the supplemental material.