Understanding and Mitigating the Bias in Sample Selection for Learning with Noisy Labels

Qi Wei[1], Lei Feng[2]*, Haobo Wang[3], Bo An[1]
[1]College of Computing and Data Science, Nanyang Technological University
[2]Information Systems Technology and Design, Singapore University of Technology and Design
[3]School of Software, Zhejiang University

Abstract

Learning with noisy labels aims to ensure model generalization given a label-corrupted training set. The sample selection strategy achieves promising performance by selecting a label-reliable subset for model training. In this paper, we empirically reveal that existing sample selection methods suffer from both data and training bias, which are represented in practice as imbalanced selected sets and accumulation errors. However, only the training bias was handled in previous studies. To address this limitation, we propose a noIse-Tolerant Expert Model (ITEM) for debiased learning in sample selection. Specifically, to mitigate the training bias, we design a robust Mixture-of-Expert network that conducts selection and learning on different layers. Compared with the prevailing double-branch network, our network performs better on both selection and prediction by ensembling multiple experts while training with fewer parameters. Meanwhile, to mitigate the data bias, we propose a weighted sampling strategy that assigns larger sampling weights to classes with smaller frequencies. Using MixUp, the model is trained on a mixture of two batches: one sampled by a weighted sampler and the other by a regular sampler, which mitigates the effect of the imbalanced training set while avoiding sparse representations that are easily caused by sampling strategies. Extensive experiments on seven noisy benchmarks and analyses demonstrate the effectiveness of ITEM. The code is released at `anonymous.4open.science/r/ITEM-18AB`.

1. Introduction

The remarkable generalization capability of deep neural networks (DNNs) is achieved through training on a large-scale dataset. However, existing training sets are usually collected by online queries [6], crowdsourcing [49], and manual annotations, which could inevitably incur wrong (or noisy) labels [41]. Since DNNs exhibit vulnerability to such low-quality training sets [54], training on the label-corrupted set presents a great challenge for the modern machine-learning community. To mitigate the adverse effect brought by noisy samples, learning with noisy labels (LNL) [38, 45] is important, contributing to improvements of the model's generalization on practical applications.

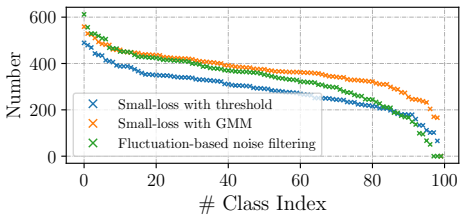
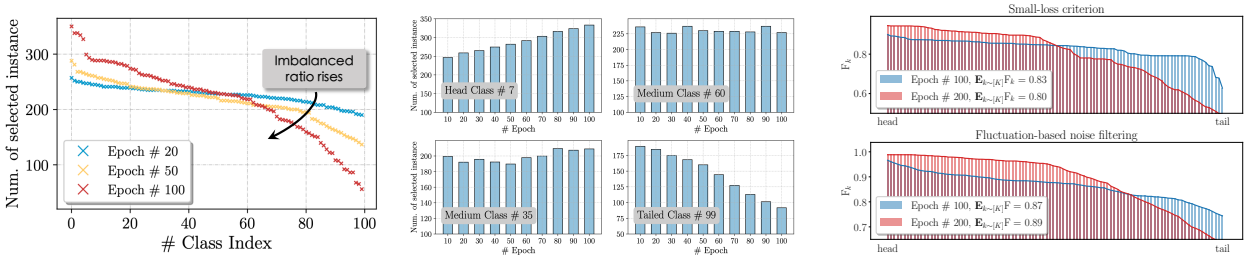


Figure 1. **Existing selection criteria always lead to an imbalanced training set**, termed as the data bias. A ResNet-34 is trained on CIFAR-100N. We visualize the class distribution of the selected set, given three typical selection criteria. The quantity of class-level samples in the last epoch is counted, while the index of classes is sorted. More results can be found in the Appendix A.

Sample selection [11, 21, 40], a prevailing strategy for LNL, achieves considerable performance in mitigating the effects of noisy labels [48] by carefully selecting clean samples from the label-corrupted training set. The performance of sample selection approaches is largely decided by the selection criteria, which can be roughly categorized into two sides: 1) *small-loss* based strategies [3, 11, 21, 26, 38, 52], which are motivated by the memorization effect [2] that DNNs learn simple patterns shared by majority examples before fitting the noise. Hence, the samples with small losses in the early learning stage can normally be taken as clean samples. 2) *fluctuation* based strategies [40, 45, 53, 58], which are motivated by the observation that DNNs easily give inconsistent prediction results for noisy samples. These methods [40, 53] normally consider an instance incorrectly labeled if its prediction results exhibit alterations within two consecutive training rounds or during a specified subsequent period.

*Corresponding author



(a) A corrupted selection tendency

(b) Class-level F-score

Figure 2. We train ResNet-18 on Sym. 60% CIFAR-100 with small-loss [11] and fluctuation-based noise filtering [40]. (**a**) Results on small-loss. *Left*: the class distribution of three training stages. *Right*: dynamic numbers of four representative categories. (**b**) Results on two selection criteria. Visualization of the class-level selection performance (F-score) F_k in 100- and 200-th epoch.

Despite the validated effectiveness, there is a common view shared by existing sample selection methods that the sample selection processes are only influenced by a training bias, i.e., the accumulated error. In this paper, we, for the first time, show that by extensive experiments, there exists another type of bias, i.e., the *data bias*, which is mainly caused by imbalanced data distribution. To be concrete, we split the bias in a selection-based learning framework into the training bias, which inherently exists in a self-training manner, and the data bias, which indicates the selected set tends to be class-imbalanced (see Figure 1). Moreover, we customize a class-level F-score and do extensive experiments in the next section, providing three nontrivial findings that reveal the adverse effect of the data bias. These observations motivate us to consider a novel selection-based learning framework that can simultaneously address training and data bias.

To this end, we propose a noIse-Tolerant Expert Model (ITEM), consisting of a noise-robust multi-experts structure and a weighted sampling strategy.

- Firstly, motivated by the idea of ensemble learning [9] and the Mixture-of-Expert model [16], we design a robust network structure integrating the classifier with multiple experts. Our network is more robust compared with existing robust networks like double-branch network [11] for the following reasons, 1) in our structure, the classifier learning and selection are disentangled, with experts selecting clean samples for the classifier's training, naturally mitigating the issue of error accumulation. 2) Ensembling selection results from different experts reduces harmful interference from incompatible patterns (*e.g.*, poor selection made by individual experts), leading to a cleaner selected set.
- Secondly, to mitigate the implicit data bias during selection, we propose a weighted resampling strategy named mixed sampling. By calculating the quantity of each class in the selected set, we assign larger weights to tail classes through a mapping function during sampling. Finanlly, introducing the MixUp [55], our network is trained on a mixed batch that combines two batches based on a regular and a weighted sampler, fulfilling class-balanced learning while avoiding the issue of sparse representations.

The promising performance of ITEM is verified on seven noisy benchmarks. Extensive ablation studies and analyses also demonstrate the effectiveness of each component.

2. Understanding the Bias in Sample Selection

In this section, we first introduce the formal problem setting of learning with noisy labels (LNL) and its learning goal. Then, we conduct experiments to deeply analyze where the bias of sample selection in LNL comes from.

Assume $\mathcal{X}$ is the feature space and $\mathcal{Y} = \{1, 2, \dots, K\}$ is the label space. Suppose the training set is denoted by $\tilde{D} = \{(\boldsymbol{x}_i, y_i)\}_{i\in[N]}$, where $[N] = \{1, 2, \dots, N\}$ is the set of indices. Since the annotator may give wrong labels in practice [18], the learner thus can only observe a label-corrupted set $D_N = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i\in[N]}$ with noisy labels.

As a prevailing strategy for LNL, sample selection [21, 24, 40, 45] aims to progressively select a reliable subset $D_M \subseteq D_N$ ($M < N$) and feed D_M to the classifier for training. We introduce $v_i \in \{0, 1\}$ to indicate whether the i-th instance is selected ($v_i = 1$) or not ($v_i = 0$). Generally, the performance of a selection-based method can be reflected by the deviation between the actual selected samples and the total clean samples, which can be measured by the F-score F, where $\mathsf{F} = \frac{2\cdot\mathsf{P}\cdot\mathsf{R}}{\mathsf{P}+\mathsf{R}}$, P and R denote selection precision and recall, respectively. In this paper, we individually calculate each class's F-score for further analysis. Specifically, the class-level F-score for class k is written as

$$\mathsf{F}_k = \frac{2\cdot\mathsf{P}_k\cdot\mathsf{R}_k}{\mathsf{P}_k+\mathsf{R}_k}, \text{ where } \begin{cases} \mathsf{P}_k = \frac{\sum_{i\in[N]} \mathbb{1}(v_i=1, y_i=\tilde{y}_i=k)}{\sum_{i\in[N]} \mathbb{1}(v_i=1, \tilde{y}_i=k)}, \\ \mathsf{R}_k = \frac{\sum_{i\in[N]} \mathbb{1}(v_i=1, y_i=\tilde{y}_i=k)}{\sum_{i\in[N]} \mathbb{1}(y_i=\tilde{y}_i=k)}. \end{cases}$$

By analyzing the selection performance under different training conditions, we have several nontrivial findings.

- *Current selection criteria easily result in an imbalanced selection subset.* As shown in Figure 1, we can see that

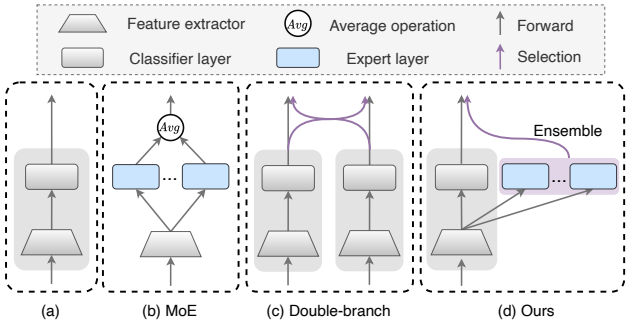


Figure 3. **Comparisons of different network structures**. **(a)** *Typical classification network*, which consists of a feature extractor and a classifier layer. **(b)** *Mixture-of-experts (MoE)* [29], a set of experts jointly gives the predicted label for the input. **(c)** *Double-branch robust structure*, the network is trained on a selected set which is considered clean by another network. **(d)** *Ours*, a mixture-of-experts module is integrated into the classification network, which works for robust selection as well as prediction ensemble.

both the two adopted selection strategies incur the imbalanced data distribution under two noisy conditions. The reason is that both the two selection criteria rely on model performance. However, the model has different capacities for different classes. For those classes with indistinguishable characteristics (i.e., rare classes), the model tends to produce a large loss or inconsistent results and further discards those samples.

- *Sample selection, conducted as self-training, will exacerbate the imbalanced ratio during the training process*. As shown in Figure 2 (a), the imbalanced ratio of the selected set increases as the training process proceeds, i.e., the number of selected instances increases in head classes and decreases in tail classes (see the right part of Figure 2). The intuitive reason is that the model's performance on tail classes hardly improves due to the limited number of available samples for training, which further degrades the effectiveness of selection criteria on these classes.
- *The selection performance is inherently influenced by the imbalanced class distribution.* As shown in Figure 2 (b), the selection performance of both two selection criteria increases in head classes but decreases in tail classes. The reason is that the selected errors are relatively small for categories with higher F-scores. Hence, the training of these classes would further improve the model's accuracy and selection performance. However, for tail classes with lower F-score, the self-training mechanism would further increase the selection error and thus fail to select instances in the follow-up phase correctly.

Based on these observations, we are motivated to design a model that confronts not only accumulated error but also the imbalanced selected set.

3. Methodology

Preliminaries. In LNL, we are given a label-corrupted training set $D_N = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i\in[N]}$ of N samples. A classifier with learnable parameters $f_{\boldsymbol{\theta}}$ is a function that maps from the input space $\mathcal{X}$ to the label space $f : \mathcal{X} \to \mathbb{R}^K$. In multi-class classification, we always update the parameter $\boldsymbol{\theta}$ by minimizing the following empirical risk:

$$\hat{R}(f) = \frac{1}{N}\sum\nolimits_{i=1}^{N} \mathcal{L}(f(\boldsymbol{x}_i, \boldsymbol{\theta}), \tilde{y}_i), \tag{1}$$

where $\mathcal{L}(\cdot)$ is the given loss function, e.g., Softmax Cross-Entropy (CE) loss. In sample selection strategy, previous works mainly focus on designing high-effectively selection criteria. Here, we give a universal training objective on the selected clean set, which is written as

$$\hat{R}_{\text{clean}}(f) = \frac{1}{N}\sum\nolimits_{i=1}^{N} \left[\text{Criterion}(f, \boldsymbol{x}_i, \tilde{y}_i) \cdot \mathcal{L}(f(\boldsymbol{x}_i, \boldsymbol{\theta}), \tilde{y}_i)\right],$$
$$\text{where Criterion}(\cdot) = \begin{cases} 1, & \text{If selected,} \\ 0. \end{cases} \tag{2}$$

In this unified formula, *the classifier intended to be optimized is consistent with the classifier serving as the noise filter during the selection phase*, which easily leads to the training bias. Empirical results in the section above show that directly optimizing $\hat{R}_{\text{clean}}(f)$ with an existing selection criterion is always biased, further resulting in inferior generalization performance.

Overview. We propose to fulfill debiased learning by handling both the training and data bias. First, to solve the training bias, we design a novel network architecture called noIse-Tolreant Expert Model (ITEM). Compared with the widely applied double-branch network, our proposal exhibits greater robustness to noisy labels. Second, to solve the data bias, we propose a weighted resampling strategy that can mitigate the side-effect caused by the class-imbalanced set D_{clean} while avoiding sparse representations.

3.1. ITEM: noIse-Tolerant Expert Model

The training bias in sample selection arises from the self-training approach. Previous works [11, 21] always resort to the double-branch network to confront this problem. Concretely, the network is trained on the clean set that is selected by the other network. Motivated by Mixture-of-experts [29, 31], we propose a robust architecture that integrates multiple experts into the classifier, which independently conducts classification and sample selection.

Specifically, compared with a normal classification network with a classifier layer f, our network ITEM contains f and a set of additional expert layers $\{g^1, ..., g^m\}$ with size m. Each expert layer has the same dimension as the classifier layer, *i.e.*, $\|f\| = \|g\|$. For robust selection, we propose to conduct existing criteria on expert layers instead of the

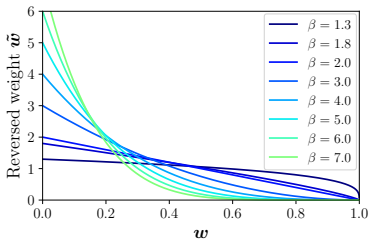


Figure 4. Mapping function $\mathcal{S}^{\beta}(\cdot)$ with different values of β.

classifier layer. Consequently, the training objective in our framework can be summarized as

$$\hat{R}_{\text{clean}}(f) = \frac{1}{N}\sum_{i=1}^{N}\big[\text{Criterion}(\{g^1,...,g^m\},\boldsymbol{x}_i,\tilde{y}_i)\cdot \mathcal{L}(f(\boldsymbol{x}_i,\boldsymbol{\theta}),\tilde{y}_i)\big]. \quad (3)$$

Intuitively, the independent selection phase prevents the classifier layer from selecting noisy samples and updating its parameters based on them, significantly mitigating the training bias. In practice, a voting strategy is applied to ensemble the selection results of m experts, where the sample selected by all experts is considered clean. It is noteworthy that the selection criterion $\text{Criterion}(\cdot)$ is not restricted to a specific form (e.g., small-loss based or fluctuation-based), which highlights the applicability of this expert model as a backbone in sample selection.

Advantages of the MoE structure. We compare different network structures in LNL in Figure 3. Compared with the prevailing double-branch networks, which have proved to be noise-robust in previous works [11, 21], our architecture offers two advantages, including 1) *Greater robustness to noisy labels*. Each expert exhibits different capabilities during training, resulting in a better inductive bias for both selection and prediction. Specifically, by ensembling the selections from different experts, the MoE minimizes negative interference from incompatible patterns (e.g., poor selection made by individual experts), thereby enhancing the framework's robustness against noisy labels. 2) *Resource-friendly training*. In contrast to the feature extractor's huge parameters, the last layer's parameters for classification are significantly fewer. The additional learnable parameters in our model are less than 2% of those in the double-branch network. Thus, our network is more friendly to GPU and computer memory.

3.2. Beta-Resampling

To overcome the data bias, we propose a mixed beta-sampling strategy that promotes the sampling frequency for tail classes in the selected clean set when sampling. Thus, the model can learn from more knowledge from the tailed classes, contributing to mitigating the data bias.

Specifically, suppose a clean set D_{clean} is selected from the D_N, where $D_{\text{clean}} \subseteq D_N$. The set of samples labeled as class k in D_{clean} is D^k_{clean}. By introducing L1 normalization, a vector of class frequencies $\boldsymbol{v}$ can be obtained,

$$\boldsymbol{v} = [w_1, w_2, ...w_K], \text{ where } w_k = \frac{\|D^k_{\text{clean}}\|}{\|D_{\text{clean}}\|}. \quad (4)$$

Here, we can regard the class frequency as the sampling weight, *i.e.*, the class with a larger frequency will be sampled with a greater weight during the data loading phase.

To mitigate the data bias, which leads the model to learn from the majority classes mainly, we propose a mapping function that maps the lower frequency to a higher weight when data sampling. Concretely, this mapping function stems from the probability density function (PDF) of a Beta distribution. Normally, the Beta distribution's PDF is controlled by two parameters α, β. To satisfy both 1) the input interval of w_k belongs to $[0, 1]$ and 2) the function contains the monotonically decreasing property, we thus fix $\alpha = 1$ and adjust β for different slopes. Given a weight w_i, the reversed weight $\tilde{w}_i$ is written as

$$\tilde{w}_i = \mathcal{S}^{\beta}(w_i) = \frac{1}{\text{B}(1,\beta)}(1-w_i)^{\beta-1}, \quad (5)$$

where β is a hyper-parameter and $\text{B}(\cdot)$ denotes a beta function written as $\text{B}(1,\beta) = \int_0^1 (1-t)^{\beta-1}dt$.

The mapping function $\mathcal{S}^{\beta}(\cdot)$ for varying values of β is illustrated in Figure 4. It can be observed that as the input weight increases, the output value decreases. Consequently, this mapping function enables the transformation of class frequencies in Eq. 4 into a tail-focused weighted vector $\tilde{\boldsymbol{v}} = [\tilde{w}_1, \ldots, \tilde{w}_K]$. By weighted sampling with $\tilde{\boldsymbol{v}}$, a tail-classes focused training batch $B_{\tilde{\boldsymbol{v}}}$ can be obtained. Training the model on $B_{\tilde{\boldsymbol{v}}}$ can effectively mitigate the data bias.

3.3. Stochastic Classifier Training

Since the noise filter and the optimizer are disentangled in Eq. (3), largely mitigating the training bias, the parameter update of expert layers is unreachable. Considering almost all existing selection criteria rely on the model's performance, dynamically updating the parameters of expert layers is necessary. To fulfill this goal, we propose a stochastic training strategy, which randomly assigns a layer from the set of $\mathbb{F} = \{f, g^1, ..., g^m\}$ as the classifier layer f^c and the rest are expert layers. Hence, Eq. (3) can be rewritten as

$$\hat{R}_{\text{clean}}(f^c) = \frac{1}{N}\sum_{i=1}^{N}\big[\text{Criterion}(\mathbb{F}\backslash f^c,\boldsymbol{x}_i,\tilde{y}_i)\cdot \mathcal{L}(f^c(\boldsymbol{x}_i,\boldsymbol{\theta}),\tilde{y}_i)\big]. \quad (6)$$

Since the aim-to-optimized layer f^c is initialized from the set $\mathbb{F}$ in each iteration, the performance of each layer can be ensured under sufficient training processes.

Despite a highly integrated framework in Eq. (6) proposed, the training manner is practically decoupled into

Table 1. Test accuracy (mean±std) of methods using ResNet-18/34 on CIFAR-10/100. Note that †, ‡ and ♯ denote three selection criteria, *i.e.*, small-loss selection with loss threshold [11], small-loss selection with Gaussian Mixture Model [21], and fluctuation-based noise filtering [40]. **Bold** values denote the best the second best performance.

	Methods	Sym. 20%	Sym. 40%	Inst. 20%	Inst. 40%	*Avg.*
CIFAR-10	Cross-Entropy	85.00 ± 0.43%	79.59 ± 1.31%	85.92 ± 1.09%	79.91 ± 1.41%	82.61
	JoCoR [38]	88.69 ± 0.19%	85.44 ± 0.29%	87.31 ± 0.27%	82.49 ± 0.57%	85.98
	Joint Optim [35]	89.70 ± 0.36%	87.79 ± 0.20%	89.69 ± 0.42%	82.62 ± 0.57%	87.45
	CDR [44]	89.68 ± 0.38%	86.13 ± 0.44%	90.24 ± 0.39%	83.07 ± 1.33%	87.28
	Me-Momentum [3]	91.44 ± 0.33%	88.39 ± 0.34%	90.86 ± 0.21%	86.66 ± 0.91%	89.34
	PES [4]	92.38 ± 0.41%	87.45 ± 0.34%	92.69 ± 0.42%	89.73 ± 0.51%	90.56
	Late Stopping [53]	91.06 ± 0.22%	88.92 ± 0.38%	91.08 ± 0.23%	87.41 ± 0.38%	89.62
	† Co-teaching [11]	87.16 ± 0.52%	83.59 ± 0.28%	86.54 ± 0.11%	80.98 ± 0.39%	84.56
	† ITEM (Ours)	93.79 ± 0.14%	90.83 ± 0.19%	93.52 ± 0.14%	91.09 ± 0.18%	92.31
	‡ ITEM (Ours)	**95.01 ± 0.21%**	**93.10 ± 0.20%**	**95.18 ± 0.19%**	**93.65 ± 0.12%**	**94.24**
	♯ SFT [40]	92.57 ± 0.32%	89.54 ± 0.27%	91.41 ± 0.32%	89.97 ± 0.49%	90.87
	♯ ITEM (Ours)	**95.26 ± 0.23%**	**92.81 ± 0.20%**	**95.80 ± 0.18%**	**93.13 ± 0.29%**	**94.25**
CIFAR-100	Cross-Entropy	57.59 ± 2.55%	45.74 ± 2.61%	59.85 ± 1.56%	43.74 ± 1.54%	51.73
	JoCoR [38]	64.17 ± 0.19%	55.97 ± 0.46%	61.98 ± 0.39%	50.59 ± 0.71%	58.17
	Joint Optim [35]	64.55 ± 0.38%	57.97 ± 0.67%	65.15 ± 0.31%	55.57 ± 0.41%	60.81
	CDR [44]	66.52 ± 0.24%	60.18 ± 0.22%	67.06 ± 0.50%	56.86 ± 0.62%	62.65
	Me-Momentum [3]	68.03 ± 0.53%	63.48 ± 0.72%	68.11 ± 0.57%	58.38 ± 1.28%	64.50
	PES [4]	68.89 ± 0.41%	64.90 ± 0.57%	70.49 ± 0.72%	65.68 ± 0.44%	67.49
	Late Stopping [53]	68.67 ± 0.67%	64.10 ± 0.40%	68.59 ± 0.70%	59.28 ± 0.46%	65.16
	† Co-teaching [11]	59.28 ± 0.47%	51.60 ± 0.49%	57.24 ± 0.69%	45.69 ± 0.99%	53.45
	† ITEM (Ours)	72.42 ± 0.17%	71.96 ± 0.24%	73.61 ± 0.16%	69.90 ± 0.30%	71.97
	‡ ITEM (Ours)	**78.20 ± 0.09%**	**75.27 ± 0.20%**	**77.91 ± 0.14%**	**70.69 ± 0.31%**	**75.51**
	♯ SFT [40]	71.98 ± 0.26%	69.72 ± 0.31%	71.83 ± 0.42%	69.91 ± 0.54%	70.86
	♯ ITEM (Ours)	**77.19 ± 0.13%**	**74.90 ± 0.24%**	**76.91 ± 0.23%**	**71.44 ± 0.29%**	**75.11**

three stages that are proceeding iteratively, 1) select clean samples via the preset criterion based on the ensemble result of all experts, 2) randomly select an optimization layer, and 3) train this layer on the selected set.

Considering training on the tail-focused training batch directly would result in the sparse representation of head classes [57], we separately conduct two times weighted samples (according to the weighted and reversed weighted vector $\boldsymbol{v}, \tilde{\boldsymbol{v}}$) in each data-loading phase, which focus on head and tail classes, respectively. By leveraging the MixUp strategy [55], the model can learn a more representative feature extractor on the mixed data. Specifically, in each training iteration, we respectively sample a head-focused batch $B_{\boldsymbol{v}} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^{b}$ and a tail-focused batch $B_{\tilde{\boldsymbol{v}}} = \{(\boldsymbol{x}'_i, \tilde{y}'_i)\}_{i=1}^{b}$ from $\tilde{D}_{\text{clean}}$ according to $\tilde{\boldsymbol{v}}$, where b denotes the size of the mini-batch. By randomly selecting a layer from $\mathbb{F}$ as the optimized-classifier layer f^c, the parameter's update can be achieved by minimizing

$$L^{\text{train}} = \frac{1}{b}\sum_{i=1}^{b} \mathcal{L}\big(f^c\big(\text{MixUp}(\boldsymbol{x}_i, \boldsymbol{x}'_i)\big),\ \text{MixUp}(\tilde{y}_i, \tilde{y}'_i)\big), \tag{7}$$

where $\text{MixUp}(,)$ is represented as $\text{MixUp}(a, b) = \gamma \cdot a + (1-\gamma)\cdot b$ and γ is a trade-off coefficient randomly sampled from a beta distribution $\texttt{beta}(\delta, \delta)$. The algorithm flowchart of ITEM is shown in Algorithm 1 (see Appdx).

4. Experiments

Datasets. We assess the performance of *ITEM* on two noise-synthetic datasets CIFAR-10 and CIFAR-100 [18], two human-annotated datasets CIFAR10N and CIFAR100N [39], and three real-world noisy benchmarks including Clothing-1M [47], Food-101N [20], and Webvision [23].

Baselines. On CIFAR-10 and CIFAR-100, we compare ITEM with prevailing methods that can be roughly divided into three parts, including 1) *small-loss based selection*, JoCoR [38], and Me-Momentum [3], 2) *fluctuation-based selection*, SFT [40] and Late Stopping [53], 3) others, Cross-Entropy, Joint Optim [35], and PES [4]. Reported results in this paper are collected from SFT and Late Stopping. More information is shown in Appdx.

Implementation details. Our code implements utilize Pytorch 1.9.0 and all experiments are run on a single RTX 4090 GPU. We keep the convention from [40, 53] and adopt ResNet-18 and ResNet-34 for CIFAR-10 and CIFAR-100, respectively. For all noisy conditions on CIFAR-10 & 100, we leverage an SGD optimizer with the momentum 0.9 and

Table 2. Test accuracy (%) of prevailing methods using ResNet-34 on CIFAR10N and CIFAR100N. Note that ✓ and ✗ indicate whether a *semi-supervised framework* is used or not.

Methods		CIFAR10N				CIFAR100N
		Worst	R1	R2	R3	
Co-teaching+ [52]	✗	83.26	89.70	89.47	89.54	60.37
Peer Loss [27]	✗	82.00	89.06	88.76	88.57	57.59
CAL [59]	✗	85.36	90.93	90.75	90.74	61.73
Late Stoping [53]	✗	85.27	-	-	-	-
‡ ITEM	✗	**91.15**	**95.12**	**95.03**	**95.17**	**69.47**
DivideMix [21]	✓	92.56	95.16	95.23	95.21	71.13
ELR+ [26]	✓	91.09	94.43	94.20	94.34	66.72
CORES* [8]	✓	91.66	94.45	94.88	94.74	61.15
DPC [60]	✓	**93.82**	95.97	95.92	95.90	71.42
‡ ITEM*	✓	93.14	**96.08**	**96.54**	**96.71**	**72.40**

Table 3. Test accuracy (%) comparison of previous methods on three real-world noisy benchmarks. Top-1 and Top-5 test accuracy are reported for the dataset WebVision.

Methods	Food101N	Clothing1M	WebVision	
			Top-1	Top-5
Co-teaching [11]	83.73	67.94	63.58	85.20
JoCoR [38]	84.04	69.06	63.33	85.06
CDR [44]	86.36	66.59	62.84	84.11
ELR+ [26]	85.77	74.81	77.78	91.68
DivideMix [21]	86.73	74.76	77.32	91.64
SFT+ [40]	-	75.00	77.27	91.50
CoDis* [46]	86.88	74.92	77.51	91.95
SURE [25]	88.00	**75.10**	78.94	92.34
‡ ITEM*	**88.14**	75.08	**80.20**	**93.07**

the weight-decay $1e-3$ to train our network. The total training epoch is set as 200. The initial learning rate is 0.02 and decayed with the factor 10 at the 100-th and 150-th epoch. For real-world noise, we adopt a ResNet-34 for CIFAR-N and a pre-trained ResNet-50 for Clothing-1M and Food101N, and Inception-ResNet-V2 [34] for WebVision. Other training set-ups like the number of training epochs, the learning rate and its adjustment, the optimizer, and so on keep the same as [25].

Hyper-parameters. Our framework contains two main hyper-parameters, *i.e.*, the number of expert m in the network architecture, and the slope parameter β in the mapping function. We keep $\beta = 2$ for all experiments and set $m = 4$ for CIFAR and $m = 2$ for three open-world datasets.

4.1. Results on Synthetic Noisy Datasets

We conduct comparison experiments with two synthetic noise types on CIFAR-10 & 100, where we manually construct different noisy types on these two datasets. Since our framework is agnostic to selection criteria, we test our proposal on three types of selection criteria. Experimental results are shown in Table 1.

On both CIFAR-10 and CIFAR-100, our method ITEM consistently achieves state-of-the-art performance in all noisy settings. Compared with other section-based methods such as Co-teaching and SFT, ITEM obtains obvious performance improvements while adopting their selection criteria. To be specific, †ITEM averagely improves the test accuracy of Co-teaching by 9.68% on CIFAR-10 and 22.06% on CIFAR-100. Compared with SFT [40], a representative method in fluctuation-based selection, ITEM also achieves remarkable improvements.

4.2. Results on Real-World Noisy Datasets

We test ITEM's performance on two human-made noisy datasets and three website noisy datasets. Considering that semi-supervised learning can utilize the sample discarded by the sample selection framework to regularize the model's learning [21], we integrate ITEM with the Pseudo-labeling technique [19] to further improve the performance.

1) *Results on CIFAR10N & 100N*. The results are shown in Table 2. First, our method achieves remarkable performance on both two settings (*w/* or *w/o* SSL). When training without SSL, ITEM outperforms previous methods by a large margin. Compared with Late Stopping, a newly proposed method, ITEM achieves 5.88% improvements on *worst*-labels. When training with SSL, ITEM is superior to the previous SOTA method, DivideMix. Even training without SSL, ITEM surprisingly gains greater performance than those methods that leverage SSL approaches.

2) *Results on Food101N, Clothing1M, and WebVision*. The results are shown in Table 3. On Food101N, our proposal achieved the best performance, outperforming SURE (the method with the second-best performance) [25] by 0.14%. On Clothing1M, the performance of our proposal is not the best but considerably competitive. Compared with the best performance, the 75.1% test accuracy, ITEM obtained a closed score, *i.e.*, 75.08%. On WebVision, we keep the convention from the work [46] that tests the performance on the validation set of WebVision. ITEM significantly improved Top-1 accuracy by nearly 1.3%.

The result verifies that our method effectively enhances model generalization in large-scale real-world scenarios.

4.3. More Analyses

Debias learning for LNL. We visualize the test accuracy in each class to demonstrate that ITEM achieves relatively balanced performance on varying categories while fulfilling greater generalization. We plot comparison results in Figure 5. First, ITEM achieves almost unbiased prediction results compared to *CE (clean)*, which is trained on 50k clean samples. Second, compared with existing methods, ITEM takes tail classes better into account since two weighted samplers. Therefore, ITEM performs better in these categories. Be-

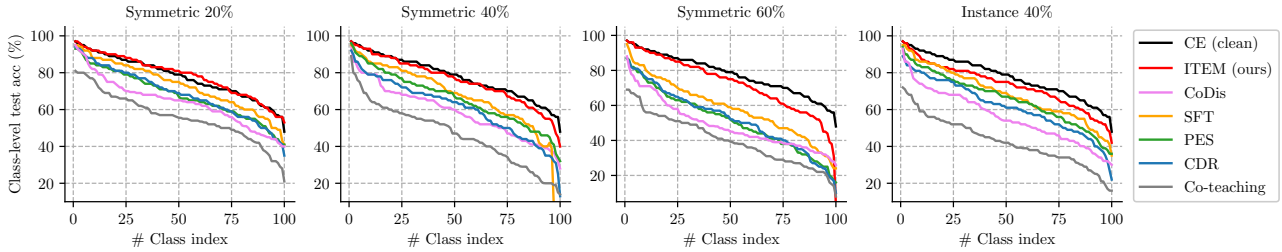
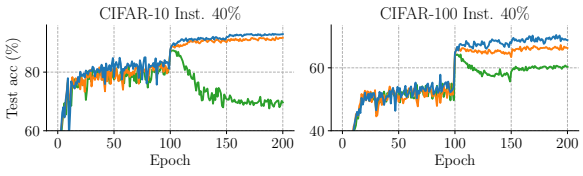
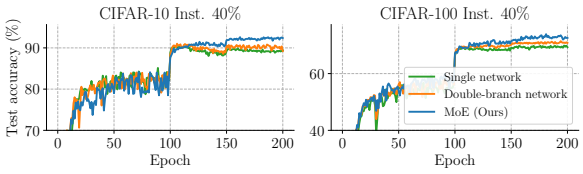


Figure 5. **Visualization of debias learning** in a class-level. We selected a ResNet-18 as the backbone and compared class-level prediction results of various methods on CIFAR-100 with four noise types. "CE (clean)" denotes training the model on the completely clean set (50k samples in total). The index of classes is sorted according to the class-level accuracy.



(a) Small-loss selection [11]



(b) Fluctuation-based selection [40]

Figure 6. **Robustness comparison** of different networks under two selection criteria. Our proposed MoE structure significantly mitigates the adverse effect of noisy labels.

sides, under 20% symmetric label noise, ITEM achieved better performance than the model trained on an absolutely clean set (see the red line *vs.* the black line). These results demonstrate that our approach has significant potential to address the implicit bias inherent in sample selection frameworks for noisy label learning.

Robustness of MoE structure. Compared with existing robust network, our proposal MoE exhibits better robustness. To clearly show this merit, we evaluate two selection criteria and plot the training curve under three networks in Figure 6. For the small-loss criterion, which easily incurs accumulation errors (see the green line (a)), the performance improvement with the MoE structure is significant. In contrast to the double-branch structure, which experiences slight performance degradation when the learning rate decreases at the 100th epoch (see the orange line in (b)), the learning curve with the MoE structure is more robust and stable. Second, in terms of fluctuation-based selection, the MoE demonstrates greater robustness, more stable training curves, and better generalization performance in noisy settings. In addition, its plug-in plug-out capability further verifies the versatility of the MoE structure.

Table 4. **Ablation studies** of each component in ITEM with varying noise conditions.

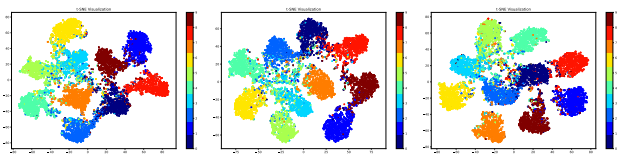
Methods	CIFAR-10 Sym 40%	CIFAR-100 Sym 40%	CIFAR10N Worst	CIFAR100N Fine
w/o MoE Network	90.75	73.01	89.72	68.19
w/o Mixed Sampling	88.77	71.09	87.66	68.24
w/o MixUp	91.19	72.50	89.90	69.10
Ours	**92.81**	**74.90**	**91.15**	**69.72**

4.4. Ablation Studies

Effect of each component. Our framework mainly contains two modules: the robust network architecture and the mixed data sampling strategy. We conduct ablation studies on CIFAR benchmarks to evaluate the effectiveness of each component from the following two perspectives.

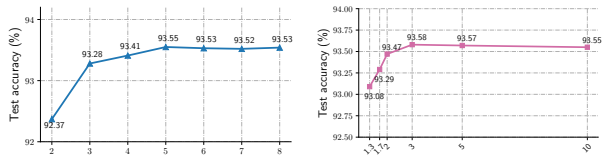
1) Quantitative analysis. We split the main component in ITEM into three parts: robust MoE, Mixed sampling, and the MixUp strategy. By solely removing them from ITEM, we reported the performance of the model in Table 4. Compared with typical ResNet architecture, our expert-based network indeed makes an obvious contribution to mitigating the noise. The average improvement among five settings roughly reaches 2%. Second, the mixed sampling strategy also promotes the performance of our network. Compared with training on randomly sampled batches, training on two weighted sampled batches gains greater generalization.

2) Representation visualization. Considering that training with different sampling leads the model to different representation performance, we compare the learned representation space of the test set of CIFAR-10 and show the results in Figure 7. We can see that the learned representation of ten classes on the normal sampling B is incompact, while the decision boundary is not well generalized. By contrast, if the proposed reversed tail-focused batch $B_{\tilde{v}}$ is adopted, each cluster is more compact where the distance between each cluster is larger, which verifies the effectiveness of our proposal sampling strategy in mitigating data bias underlying the selection phase. Eventually, by leveraging the Mixup operation, a feature extractor with better representation performance is obtained, as shown in the last plot.



(a) $acc = 88.94\%$ (b) $acc = 91.29\%$ (c) $acc = 92.97\%$

Figure 7. Visualization of feature representations on the CIFAR-10 test set by T-SNE [36] with **different sampling strategies**. (**a**) Training on randomly sampled batch B, (**b**) Training on $B_{\tilde{v}}$, and (**c**) Training on $\texttt{MixUp}(B_{\boldsymbol{v}}, B_{\tilde{\boldsymbol{v}}})$.



(a) Number of experts m (b) The slope parameter β

Figure 8. **Sensitivity analyses** of two hyper-parameters in ITEM. A ResNet-18 is trained on CIFAR-10 with 40% sym. noisy labels.

Sensitivity analyses. ITEM has two main hyper-parameters, m and β. We conduct ablation studies to select optimal values for experiments. The result is shown in Figure 8. First, the number of expert layers m is the most important parameter in our framework, which decides the overall network structure and computation costs. From the right figure, we can observe that the performance of ITEM gradually increases with the value of m increases. When arriving at a peak, i.e., 93.55% top-1 accuracy at $m = 5$, the performance will not increase if the value of m becomes larger. Therefore, we recommend a larger value of m given an unknown dataset. Second, the value of β has slightly influenced the performance of our method, where we also prefer a larger value of β.

5. Related Work

There are three main research lines in sample selection against the incorrect information from noisy labels.

1) Robust selection criteria. Previous sample selection methods normally exploit the memorization effect of DNNs, i.e., DNNs first memorize training data with clean labels and then those with noisy labels [11, 44]. The small-loss criterion [11] is a typical method stemming from the memorization effect, which splits the noisy training set via a loss threshold. The samples with small losses are regarded as clean samples. Some improved criteria [1, 21] based on the small-loss criterion are proposed in later works. In addition to the family of small loss criterion, some methods motivated by prediction fluctuation are designed. These methods [45, 58] observe that the model tends to give inconsistent prediction results for noisy samples and thus filter noisy labels by identifying high-frequency fluctuation samples.

2) Robust network architecture. Easily works based on constructing the noise transition matrix always resort to an additional adaptation layer at the top of the softmax layer [14] or design a new dedicated architecture [10, 50]. Recently, a family of double-branch networks [11, 21, 38, 52] has been proposed, which integrates two networks with the same architecture and selects clean samples for another network. Besides, better performance would be achieved when ensembling prediction results from two networks [11].

3) Robust training procedures. Overall, current methods in sample selection always resort to a self-training manner, i.e., iteratively conducting clean label selection and model retraining. However, the incorrect selection would degrade the subsequent model learning. Recently, some robust training manners were proposed. Concretely, [58] designed a framework based on curriculum learning [5], which starts with learning from clean data and then gradually moves to learn noisy-labeled data with pseudo labels. There are also some works [32, 51] utilizing active learning to gradually modify the label of noisy samples.

Relations to us. The advantages of our proposal compared with previous methods can be divided into the following three parts, which are summarized in Table 5 in Appendix.

- *Flexibility of the network structure.* To our knowledge, we are the first to apply a mixture-of-experts (MoE) structure to LNL. Despite the core idea of MoE and double-branch structure being similar (*i.e.*,ensembling prediction results and the selection result over different views), MoE exhibits more flexibility. Expanding the number of experts is simple and only increases slight parameter counts (0.01 million per expert). By contrast, the double-branch structure is hardly extended to a three- or four-branch structure since the number of learnable parameters.
- *Expansibility of the framework.* Compared with previous selection-based methods, our proposal is agnostic to the selection criteria, which can be easily integrated with different selection criteria and improves their performance.
- *Debias learning.* Previous mainly focus on the accumulation error, a training bias. In this paper, we empirically show the existence of data bias (*i.e.*, the imbalanced dataset selected by current selection criteria) and design an expert-based sampling strategy, which mitigates this bias by adjusting selection probabilities and increasing the visibility of underrepresented classes.

6. Conclusion

In this paper, we disclosed the data bias, an implicit bias underlying the sample selection strategy. To solve the training and data bias simultaneously, we proposed ITEM that introduces a noise-robust multi-experts network and a mixed sampling strategy. First, our structure integrates the classifier with multiple experts and leverages expert layers to conduct selection independently. Compared with the prevailing

double-branch network, it exhibits great potential in mitigating the training bias, i.e., the accumulated error. Second, our mixed sampling strategy yields class-aware weights and further conducts weighted sampling. The effectiveness of ITEM in real applications is verified on diverse noise types and both synthetic and real-world datasets.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *ICML*, 2019. 8, 14

[2] Devansh Arpit, Stanislaw Jastrzkebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *ICML*, 2017. 1

[3] Yingbin Bai and Tongliang Liu. Me-momentum: Extracting hard confident examples from noisily labeled data. In *ICCV*, 2021. 1, 5

[4] 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. In *NeurIPS*, 2021. 5

[5] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009. 8

[6] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *JACM*, 2003. 1

[7] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014. 12

[8] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. In *ICLR*, 2021. 6, 12

[9] Xibin Dong, Zhiwen Yu, Wenming Cao, Yifan Shi, and Qianli Ma. A survey on ensemble learning. *FCS*, 2020. 2

[10] Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. Masking: A new perspective of noisy supervision. In *NeurIPS*, 2018. 8

[11] 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*, 2018. 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14

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

[13] Shenwang Jiang, Jianan Li, Ying Wang, Bo Huang, Zhang Zhang, and Tingfa Xu. Delving into sample loss curve to embrace noisy and imbalanced data. In *AAAI*, 2022. 12, 13

[14] Ishan Jindal, Matthew Nokleby, and Xuewen Chen. Learning deep networks from noisy labels with dropout regularization. In *ICDM*, 2016. 8

[15] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *CVPR Workshop*, 2011. 14

[16] Sotiris B Kotsiantis, Ioannis D Zaharakis, and Panayiotis E Pintelas. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 2006. 2

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 11

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 2, 5

[19] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop*, 2013. 6

[20] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *CVPR*, 2018. 5, 12

[21] Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, 2020. 1, 2, 3, 4, 5, 6, 8, 11, 12, 14

[22] Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu. Selective-supervised contrastive learning with noisy labels. In *CVPR*, 2022. 12, 13

[23] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv:1708.02862*, 2017. 5, 12

[24] Yifan Li, Hu Han, Shiguang Shan, and Xilin Chen. Disc: Learning from noisy labels via dynamic instance-specific selection and correction. In *CVPR*, 2023. 2

[25] Yuting Li, Yingyi Chen, Xuanlong Yu, Dexiong Chen, and Xi Shen. Sure: Survey recipes for building reliable and robust deep networks. In *CVPR*, 2024. 6, 12

[26] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *NeurIPS*, 2020. 1, 6, 12

[27] Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *ICML*, 2020. 6, 12

[28] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *ICML*, 2020. 14

[29] Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 2014. 3

[30] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv:1701.06548*, 2017. 14

[31] Lior Rokach. Ensemble-based classifiers. *Artificial intelligence review*, 2010. 3

[32] Burr Settles. Active learning literature survey. 2009. 8

[33] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 11

[34] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017. 6

[35] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, 2018. 5

[36] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008. 8

[37] Haobo Wang, Ruixuan Xiao, Yiwen Dong, Lei Feng, and Junbo Zhao. Promix: Combating label noise via maximizing clean sample utility. In *IJCAI*, 2022. 12

[38] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, 2020. 1, 5, 6, 8, 12, 14

[39] Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. *arXiv:2110.12088*, 2021. 5, 11, 12

[40] Qi Wei, Haoliang Sun, Xiankai Lu, and Yilong Yin. Self-filtering: A noise-aware sample selection for label noise with confidence penalization. In *ECCV*, 2022. 1, 2, 5, 6, 7, 11, 12

[41] Qi Wei, Lei Feng, Haoliang Sun, Ren Wang, Chenhui Guo, and Yilong Yin. Fine-grained classification with noisy labels. In *CVPR*, 2023. 1, 14

[42] Tong Wei, Jiang-Xin Shi, Wei-Wei Tu, and Yu-Feng Li. Robust long-tailed learning under label noise. *arXiv:2108.11569*, 2021. 12, 13

[43] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. Technical report, 2010. 14

[44] Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *ICLR*, 2020. 5, 6, 8, 12

[45] Xiaobo Xia, Tongliang Liu, Bo Han, Mingming Gong, Jun Yu, Gang Niu, and Masashi Sugiyama. Sample selection with uncertainty of losses for learning with noisy labels. In *ICLR*, 2022. 1, 2, 8

[46] Xiaobo Xia, Bo Han, Yibing Zhan, Jun Yu, Mingming Gong, Chen Gong, and Tongliang Liu. Combating noisy labels with sample selection by mining high-discrepancy examples. In *ICCV*, 2023. 6, 12

[47] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015. 5, 11

[48] 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*, 2019. 1

[49] Yan Yan, Rómer Rosales, Glenn Fung, Ramanathan Subramanian, and Jennifer Dy. Learning from multiple annotators with varying expertise. *MLJ*, 2014. 1

[50] Jiangchao Yao, Jiajie Wang, Ivor W Tsang, Ya Zhang, Jun Sun, Chengqi Zhang, and Rui Zhang. Deep learning from noisy image labels with quality embedding. *IEEE TIP*, 2018. 8

[51] Taraneh Younesian, Zilong Zhao, Amirmasoud Ghiassi, Robert Birke, and Lydia Y Chen. Qactor: Active learning on noisy labels. In *ACML*, 2021. 8

[52] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *ICML*, 2019. 1, 6, 8, 12

[53] Suqin Yuan, Lei Feng, and Tongliang Liu. Late stopping: Avoiding confidently learning from mislabeled examples. In *ICCV*, 2023. 1, 5, 6, 12

[54] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017. 1

[55] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv:1710.09412*, 2017. 2, 5

[56] Manyi Zhang, Xuyang Zhao, Jun Yao, Chun Yuan, and Weiran Huang. When noisy labels meet long tail dilemmas: A representation calibration method. In *ICCV*, 2023. 12, 13

[57] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *CVPR*, 2020. 5

[58] Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Robust curriculum learning: From clean label detection to noisy label self-correction. In *ICLR*, 2020. 1, 8

[59] Zhaowei Zhu, Tongliang Liu, and Yang Liu. A second-order approach to learning with instance-dependent label noise. In *CVPR*, 2021. 6, 12

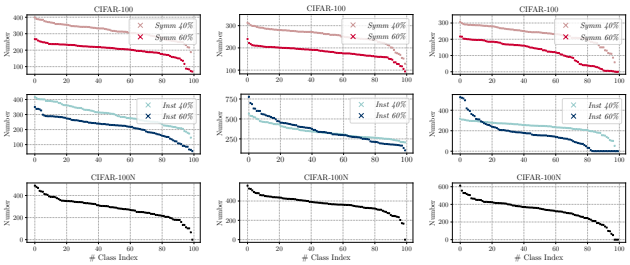
[60] Chen-Chen Zong, Ye-Wen Wang, Ming-Kun Xie, and Sheng-Jun Huang. Dirichlet-based prediction calibration for learning with noisy labels. In *AAAI*, 2024. 6, 12

Supplementary materials

A. Visualization of Data Bias

In this section, we conducted extensive experiments that discover the existence of the data bias, *e.g.*, the imbalanced subset is selected by current selection-based frameworks. Despite the Figure 1 in the section of Introduction, we show more results in the following figure (See Figure 9).

We can observe that *data bias widely exists in selection-based frameworks no matter different noise types and different datasets.*



(a) (b) (c)

Figure 9. **Extended results** of Figure 1. (a) Small-loss with threshold [11], (b) Small-loss with GMM [21], and (c) Fluctuation-based noise filtering [40].

B. Algorithm of ITEM

We provided the pseudocode for ITEM in Algorithm 1.

ITEM with semi-supervised learning. Considering the large scale of unreliable samples discarded, especially in considerable noise ratios, leveraging these samples through semi-supervised learning (SSL) to regularize the model is necessary as well as effective. *Thanks to the flexible training framework, any SSL method can be integrated into our proposal ITEM.*

Here, we provide a novel version of ITEM with stronger performance. The algorithm is shown in Algorithm 2. Besides, a strong augmentation `CTAugment` is used, which refers to [33]. Compared with its original version, the new training framework only adds two operations (shown in Lines 12 and 13). Thus, the code implementation is relatively convenient.

C. Experimental Settings

In this section, we reported the statistical data of seven datasets utilized in this paper and the whole experimental setup in Table 6.

Algorithm 1 ITEM

Input: Noisy training set D_N, training epoch T, warmup iterations T_w, a selection criterion $\mathsf{Criterion}(\cdot)$, a classifier network with m experts, the mapping function $\mathcal{S}^{\beta}(\cdot)$ with a hyper-parameter β.

1: **Initialize** our network with one classifier layer and m expert layers $\mathbb{F} = \{f, g^1, ..., g^m\}$.
2: **for** t = 1, ..., T_w **do**
3: Randomly sample f^c from $\mathbb{F}$, and WarmUp f^c on D_N.
4: **end for**
5: **for** t = 1, ..., T **do**
6: Select the clean set D_{clean} from D_N based on $\mathsf{Criterion}(\mathbb{F})$.
7: Calculate weighted vector $\boldsymbol{v} = [w_1, ..., w_K]$ on D_{clean}.
8: Calculate reversed weighted vector $\tilde{\boldsymbol{v}} = [\tilde{w}_1, ..., \tilde{w}_K]$ with Equation (5).
9: **while** i = 1, ..., iterations **do**
10: Sample a mini batch $B_{\boldsymbol{v}} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^{b}$ from D_{clean} according to $\boldsymbol{v}$.
11: Sample a mini-batch $B_{\tilde{\boldsymbol{v}}} = \{(\boldsymbol{x}'_i, \tilde{y}'_i)\}_{i=1}^{b}$ from D_{clean} according to $\tilde{\boldsymbol{v}}$.
12: Sample f^c from $\mathbb{F}$.
13: Update the network parameters by minimizing the loss in Equation (7).
14: **end while**
15: **end for**

C.1. Dataset

CIFAR-10 and CIFAR-100 [17] dataset consists of 60,000 images of 10 and 100 categories. Each image is with a resolution of 32 × 32.

CIFAR10N and CIFAR100N [39] also consists of 60,000 images. Meanwhile, the noise labels in these two are human-made with varying noise conditions, including *worst* and *random 1&2&3* for CIFAR10N and *fine noise* for CIFAR100N.

Clothing1M [47] is a large-scale dataset that is collected from real-world online shopping websites. It contains 1 million images of 14 categories whose labels are generated based on tags extracted from the surrounding texts and keywords, causing huge label noise. The estimated percentage of corrupted labels is around 38.46%. A portion of clean data is also included in Clothing1M, which has been divided into the training set (903k images), validation set (14k images), and test set (10k images). We resize all images to

Table 5. We compare ITEM with various selection-based methods over four perspectives.

Methods	Network architecture	Selection-criterion specific	Training debias	Data debias
Co-teaching [11], CoDis [46]	Double-branch	Small-loss criterion	✗	✗
JoCoR [38], DivideMix [21], ProMix [37]	Double-branch	Small-loss criterion	✓	✗
SFT [40], Late Stopping [53]	Single-branch	Fluctuation-based selection	✓	✗
ITEM (ours)	Mixture-of-Experts	Not specific	✓	✓

Algorithm 2 ITEM with a semi-supervised framework

Input: The training set D_N, training epoch T, warmup iterations T_w, a selection criterion $\mathsf{Criterion}(\cdot)$, a classifier network with m experts, the mapping function $\mathcal{S}^{\beta}(\cdot)$ with a hyperparameter β.

Initialize our network with one classifier layer and m expert layers $\mathbb{F} = \{f, g^1, ..., g^m\}$.
for t = 1, ..., T_w **do**
  Randomly sample f^c from $\mathbb{F}$, and WarmUp f^c on D_N.
end for
for t = 1, ..., T **do**
  Split D_N into the clean set D_{clean} and the unreliable set $\tilde{D}_{\text{noise}}$ based on $\mathsf{Criterion}(\mathbb{F})$, and discard labels in $\tilde{D}_{\text{noise}}$.
  Calculate weighted vector $\boldsymbol{v} = [w_1, ..., w_K]$ on $\tilde{D}_{clean}$.
  Calculate reversed weighted vector $\tilde{\boldsymbol{v}} = [\tilde{w}_1, ..., \tilde{w}_K]$ with Eq. (5).
  while i = 1, ..., iterations **do**
    Randomly sample a batch $B_{\boldsymbol{v}} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^{b}$ from D_{clean} according to $\boldsymbol{v}$.
    Randomly sample a batch $B_{\tilde{\boldsymbol{v}}} = \{(\boldsymbol{x}'_i, \tilde{y}'_i)\}_{i=1}^{b}$ from D_{clean} according to $\tilde{\boldsymbol{v}}$.
    Draw a mini-batch $\hat{B} = \{(\boldsymbol{x}''_i)\}_{i=1}^{2b}$ from D_{noise} without labels.
    Generate pseudo-labels for each sample in $\hat{B}$, then have $\hat{B} = \{(\boldsymbol{x}''_i, \tilde{y}''_i)\}_{i=1}^{2b}$, where $\tilde{y}''_i = \mathbb{E}_{f\sim\mathbb{F}}[f(\boldsymbol{x}''_i)]$.
    Randomly sample f^c from $\mathbb{F}$.
    Update network parameters via minimizing $\mathcal{L}\big(f^c\big(\texttt{MixUp}(B_{\boldsymbol{v}} \text{ and } B_{\tilde{\boldsymbol{v}}}, \hat{B})\big)$.
  end while
end for

256×256 as in [21] and then random crop to 224×224.

Food-101N [20] is constructed based on the taxonomy of 101 categories in Food-101 [7]. It consists of 310k images collected from Google, Bing, Yelp, and TripAdvisor. The noise ratio for labels is around 20%. Following the testing protocol in [20], we learn the model on the training set of 55k images and evaluate it on the testing set of the original Food-101.

WebVision [23]. Following [26], we use a mini version of WebVision where only the top 50 classes are utilized. This mini WebVision dataset contains approximately 66 thousand images and the corresponding noise ratio is roughly 20%.

C.2. Other Baselines

On two human-annonated datasets CIFAR10N and CIFAR100N, we compare ITEM with Co-teaching+ [52], Peer Loss [27], CAL [59], DivideMix [21], ELR+ [26], CORES* [8], and DPC [60]. The results are collected from the literature [39]. On three real-world noisy datasets, we compared ITEM with SURE [25], Co-teaching [11], MentorNet [12], JoCoR [38], CDR [44], ELR [26], DivideMix [21], SFT [40], CoDis [46], SURE [25].

D. More Experimental Results

D.1. Results on Imbalanced Noisy Labels

Recently, some literature [13, 42, 56] studied a more challenging noise learning task, i.e., learning with noisy labels on imbalanced datasets, which is a more realistic scenario. Our method can also tackle this task thanks to the class-aware data sampling strategy in ITEM. Following the imbalanced noise setting and experimental setup of RCAL [56], we verify the performance of ‡ITEM on comprehensive conditions.

Dataset construction. Let $T_{ij}(\boldsymbol{x})$ denote a probability that the true label i of the instance $\boldsymbol{x}$ is corrupted to the noisy label. Therefore, given a noise ratio rho, the noise transition matrix satisfies that $T_{ij}(\boldsymbol{x}) = \rho$ if $i = j$, otherwise $T_{ij}(\boldsymbol{x}) = \frac{1}{C-1}\rho$, where C denotes the number of classes. We first construct an imbalanced dataset on CIFAR with different imbalanced ratios $\xi = \{10, 100\}$ and then corrupt the labels with varying noisy ratio $\rho = \{10\%, ..., 50\%\}$.

We compare our proposal ITEM with various methods in LNL and imbalanced LNL, including Co-teaching [11], RoLT [42], Sel-CL+ [22], and RCAL [56]. Comparison results are shown in Table 7. Weighted sampling, which mitigates data bias by adjusting selection probabilities and increasing the visibility of underrepresented classes during training, helps ITEM tackle long-tailed classification tasks. On both imbalanced noisy datasets, ITEM exhibits greater generalization under a high noise ratio. For exam-

Table 6. Experimental settings about training procedure of *ITEM*.

Datasets	CIFAR-10	CIFAR-100	CIFAR-10N	CIFAR-100N	Clothing-1M	Food-101N	WebVision
Class number	10	100	10	100	14	101	50
Training size	50,000	50,000	50,000	50,000	1,000,000	310,009	66,000
Testing size	10,000	10,000	10,000	10,000	10,000	25,250	2,500
Training procedure							
Network	ResNet-18	ResNet-34			ResNet-50		InceptionResNetV2
Batch size	64				100	32	
Epoch	200				10	100	30
Warmup epoch	10	30	10	30	1	5	3
Learning rate (LR)	0.02				0.001		0.02
Weight decay	1e-3				5e-4		
LR scheduler	divide 10 at [100,150]th epoch				divide 10 at 5th epoch	divide 10 at 50th epoch	divide 10 at 20th epoch
Optimizer	SGD						
Momentum	0.9						
Hyperparameters							
experts number m	$m = 4$				$m = 2$	$m = 3$	
the slope parameter β	$\beta = 3$						

Table 7. Test accuracy (%) of prevailing methods using the ResNet-32 on imbalanced noisy CIFAR-10 and CIFAR-100. The best and the second-best performance are highlighted with **bold** and underline, respectively.

	Imbalanced ratio ξ	10					100				
	Noise ratio ρ	10%	20%	30%	40%	50%	10%	20%	30%	40%	50%
CIFAR-10	CE	80.41	75.61	71.94	70.13	63.25	64.41	62.17	52.94	48.11	38.71
	Co-teaching [11]	80.30	78.54	68.71	57.10	46.77	55.58	50.29	38.01	30.75	22.85
	RoLT [42]	85.68	85.43	83.50	80.92	78.96	73.02	71.20	66.53	57.86	48.98
	Sel-CL+ [22]	86.47	85.11	84.41	80.35	77.27	72.31	71.02	65.70	61.37	56.21
	CurveNet [13]	87.12	85.02	84.39	82.99	78.57	76.55	74.19	71.90	67.20	64.83
	RCAL [56]	**88.09**	**86.46**	84.58	83.43	80.80	**78.60**	75.81	72.76	69.78	65.05
	Ours	87.32	86.31	**84.91**	**84.72**	**81.04**	78.39	**76.21**	**74.19**	**70.99**	**67.28**
CIFAR-100	CE	48.54	43.27	37.43	32.94	26.24	31.81	26.21	21.79	17.91	14.23
	Co-teaching [11]	45.61	41.33	36.14	32.08	25.33	30.55	25.67	22.01	16.20	13.45
	RoLT [42]	54.11	51.00	47.42	44.63	38.64	35.21	30.97	27.60	24.73	20.14
	Sel-CL+ [22]	55.68	53.52	50.92	47.57	44.86	37.45	36.79	35.09	31.96	28.59
	CurveNet [13]	55.96	54.60	51.28	48.10	44.61	40.91	39.71	35.90	32.09	29.71
	RCAL [56]	**57.50**	54.85	51.66	**48.91**	44.36	41.68	39.85	36.57	33.36	30.26
	Ours	56.10	**55.93**	**53.07**	48.29	**45.17**	**44.00**	**41.90**	**40.44**	**37.28**	**34.80**

ple, ITEM outperforms RCAL by 1.29% and 0.24% under 40% and 50% noise ratio, respectively. On CIFAR-100 with an imbalanced ratio $\xi = 100$, ITEM consistently achieves the best generalization performance over other comparison methods. Under an extreme noise rate $\rho = 50\%$, ITEM achieves more than 4.5% improvement of top-1 test accuracy. Therefore, we believe that ITEM can also fulfill strong robustness even under imbalanced real-world noisy conditions.

D.2. Visualization of Class-Level Selection

As shown in Figure 10, we further visualize the class-level selection performance of our proposal. We can see that the total selection performance increases as the training progresses. Further, in the tailed classes where the value F_k in other selection frameworks would decrease, ITEM gradually improves the F-score in all classes instead of only in head classes, demonstrating that ITEM effectively mitigates the data bias.

D.3. Training Time Analysis

In Table 8, we compare the training times of ITEM and (semi) ITEM* with three typical methods, using a single Nvidia 4090 GPU. We can observe that the training time of these methods based on double-brach networks (such as Co-teaching and DivideMix) is always twice as slow as training on a single network. However, in our proposed MoE structure, increasing the number of experts number will not significantly increase the computation cost. Besides, integrating ITEM with a semi-supervised learning framework can

Table 8. Comparison of total training time in hours on CIFAR10N with *Worst* noise labels with varying experts number.

Metric	Single Network	Double-branch Network		**Ours: MoE**			
	CE	Co-teaching [11]	DivideMix [21]	ITEM ($m = 3$)	ITEM ($m = 5$)	ITEM* ($m = 3$)	ITEM* ($m = 5$)
Test accuracy (%) ↑	77.69	83.26	92.56	91.15	91.40	93.14	93.32
Times (hours) ↓	1.6	4.2	4.4	1.6	1.7	1.8	1.8

Table 9. Test accuracy (%) on two fine-grained benchmarks with two noisy settings. The best and the second best performances are highlighted with **Bold** and underline, respectively.

Methods	Stanford Dogs		CUB-200-2011		*Avg.*
	Sym 40%	Asy 30%	Sym 40%	Asy 30%	
Cross-Entropy	51.42	58.08	64.01	56.02	57.38
Confidence Penalty [30]	68.69	64.50	52.40	54.33	59.98
Label Smooth [28]	70.22	<u>64.99</u>	54.39	56.80	61.60
Co-teaching [11]	49.15	50.50	46.57	50.60	49.21
JoCoR [38]	49.62	53.59	52.64	51.70	51.89
SNSCL [41]	**75.27**	64.49	<u>68.83</u>	<u>61.48</u>	<u>67.52</u>
Ours	<u>74.92</u>	**68.19**	**70.04**	**66.58**	**69.93**

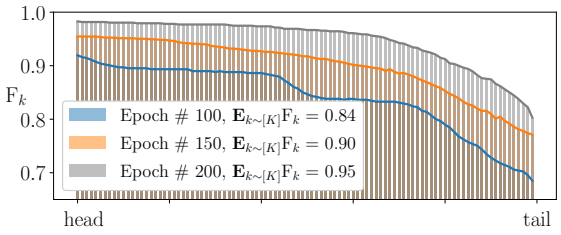


Figure 10. **Data debias** on selection. We plot the class-level selection performance of ITEM on CIFAR-100 with 60% symmetric label noise.

bring considerable performance improvement with fewer extra costs.

D.4. Results on Fine-Grained Noisy Labels

Recently, one literature [41] pointed out that noisy label generation is strongly associated with fine-grained datasets. Considering the property of wide existence and strong meaningfulness of noisy fine-grained classification, following settings in SNSCL [41], we conduct experiments on four noisy fine-grained datasets, including Stanford Dogs [15] and CUB-200-2011 [43]. Given the selection criterion GMM-based small loss selection, we fine-tuned a (pre-trained) ResNet-18 with three experts. We synthetically construct two noise types, symmetric and asymmetric noise labels.

The results are shown in Table 9. Our method achieves significant generalization performance even when tackling fine-grained noisy classification and consistently outperforms other methods by a large margin. For example, under Stanford Dogs with 30% asymmetric noise, ITEM improves the best method (label smooth) by more than 4%. We consider the MixUp operation to be the primary contribution to the advanced performance on fine-grained datasets for two reasons, 1) it encourages the model to learn from structured data instead of the unstructured noise [1], and 2) it encourages the model to learn a more generalized decision boundary, which largely benefits the fine-grained classification task.