NoiseBox: Toward More Efficient and Effective Learning With Noisy Labels

Chen Feng, Georgios Tzimiropoulos, and Ioannis Patras, Senior Member, IEEE

Abstract—Despite the large progress in supervised learning with neural networks, there are significant challenges in obtaining high-quality, large-scale and accurately labelled datasets. In such contexts, how to learn in the presence of noisy labels has received more and more attention. Addressing this relatively intricate problem to attain competitive results predominantly involves designing mechanisms that select samples that are expected to have reliable annotations. However, these methods typically involve multiple off-the-shelf techniques, resulting in intricate structures. Furthermore, they frequently make implicit or explicit assumptions about the noise modes/rationis within the dataset. Such assumptions can compromise model robustness and limit its performance under varying noise conditions. Unlike these methods, in this work, we propose an efficient and effective framework with minimal hyperparameters that achieves SOTA results in various benchmarks. Specifically, we design an efficient and concise training framework consisting of a subset expansion module responsible for exploring non-selected samples and a model training module to further reduce the impact of noise, called NoiseBox. Moreover, diverging from common sample selection methods based on the “small loss” mechanism, we introduce a novel sample selection method based on the neighbouring relationships and label consistency in the feature space. Without bells and whistles, such as model co-training, semi-supervised pre-training and semi-supervised learning, and with robustness concerning the settings of its few hyper-parameters, our method significantly surpasses previous methods on both CIFAR10/CIFAR100 with synthetic noise and real-world noisy datasets such as Red Mini-ImageNet, WebVision, Clothing1M and ANIMAL-10N.

Index Terms—Noisy labels, sample selection, K-nearest neighbours, feature consistency, class imbalance.

I. INTRODUCTION

Over the past two decades, deep neural networks have achieved remarkable success in various vision tasks, largely due to the application of supervised learning methods. The effectiveness of these methods can be attributed to the availability of high-precision, large-scale datasets, such as the ImageNet-1K dataset, which contains over one million images. However, the collection of such datasets for most vision tasks is typically time-consuming and labour-intensive.

Manuscript received 25 February 2024; revised 24 June 2024; accepted 6 July 2024. Date of publication 11 July 2024; date of current version 27 November 2024. This work was supported by the EU H2020 AHMedia under Project 951911. This article was recommended by Associate Editor R. He. (Corresponding author: Chen Feng.)

The authors are with the School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS London, U.K. (e-mail: chen.feng@qmul.ac.uk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2024.3426994.

Digital Object Identifier 10.1109/TCSVT.2024.3426994

To alleviate the reliance on labels, researchers have proposed various weakly supervised learning methods, including common semi-supervised and self-supervised learning, among others. In this paper, we focus on another class of typical weakly supervised learning problems, that is, learning with noisy labels. For example, in tasks such as medical image pathological diagnosis, the obtained labels may contain a large number of errors due to the experience gap of annotators. As another example, people sometimes utilize web-crawlers to automatically collect large-scale datasets, which also leads to a significant amount of label noise. Therefore, learning in the presence of noisy labels has emerged as an increasingly important research problem.

Formally speaking, label noise can be categorized into one of the following two types: closed-set noise where the true labels belong to one of the given classes (Set B in fig. 1) and open-set noise where the true labels do not belong to the set of labels of the classification problem (Set C in fig. 1).

To deal with these different label noise, roughly two main types of methods have been proposed, which we refer to as statistical-consistent methods and statistical-approximate methods. Statistical-consistent methods usually try to model noise mode explicitly and propose corresponding probabilistic adjustment techniques, e.g., robust loss functions [1], [2], [3] and label corrections based on noise transition matrix [4], [5]. However, accurate modelling of different noise modes is non-trivial. Due to the necessary simplifications of probabilistic models, such methods often perform poorly with heavy and complex noise.

More recently, statistical-approximate methods that do not model the noise modes explicitly become the dominant
paradigm, especially ones that are based on sample selection [6], [7]. Specifically, these methods often reduce the influence of noise samples by selecting and training with a clean subset [8], [9], [10], [11]. We note that while the majority of these methods do not explicitly model the noise modes, they still inherently rely on them. For example, DivideMix [12] has been tailored for datasets including asymmetric noise by introducing an extra designed confidence penalty, while Co-teaching [8] employs the threshold for sample selection based on the noise ratio. However, in reality, we often cannot know the noise mode and ratio. Moreover, in response to more complicated and severe noise, recent methods have shown a tendency to further integrate multiple off-the-shelf techniques, such as model co-training, model pre-training, and semi-supervised learning, among others. In DivideMix [12], an existing semi-supervised learning approach, MixMatch [13], has been incorporated by treating the selected subset as the labelled part and the remaining non-selected subset as the unlabelled, thereby thoroughly exploring the entire dataset. Co-teaching [8] employs model co-training to exchange the selected subsets of two models, aiming to mitigate the self-confirmation bias. However, despite bringing a steady performance increase, co-training also affects the model robustness and increases the computational complexity.

In this paper, we aim to alleviate the above limitations towards more efficient and effective learning with noisy labels. Firstly, we introduce a novel training framework called NoiseBox as a plug-and-play module for most existing sample selection methods, which comprises two key components: subset expansion and model training. In the subset expansion phase, we leverage the in-training model classifier to identify missing clean samples by the applied sample selection method and reassign labels to noisy samples when we are confident to do so. These samples are then added to the initially selected samples, followed by a class re-balancing step to create a well-labelled class-balanced subset. Then, in the model training phase, we employ a plain supervised learning scheme employing the cross-entropy loss on this refined subset. We also propose the incorporation of a feature consistency regularization for the entire dataset, facilitating the safe utilization of potential open-set noise while preserving the integrity of closed-set classification.

NoiseBox is designed to seamlessly integrate with almost all existing sample selection methods, for example, with sample selection methods motivated by “small loss” mechanism [12], [14]. Moreover, to further improve the sample selection performance, we also introduce a novel sample selection method based on the neighbouring relationships and label consistency in the samples’ feature space. Unlike previous sample selection methods that mostly rely on model predictions (e.g., “small loss” mechanism), we seek help from the pre-logit feature space and identify samples with labels consistent with those of their neighbouring samples as highly likely to be clean.

Preliminary findings of this work have previously been published in [15]. In this extended version, we further incorporate several notable novelties: i) We reformulate the “sample relabelling” and “model training” components in [15], and additionally introduce a “minority appending” mechanism to form a novel unified framework - NoiseBox, that most of the existing sample selection methods can be used with. ii) We test NoiseBox with several sample selection mechanisms, including a GMM-based sample selection method and a new minimum sample selection method and show that within the proposed NoiseBox framework both methods show strong performance. iii) We provide more detailed motivations of the proposed methods and conduct detailed corresponding ablation studies, including a detailed comparison between our proposed sample selection method and the GMM-based sample selection method, the impact of additional feature consistency loss on the model’s open-set testing performance, and the influence of incorporating historical information for subset expansion, etc. iv) To further validate the efficiency and effectiveness of our method, we conduct additional experiments on the more realistic Red Mini-ImageNet [16] dataset, which contains real-world label noise with different noise ratios. We also test our sample selection method on the clean CIFAR10, CIFAR100, and TinyImageNet datasets, demonstrating its great potential even with clean datasets.

In summary, our contributions are:

- We introduce a powerful and simple training framework called NoiseBox, which consists of two main components: subset expansion and model training, which achieves strong performance when used in conjunction with both our sample selection method and existing sample selection methods.
- We propose a simple and efficient sample selection method by examining the consistency between the labels of each sample and its neighbouring samples in the feature space. Our sample selection method not only reduces reliance on noise information in the dataset but also shows significantly improved performance in high noise ratios. Additionally, it also demonstrates strong performance on clean datasets.
- We conduct extensive experiments to validate the efficiency and effectiveness of our method. Our method achieves the best results with a simpler structure and fewer parameters than other methods in the literature.

The paper is organized as follows: in Section II we summarize related works in the field; in Section III, we explain our proposed sample selection method and the NoiseBox training framework; in Section IV, we present experimental results and extensive ablation studies on various noisy datasets; finally, in Section V we derive conclusions and draw directions for future work.

II. RELATED WORK

This section provides a brief review of the relevant works on learning with noisy labels.

A. Robust Loss Function and Noise Transition Matrix

Robust loss functions aim to reduce the impact of noisy samples during training, making the models more resilient to label noise. Ghosh et al. [17] discuss several common loss functions and demonstrate that certain ones, such as
Mean Absolute Error (MAE) loss, exhibit robustness to noise, while Cross-Entropy loss is sensitive to noise. Wang et al. [3] introduce the Symmetric Cross Entropy (SCE) loss, which is a linear combination of Cross Entropy (CE) loss and Reverse Cross Entropy (RCE) loss. Ma et al. [18] prove that any loss function can be transformed into a robust loss function by decomposing it into active and passive components.

The noise transition matrix models the probabilities of label transition from true labels to noisy labels. Methods such as loss correction [19] attempt to first estimate the noise transition matrix and then utilize forward and backward correction to mitigate the impact of label noise. Hendrycks et al. [20] utilize a validation dataset to estimate the noise transition matrix. More recent works primarily focus on improving the quality of the noise transition matrix. For example, Xia et al. [21] theoretically and empirically alleviate the reliance on anchor points for noise transition matrix estimation.

Specifically, compared to statistical-consistent methods that focus on proposing robust loss functions or noise transition matrices, our sample selection method avoids explicit assumptions about noise modes and demonstrates significantly better performance across various benchmark datasets.

B. Sample Selection for Learning With Noisy Labels

1) Prediction-Based Sample Selection: Most recent sample selection methods rely on the predictions of the model classifier, such as the per-sample losses [12], [14] or model predictions [11], [22]. Some works focus on further improving the sample selection quality by modelling the losses with a Markov chain [23] or checking the dynamic consistency of labels [24]. To explicitly identify open-set noise, several methods utilize the entropy of the model predictions [25], [26] as open-set noise samples tend to have larger entropy values.

However, the prediction-based selection is often unstable, especially in heavy noise scenarios, as early-stage model predictions may exhibit pronounced volatility and bias. In contrast, our method relies on the feature space of samples to perform more stable sample selection, especially when the noise ratio is high.

2) Feature-Based Sample Selection: More closely related to our proposed sample selection method, instead of selecting samples based on the model prediction, some works also try to utilize the feature representations for sample selection [27]. FINE [28] proposes to do sample selection by measuring the alignment between the latent distribution of dataset and each sample’s representation using the eigen decomposition. TopoFilter [29] and NGC [30] build a KNN graph based on sample features and identify clean samples through connected sub-graphs. Deep-KNN [31] selects clean samples with a KNN classifier in the prediction logit space, while MOIT [32] proposes an iterative KNN to alleviate the effect of noisy labels. Different from the above methods which usually adopt graphical models or designed feature-based sample selection indicators, we adopt the original KNN classifier to pursue the simplest model structure. Nevertheless, our method still achieves good results.

3) Fully Exploiting the Whole Dataset: To fully utilize the whole dataset during training, recent methods [12], [32] usually apply semi-supervised training methods, such as MixMatch [13], by considering the selected subset as labelled and the non-selected subset as unlabelled. More recent works [33], [34] also apply the improved version of MixMatch - FixMatch [35] for semi-supervised learning. However, most current semi-supervised learning methods cannot deal with open-set samples properly and tend to be complicated with multiple hyperparameters. How to properly perform semi-supervised learning in this setting is often referred to as open-set semi-supervised learning [36], [37]. In this paper, instead of adopting existing semi-supervised learning schemes, we adopt a simple subset expansion&class re-balancing scheme to construct a clean and well-labelled train set. We then train with a simple cross-entropy loss on the clean, well-labelled set, and optionally, with a feature consistency loss on the whole dataset to safely utilize the non-selected subset which may contain open-set noise.

C. Off-the-Shelf Techniques for Learning With Noisy Labels

Several off-the-shelf techniques have been widely applied in learning with noisy labels. For example, common regularization techniques such as dropout [38] and weight decay [39] can improve the model’s resistance to overfitting and thus alleviate the impact of noisy labels. More recently, techniques such as Mixup [40] and label smoothing [41] have gained widespread adoption due to their efficiency.

Some other techniques involving additional computational costs have also been investigated, such as model co-training [8], [9], [10], [12], [42], [43], self-supervised pre-training [44] and model re-training [32], [45]. For example, model co-training is often employed to reduce self-confirmation bias during the sample selection process, involving exchanging the sample selection results between two models or generating soft labels for each other.

III. METHODOLOGY

In this section, we provide a detailed explanation of the proposed method - see fig. 2 for a clearer explanation of the logic of our method. In Section III-A, we present a comprehensive formulation of the learning with noisy labels problem and essential notations. In Section III-B, we present our proposed and two other sample selection methods for learning with noisy labels. In Section III-C, we present the NoiseBox training framework, which can be combined with any existing sample selection method.

A. Problem Formulation

Let us denote with $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, $\mathbf{x}_i \in \mathbb{R}^d$, a noisy-labelled dataset with the corresponding one-hot vector labels $\mathcal{Y} = \{\mathbf{y}_i\}_{i=1}^{N}$, $\mathbf{y}_i \in \{0, 1\}^M$, where $M$ is the number of classes and $N$ is the number of samples. For the sake of convenience, we introduce the concept of the logit label for each sample $\mathbf{x}_i$, which corresponds to the one-hot label $y_i$ and is denoted as $l_i = \arg\max_{j} (y_{ij} = 1) \in \{1, \ldots, M\}$. Furthermore, we shall denote the true labels with $\mathcal{Y}' = \{y'_{ij}\}_{i=1}^{N}$. Clearly, for an open-set noisy label, it is the case that $y'_{ij} \neq y_i$, $y'_{ij} \not\in \{0, 1\}^M$, while for closed-set noisy labels we have $y'_{ij} \neq y_i$, $y'_{ij} \in \{0, 1\}^M$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Given a noisy-labelled dataset, our objective is to train a model, denoted as \( \mathcal{M} : \{f, g_p\} \), whose performance closely approaches that of a model trained on the clean dataset. Concretely, we conceptualise this model as comprising an encoder \( f \) responsible for feature extraction and a parameterized classifier \( g_p \) tailored to address the classification task at hand. Furthermore, leveraging the feature representations produced by the encoder \( f \), we introduce a non-parametric K-nearest neighbours (KNN) classifier, denoted as \( g_q \), which plays a pivotal role in the proposed sample selection methodology outlined in Section III-B.1. For the sake of conciseness, we introduce the notation \( f_i \equiv f(x_i) \) to denote the feature representation of sample \( x_i \), \( p_i \equiv g_p(f_i) \) to denote the prediction by the parametric classifier \( g_p \), and \( q_i \equiv g_q(f_i) \) to denote the prediction by the KNN classifier \( g_q \).

B. Sample Selection for Noisy Dataset

We first introduce our proposed sample selection methods in Section III-B.1, then two sample selection methods based on model predictions in Section III-B.2. Formally, we assume that all sample selection methods take the entire noisy dataset \( (X, Y) \) as input, aiming to obtain a clean subset \( (X_c, Y_c) \) as output. For the sake of brevity, we define aliases for the methods proposed in the method section. Specifically, the sample selection method we proposed is referred to as SS-KNN. The other two sample selection methods, in order, are named SS-GMM (Section III-B.2.a) and SS-Minimum (Section III-B.2.b).

1) SS-KNN - Sample Selection With Neighbouring Label Consistency: Our sample selection method is based on the neighbouring label consistency, i.e., we expect samples with highly consistent labels among their neighbours to be more likely to be clean. Specifically, we propose a measure \( c_i \), between the noisy label \( y_i \) of sample \( x_i \) and the prediction \( q_i \) by the KNN classifier \( g_q \) relying on the samples’ feature space, to quantify the neighbouring label consistency. Samples with high consistency \( c_i \) are then identified as clean and selected.

Let us denote the similarity between the representations \( f_i \) and \( f_j \) of any two samples \( x_i \) and \( x_j \) by \( s_{ij} \), \( i, j = 1, \ldots, N \). By default, we use the cosine similarity, that is,

\[
s_{ij} = \frac{f_i^T f_j}{\|f_i\|_2 \|f_j\|_2}.
\]

Let us denote the index set of the \( K \) nearest neighbours of sample \( x_i \) in \( X \) based on the calculated similarity. Then, for each sample \( x_i \), we can calculate the KNN-voted label distribution \( q'_i = \frac{1}{K} \sum_{n \in N_i} y_n \) in its neighbourhood, and a balanced version, \( q_i \), of it that takes into consideration the accumulated distribution \( \pi = \sum_{i=1}^{N} y_i \) of the labels in the dataset. More specifically,

\[
q_i = \pi^{-1} q'_i,
\]

where we denote with \( \pi^{-1} \) the vector whose entries are the inverses of the entries of the vector \( \pi \) — in this way we alleviate the negative impact of possible class imbalances in sample selection. The vector \( q_i \) can be considered as the (soft) prediction of the KNN classifier \( g_q \). We then, define a consistency measure \( c_i \) based on the sample’s logit label \( l_i \) and the prediction \( q_i \) of the KNN classifier as

\[
c_i = \frac{q_i(l_i)}{\max_j q_i(j)},
\]

that is the ratio of the value of the distribution \( q_i \) at the label \( l_i \) divided by the value of its highest peak \( \max_j q_i(j) \). We further validate the intuition of our method in Fig. 3 - we notice that both closed-set noise and open-set noise exhibit lower \( c_i \) (less consistent with its neighbours in the feature space) than clean ones.

Roughly speaking, a high consistency measure \( c_i \) of a sample \( x_i \) means that its neighbours agree with its annotated label \( l_i \) — this indicates that \( l_i \) is likely to be correct. By setting a threshold \( \theta_s \) to \( c_i \), a “clean” subset \( (X_c, Y_c) \) can be...
extracted as below:

\[ \mathcal{X}_c, \mathcal{Y}_c = \{(x_i, y_i) \mid c_i \geq \theta_i\}. \quad (4) \]

In our method, we set \( \theta_i = 1 \) by default, that is, we consider a sample \( x_i \) to be clean only when its neighbours’ voting \( q_i \) is consistent with its current label \( y_i \).

2) Other Applicable Sample Selection Method: To compare with our sample selection method and to evaluate the performance of the NoiseBox framework we will propose later, we also introduce two sample selection methods based on model predictions (losses). The first one is the most widely GMM-based sample selection - SS-GMM. The second one is a hyperparameter-free minimum sample selection method based on ranking sample losses - SS-Minimum.

Recalling previous notations, for each sample \( x_i \), we have \( p_i \) as the current model’s prediction. Assuming that the loss function used is denoted by \( L \), such as the commonly used cross-entropy loss, we obtain the per-sample losses as \( L = \{L(p_i, y_i)\}_{i=1}^N \).

a) SS-GMM - GMM-based sample selection with losses:

Following previous works [12], [14], regarding the per-sample losses \( L \), we use a Gaussian Mixture Model (GMM) to fit it and obtain the probabilities of each sample belonging to the component with a smaller mean. For consistency, we here denote the calculated probability also as \( c_i \).

\[ c_i = \text{GMM}_\text{SCORE}(L, L(p_i, y_i)). \quad (5) \]

We simplify the specific GMM modelling and probability calculation process above because it is straightforward. By default, the threshold \( \theta_i \) here is set as \( 0.5 \).

b) SS-Minimum - Minimum sample selection with losses:

The reason why the above GMM sample selection method has gained widespread application is that, compared to earlier sample selection methods such as co-teaching [8], [9] which relies on a known noise ratio as sample selection threshold in the dataset, the GMM modelling can automatically fit the per-sample losses and obtain a reasonable sample selection result. Here, as a weaker version, also to demonstrate the excellent performance of the NoiseBox training framework in the next section, we propose a parameter-free minimum sample selection method.

Specifically, we assume the noisy dataset is distinguishable — clean samples should be dominant for each class. Based on this, we believe that each class of samples in the dataset should have at least \( 1/M \) clean samples, with \( M \) as the number of semantic classes in the dataset as defined previously. Using per-sample losses, we sort them in ascending order and select the top \( 1/M \) samples with the smallest losses for each class to form the clean subset. Following previous notations, we have:

\[ c_i = \text{LOSS}_\text{RANKING}(L, L(p_i, y_i)). \quad (6) \]

Similarly, we also apply eq. (4) to generate the clean subset while the selection threshold here is fixed as \( 1/M \).

C. NoiseBox: An Efficient Training Framework for Noisy Datasets

In this section, we introduce an efficient training framework called NoiseBox, which can be flexibly combined with existing sample selection methods. Specifically, NoiseBox attempts to address several challenges after sample selection. Firstly, completely ignoring the unselected samples and relying solely on the selected clean subset \( (\mathcal{X}_c, \mathcal{Y}_c) \) for training is insufficient to achieve highly competitive results, especially when the noise ratio is high (table I). Secondly, even if the selected clean subset \( (\mathcal{X}_c, \mathcal{Y}_c) \) is expected to be cleaner compared to the original dataset \( (\mathcal{X}, \mathcal{Y}) \), it is still likely to contain noisy samples. Furthermore, we also consider enhancing the model’s robustness in dealing with issues such as class imbalance during the training process and the model’s adaptability to unknown noise mode/ratio.

NoiseBox consists of two main components: subset expansion (Section III-C.1) and model training (Section III-C.2).

1) Subset Expansion: To further utilize the non-selected samples (denoted as \( (\mathcal{X}_n, \mathcal{Y}_n) \equiv (\mathcal{X}, \mathcal{Y}) \setminus (\mathcal{X}_c, \mathcal{Y}_c) \)), we propose our subset expansion method, consisting of two steps: initial expansion and class re-balancing.

a) Initial expansion: To utilize the non-selected subset \( (\mathcal{X}_n, \mathcal{Y}_n) \), we consider those samples in it for which the parametric classifier prediction is confident, that is, all samples \( x_i \) for which the prediction \( p_i \) of the parametric classifier \( g_p \) exceeds a threshold \( \theta_i \). Here are two types of such samples. One type of sample will have model predictions that are consistent with the annotated labels. These samples are likely potential clean samples that might have been overlooked by the sample selection step. The second type of sample will have model predictions that do not match the annotated labels. We conjecture these samples to be initially wrongly labelled. Since the model is trained with a relatively clean dataset, their correct labels can be inferred from the model predictions. Denoting the expanded subset as \( (\mathcal{X}_e, \mathcal{Y}_e) \), we have:

\[ \mathcal{X}_e, \mathcal{Y}_e = \{ (x_i, y_i') \mid \max p_i \geq \theta_i \}. \quad (7) \]

Here, we abbreviate the corresponding one-hot label of \( \arg \max_l p_i(l) \) as \( y_i' \). Specifically, we empirically validate that by setting a relatively high threshold, we can ensure high precision of labels in this initially expanded subset (fig. 4(a)).

Additionally, please note, that this scheme implicitly avoids mislabeling open-set noisy samples because these samples often lack highly confident predictions and, instead, have relatively lower max \( p_i \) as evidenced in fig. 4(b).

b) Class re-balancing: By appending the expanded subset \( (\mathcal{X}_e, \mathcal{Y}_e) \) to the clean subset \( (\mathcal{X}_c, \mathcal{Y}_c) \), we have to some extent addressed the issue of insufficient data. However, an additional problem remains: the extended subset may suffer from class imbalance. This class imbalance can arise due to various reasons. Sometimes the initial noisy dataset itself might be class-imbalanced, as are datasets like WebVision [46]. More commonly, even if the original dataset

| Table I | Evaluation on CIFAR10 Datasets with Only Sample Selection |
|---------|----------------------------------------------------------|
|          | 0.5 sym | 0.9 sym |
| Ours     | 96.24   | 95.35   |
| GT subset| 94.84   | 90.83   |
| Clean dataset | 96.55 |
we follow previous works and apply Mixup interpolation [40].

Specifically, with two random samples $x_1$ and $x_2$, the mixed new samples $x_m, y_m$ will be generated as below:

$$\delta \sim \text{Beta}(\alpha, \alpha)$$
$$\delta' = \max(\delta, 1 - \delta)$$
$$x_m = \delta x_1 + (1 - \delta')x_2$$
$$y_m = \delta' y_1 + (1 - \delta')y_2$$

(9)

Then, we train the encoder $f$ and parametric classifier $g_p$ with common cross-entropy loss, that is,

$$L_{ce} = -y_m^T \log g_p(f(x_m))$$

(10)

b) Feature-consistency regularization with the entire dataset: Besides the induced train set $(\mathcal{X}_t, \mathcal{Y}_t)$, remaining samples, even open-set samples can also improve generalization. Motivated by commonly used prediction consistency regularization [13], we propose a feature consistency loss $L_{fc}$ to fully utilize the whole dataset $(\mathcal{X}, \mathcal{Y})$ without affecting the parametric classifier $g_p$. Specifically, motivated by recent works in self-supervised learning [47] and BYOL [48], we follow an asymmetric structure for regularizing the consistency of two projections. Briefly, consistency regularization based on a symmetric structure may lead to degenerate representations, meaning all samples produce the same features. However, an asymmetric structure has been proven to effectively avoid this. With a projector $h_{proj}$ and predictor $h_{pred}$, we define the feature consistency loss as the cosine distance between two different augmented views ($x_1$ and $x_2$) of the same sample $x$.

That is,

$$L_{fc} = -\frac{h_{1}^T h_2}{\|h_1\|_2 \|h_2\|_2}$$

(11)

where $h_1 \triangleq h_{proj}(h_{proj}(f(x_1)))$ and $h_2 \triangleq h_{proj}(f(x_2))$.

In summary, the overall training objective is to minimize a weighted sum of $L_{ce}$ and $L_{fc}$, that is

$$L = L_{ce} + \lambda L_{fc}.$$  

(12)

By default, we set $\lambda = 1$. 

IV. EXPERIMENTS

In this section, we conduct extensive experiments on two standard benchmarks with synthetic label noise, CIFAR10 and CIFAR100, and four real-world noisy datasets, Red Mini-ImageNet [16], ANIMAL-10N [22], WebVision [46], and Clothing1M [49]. In Section IV-A we give details of all datasets and default hyperparameters settings except for specific ablation experiments. In Section IV-B, we conduct extensive ablation experiments to show the great performance and robustness of our sample selection method - SS-KNN. In Section IV-C, we conduct ablation experiments about different modules of the NoiseBox training framework. In Section IV-D and Section IV-E, we compare our method with the state-of-the-art in synthetic noisy datasets and real-world noisy datasets.

When using the proposed SS-KNN sample selection method in combination with the NoiseBox, we refer to it as NoiseBox + SS-KNN, and so on for the other methods. For more details about the training process, please refer to Algorithm 1.
A. Implementation and Experiment Details

1) Datasets: We first introduce the details of all involved datasets, mainly divided into two types:

   a) Synthetic noisy dataset: CIFAR10 and CIFAR100 both consist of 50,000 images. Following the standard practice, for CIFAR10 and CIFAR100, we evaluate our method with two types of synthetic noise: symmetric noise by randomly replacing labels of all samples using a uniform distribution; and asymmetric noise by randomly exchanging labels of visually similar categories, such as Horse ↔ Deer and Dog ↔ Cat. For the closed-set noise-only dataset, we test with 20%, 50%, 80% and 90% symmetric noise and 40% asymmetric noise following DivideMix [12]. For datasets including also open-set noise, following settings in EDM [25], we test with 30%, 60% total noise ratio and 50%, 100% open-set noise ratio on the CIFAR10 dataset. The total noise ratio denotes the proportion of noisy samples in the dataset while the open-set noise ratio denotes the proportion of open-set noise in the noisy samples. The closed-set noise is generated as symmetric noise while the open-set noise is randomly sampled from CIFAR100. For brevity, we define abbreviated names for the corresponding noise settings, such as “0.5sym” for 50% symmetric noise, “0.4asym” for 40% asymmetric noise and “0.3all_0.5open” for 30% total noise with 50% open-set noise.

   b) Real-world noisy dataset: Red Mini-ImageNet dataset [16] is a real-world dataset containing a total of 100 categories. It is an extension of the Mini-Imagenet dataset, where noise is introduced at varying ratios. Specifically, noisy images and their respective labels are obtained by crawling the internet, and these noisy images replace the original images in the Mini-ImageNet dataset, with different noise ratios.

   WebVision [46] is a large-scale dataset of 1000 classes of images crawled from the Web. Following previous work [10], [12], [32], we compare baseline methods on the top 50 classes from the Google images Subset of WebVision. The noise ratio is estimated to be around 20%.

   ANIMAL-10N [22] is a smaller and recently proposed real-world dataset consisting of 10 classes of animals, that are manually labelled with an error rate that is estimated to be approximately 8%. ANIMAL-10N has similar size characteristics to the CIFAR datasets, with 50,000 train images and 10,000 test images.

   Clothing1M [49] is a large-scale dataset of 14 classes of clothing images crawled from online shopping websites, consisting of 1 million noisy images. The noise ratio is estimated to be around 38.5%.

2) Hyperparameters: We use a PreActResNet-18 [50] as the backbone for all CIFAR10/100 experiments following previous works. Unlike previous methods that use specific warm-up settings for CIFAR10/CIFAR100, we train the model after 10/30 warm-up epochs with $\theta_1 = 1.0$ in all experiments. We set $\theta_2 = 0.8$ for all CIFAR experiments except in the corresponding ablation part. In all experiments we train with an SGD optimizer for 300 epochs with a momentum of 0.9 and a weight decay of 5e-4. The initial learning rate is 0.02 and is controlled by a cosine annealing scheduler. The batchsize is fixed as 128.

   For Red Mini-ImageNet, we also use a PreActResNet-18 as the backbone following previous works [51], [52]. We train the network with an SGD optimizer for 300 epochs with a momentum of 0.9 and a weight decay of 5e-4. The initial learning rate is 0.02 and reduced by a factor of 10 after 200 and 250 epochs. The batchsize is fixed as 64. The images are resized from their original size of 84×84 pixels to 32×32 pixels. Moreover, we utilize noise ratios of 20%, 40%, 60%, and 80%. For all noise ratios, we train the model after 10 warmup epochs with $\theta_3 = 1$, while $\theta_4$ is fixed as 0.9.

   For WebVision, we use InceptionResNetv2 following [12]. We train the network with an SGD optimizer for 150 epochs with a momentum of 0.9 and a weight decay of 1e-4. The initial learning rate is 0.01 and reduced by a factor of 10 after 50 and 100 epochs. The batchsize is fixed as 32. We train the model after 5 warmup epochs with $\theta_3 = 1$, while $\theta_4$ is fixed as 0.95.

   For Clothing1M, we use ResNet50 following [12] with ImageNet pretrained weights. We train the network with an SGD optimizer for 150 epochs with a momentum of 0.9 and a weight decay of 1e-3. The initial learning rate is 0.002 and reduced by a factor of 10 after 50 and 100 epochs. The batchsize is fixed as 32. We train the model after 1 warmup epoch with $\theta_3 = 1$, while $\theta_4$ is fixed as 0.95.

   For ANIMAL-10N, we use VGG-19 [53] with batch-normalization following [22]. We train the network with an SGD optimizer for 150 epochs with a momentum of 0.9 and weight decay of 5e-4. The initial learning rate is 0.02 and reduced by a factor of 10 after 50 and 100 epochs. The batchsize is fixed as 128. We train the model after 10 warmup epochs with $\theta_3 = 1$, while $\theta_4$ is fixed as 0.95.

   Following recent works [32], [54], [55], in this work, we define three augmentation strategies: an original image

---

Algorithm 1 NoiseBox + Sample selection.

```
Input: dataset $(\mathcal{X}, \mathcal{Y})$, sample selection threshold $\theta_s$, subset expansion threshold $\theta_r$, weight of feature consistency loss $\lambda$, max epochs $T$

1 while $i < T$ do
2     Generate clean subset $(\mathcal{X}_c, \mathcal{Y}_c)$ with applicable sample selection method (SS-XXX);
3     Generate expanded subset $(\mathcal{X}_e, \mathcal{Y}_e)$ with eq. (7);
4     Generate balanced training subset $(\mathcal{X}_b, \mathcal{Y}_b)$ with eq. (8);
5     Model training with eq. (12).
6 end
```

---

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
which we denote as “none” augmentation for testing, random cropping+horizontal flipping which we denote as “weak” augmentation, and “strong” augmentation that further combines the augmentation policy from [56]. For \( L_{ce} \) we use “strong” augmentation with Mixup interpolations [40] while for \( L_{fc} \), we use “weak” augmentation for \( x_2 \) and “strong” augmentation for \( x_1 \). For Mixup interpolation, following DivideMix [12], we set \( \alpha = 4 \) for the beta mixture for the CIFAR10/CIFAR100 datasets, and as 0.5 for all other datasets.

B. Ablations on Sample Selection

In this section, we primarily investigate the performance of our proposed SS-KNN sample selection method, including the following aspects: comparing the SS-KNN sample selection method with the SS-GMM sample selection method (Section IV-B.1); exploring the robustness of the SS-KNN method concerning its hyperparameters (Section IV-B.2); analyzing the impact of different distance metrics used in the SS-KNN sample selection (Section IV-B.3). Furthermore, we also examine the feasibility of employing the SS-KNN sample selection method as a learning curriculum on clean datasets (Section IV-B.4).

1) Effectiveness of the Proposed Sample Selection Method:

In this section, we compare the quality of SS-KNN sample selection under different noise types/ratios and different datasets versus SS-GMM sample selection. To isolate the performance of the sample selection module from the influence of NoiseBox, we conduct the comparisons solely with sample selection mechanisms.

Specifically, we perform basic supervised training for model \( M :\{f, g_p\} \) over different epochs using only the cross-entropy loss with the entire dataset. This is commonly referred to as the “warm-up stage” and is an essential step in most contemporary methods. Subsequently, we evaluate and compare the sample selection performance between our proposed SS-KNN sample selection method and the SS-GMM sample selection method after different warm-up epochs. We set \( \theta_s = 1 \) for SS-KNN and \( \theta_s = 0.5 \) for SS-GMM.

In fig. 5, we report the sample selection performance of both methods in six different noise settings after \{0, 10, 20, 30, 50\} warm-up epochs, respectively. Firstly, we observe that with a randomly initialized model (Epoch 0), our method consistently demonstrates significantly better performance than the SS-GMM method. This is because even a randomly initialized neural network can be considered an effective feature extractor and suggests meaningful neighbouring relations between samples in the feature space, while a randomly initialized classifier is meaningless. This implies that our sample selection method eliminates the need for a mandatory warm-up phase and allows us to train from scratch, thus relieving the reliance of sample selection on the noise information in the dataset.

Secondly, when dealing with closed-set noise only, our method generally achieves higher sample selection precision (Clean / (Clean + Open-set noise + Closed-set noise)). This performance gap becomes more evident as the noise ratio increases (e.g., CIFAR10 0.5sym→CIFAR10 0.9sym) and as the dataset complexity increases (e.g., CIFAR10 0.5sym→CIFAR100 0.5sym).

Lastly, in the presence of open-set noise, the SS-GMM method achieves a slightly lower open-set noise ratio but also results in a slightly lower clean sample ratio. However, this trend diminishes as the number of warm-up epochs increases.

2) Robustness to Sample Selection Hyperparameters:

In this section, we conduct ablation studies to show the robustness of the values of the only hyperparameter \( \theta_s \) of the SS-KNN method. The choice of \( \theta_s \) controls the sample selection quality and proportion – the lower the \( \theta_s \), the more samples will be selected. We also investigate the effects of \( K \) — the size of the neighbourhood of KNN classifier \( g_q \) during the sample selection stage. Please note that we report the testing accuracy of the NoiseBox+SS-KNN method here.

In fig. 6a we report results with different \( \theta_s \) on four different noise ratios on the CIFAR10 dataset. We notice that removing sample selection (\( \theta_s = 0 \)) leads to severe degradation especially for a high noise ratio (90% symmetric noise), while a relatively high threshold (\( \theta_s = 0.8 \) or 1.0) brings consistently high performance. In fig. 6b, we report results with different \( K \) for the CIFAR10 dataset with 40% asymmetric noise. Except for extremely small \( K \), our method is stable and consistently better than the state-of-the-art.

3) Distance Metric for Sample Selection:

By default, we employ cosine similarity as the distance metric for K-nearest neighbours (KNN). For comparison, we also explore the utilization of the L2 distance metric. The corresponding results are presented in Table II. It is noteworthy that while there are minor differences, the overall performance is superior when employing cosine similarity.

4) Sample Selection as Learning Curriculum on Clean Datasets:

In this section, we evaluate the SS-KNN sample selection method as a curriculum learning approach on three clean datasets (CIFAR10, CIFAR100, and Tiny ImageNet). Please note that, unlike conventional curriculum learning methods\(^1\) that only change the order in which samples are added to the train set, our sample selection method may entirely remove certain samples from the training process. For the sake of experimental fairness, we do not utilize NoiseBox here, but solely incorporate the proposed SS-KNN sample selection mechanism on top of the common supervised learning scheme.

To our surprise, the results on the three different datasets produce three distinct outcomes (fig. 8a). On the CIFAR10 dataset, the inclusion of the SS-KNN sample selection method does not lead to significant differences. On the CIFAR100 dataset, SS-KNN sample selection has a noticeable negative

---

\(^1\)Previous research [57] has indicated that under normal training procedures, additional learning curriculum has limited impact on the performance of baseline datasets like CIFAR.
Fig. 5. Sample selection results after certain warm-up epochs. The figure depicts ratios of different sample types after the SS-KNN (Ours) and SS-GMM selection across various noise settings. We employ distinct colours to denote clean samples, closed-set noisy samples, open-set noisy samples, and non-selected samples, providing the corresponding proportion values. Specifically, the first three constitute the selected subset by the sample selection mechanism. Metrics like precision (Clean / (Clean + Closed-set noise + Open-set noise)) can be derived from these values.

Fig. 6. Hyperparameter robustness w.r.t $\theta_s$ and $K$ of SS-KNN sample selection method. (a) $\theta_s = [0, 0.8, 1.0]$; (b) $K = [1, 10, 50, 100, 150, 200, 250, 300]$. Dashed line dnomte the SotA results.

C. Ablations on NoiseBox Training Framework

In this section, we conduct extensive ablation experiments on various modules of the NoiseBox training framework, including: hyperparameters robustness in the initial expansion of subset expansion step (Section IV-C.1); analysis of the proposed feature consistency regularization (Section IV-C.2); the impact of other two computational-free components: class re-balancing in subset expansion and Mixup in supervised interpolated training (Section IV-C.3); and an analysis of the computation cost of each module (Section IV-C.4). Unless otherwise specified, all experimental results are obtained using the NoiseBox+SS-KNN method.

1) Robustness of Initial Expansion Step in Subset Expansion: The choice of $\theta_s$ controls the subset expansion quality and proportion. Roughly speaking, the lower the $\theta_s$, the more samples will be included. In fig. 10a we report the performance with different $\theta_s$ on the synthetic CIFAR10 noisy dataset. We find that our method achieves consistently superior performance than the state-of-the-art with different $\theta_s$ across different noise settings (0.4asym, 0.5sym and 0.9sym).

We also check the labelling quality of included samples in the expanded subset. In fig. 10b, we show that we obtain very high expansion precision with $\theta_s = 0.9$, e.g., only 19% samples have correct labels originally for 90% symmetric noise while >95% samples are correctly relabelled finally.

Motivated by recent works [12], [23], we also test the impact of incorporating historical predictions into subset expansion. In essence, instead of solely relying on the current model's
Fig. 7. Three kinds of non-selected samples in training on Tiny ImageNet dataset. “Correct” includes samples whose original labels are incorrect which should be discarded. “Ambiguous” comprises samples with inherent ambiguity, meaning they can be associated with multiple labels. “Wrong” represents a small portion of samples that should be selected but are mistakenly discarded. Upper row: original annotated label. Lower row: predicted label of the model with KNN sample selection.

Fig. 8. SS-KNN sample selection with \( \theta_s = [0, 0.8, 1.0] \) on clean datasets. (a) Testing accuracy difference over baseline; (b) Ratio of selected samples.

### Table III

| Noise    | Subset Expansion With Historical Predictions on CIFAR10 Dataset |
|----------|---------------------------------------------------------------|
| \( \theta_s \) | 0.5sym | 0.9sym | 0.5sym | 0.9sym |
| 1        | 96.21 | 96.33 | 96.52 | 96.41 | 93.87 | 95.35 | 93.24 | 88.75 |
| 10       | 95.90 | 96.38 | 96.46 | 96.42 | 91.25 | 94.87 | 89.63 | 88.68 |

predictions, we expand the subset using averaged prediction from the past over a certain number of epochs. In table III, we find that incorporating historical predictions leads to slight improvements in low noise scenario (0.5sym), while in high noise scenario (0.9sym), it results in a degradation in model performance. We suspect that this is due to the expansion criteria becoming more stringent, leading to fewer samples being included in the training process.

2) Analyzing Feature-Consistency Regularization: In this section, we conduct an ablation study on the proposed feature consistency regularization. It is important to reiterate here that the objective of feature consistency regularization is to effectively utilize non-selected samples, which include both erroneously labelled closed-set noise and open-set noise not belonging to any class. To provide a comprehensive evaluation, we first assess the learning model \( \mathcal{M} : \{f, g_p\} \) on an in-distribution test set and report its classification accuracy following a common supervised learning paradigm. Additionally, to evaluate the impact of feature consistency regularization on out-of-distribution samples, we introduce a simple open-set normalized entropy measure here. For example, our goal is to train a 10-way classifier on a noisy CIFAR10 train set which is unfortunately contaminated with open-set noise from the CIFAR100 dataset. In this context, we aim to achieve high classification performance on the in-distribution test set based on CIFAR10, while also reducing overconfidence on the out-of-distribution test set based on CIFAR100. Intuitively, we expect a better CIFAR10-based classifier to have lower prediction confidence for CIFAR100 dataset samples, as these samples do not belong to any CIFAR10 class. To this end, we define the “open\_norm” as the average entropy of model predictions on the out-of-distribution test set:

\[
\text{open\_norm} = 1 + \frac{1}{N_o} \sum_{i=1}^{N_o} \frac{\text{entropy}(p_i)}{\log(M)}. \tag{13}
\]

Here, \( N_o \) represents the number of samples in the out-of-distribution test set. To compare with our feature consistency regularization, we consider two other techniques, including the commonly used prediction consistency regularization technique in semi-supervised learning and a blank control group. Here, we follow the implementation of MixMatch [13] for the prediction consistency regularization.

In fig. 9, we report testing accuracy on the CIFAR10 test set and open\_norm on the CIFAR100 test set with the model trained on noisy CIFAR10 train set and with different sample selection methods. Firstly, we examine the impact of three different regularizations (with “None” denoting a blank control) on testing accuracy. Compared to the blank control, when there is only open-set noise (CIFAR10 0.3all_1.0open & CIFAR10 0.6all_1.0open), the additional prediction consistency regularization often leads to a decrease in accuracy, while our feature consistency regularization results in a stable accuracy improvement. This suggests that blindly using open-set noise samples to train classifiers may have negative impacts. When only closed-set noise is present (CIFAR10 0.5sym), both the prediction consistency regularization and our feature consistency regularization achieve accuracy improvements, further
Fig. 9. Feature consistency VS prediction consistency along with three sample selection methods. We report the results of three different options for consistency loss w.r.t. different sample selection methods and noise ratios. Upper: Testing accuracy (%) on CIFAR10 dataset. Lower: Open-set normalized entropy on CIFAR100 dataset.

TABLE IV

| Dataset          | Sample selection & Subset expansion | 50% sym CIFAR10 | 90% sym CIFAR10 |
|------------------|-------------------------------------|-----------------|-----------------|
|                  | Model training                      | weak | strong     | weak | strong     | weak | strong     | weak | strong     |
|                  | ACC (%)                             | 96.18 | 96.05     | 96.21 | 96.24     | 96.14 | 96.22     | 94.71 | 95.11     | 94.65 | 95.35     | 93.70 | 94.51     |

Fig. 10. Quality of subset expansion. (a) Testing accuracy with \( \theta_r = [0.7, 0.8, 0.9, 1] \) under different noise ratios/modes. The dashed line denotes the state-of-the-art (SOTA) results; (b) The ratio of correctly relabeled clean samples within the expanded subset.

validating our assumption. In the presence of both types of noise (CIFAR10 0.3all_0.5open & CIFAR10 0.6all_0.5open), the blank control group and prediction consistency exhibit distinct advantages and disadvantages, while our method consistently achieves the best results.

Next, we observe that both additional regularizations will increase the open_norm compared to the blank control group. We suspect that, even though our feature value normalization loss does not directly involve the classifier \( g_p \), the additional usage of open-set noise still has some influence, such as the batch-norm statistics. However, it is worth noting that compared to the significant open_norm increase brought by the prediction consistency regularization, our feature consistency loss has a smaller negative impact.

In conclusion, our feature consistency loss achieves consistent closed-set test accuracy improvement with a slight impact of the model’s robustness on an open-set test set and adapts well to various noise modes.

We also report the impact of distance metric for feature consistency loss in table V. We can find that replacing to the \( L_2 \) distance does not show a significant difference.

3) Effect of Computational-Free Techniques: In this section, we explore the impacts of remaining computational-free techniques in the NoiseBox framework, including Mixup interpolation, class re-balancing, and the usage of different data augmentation strategies throughout the entire process.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
In table IV, we report the impact of different data augmentation techniques. Specifically, since sample selection and subset expansion occur in the same stage, we apply the same data augmentation technique. It can be observed that stronger data augmentation during model training often leads to stable improvements, while overly strong or weak data augmentation during sample selection and subset expansion has a negative impact.

In table VI, we present the impact of Mixup and class re-balancing on three noisy datasets. It can be observed that both techniques, whether used individually or in combination, have led to stable performance improvements.

4) Computational Cost Analysis: In table VII, we report the running time of each step of our model. We also report the computational time of DivideMix [12] method with the same device and training budget. Please note that we primarily focus on the relative time consumed by each step, as the specific running times may vary depending on the hardware used. Here, “feature extraction” refers to the forward passing step before sample selection and subset expansion to extract sample features and predictions. It can be observed that the time for sample selection and subset expansion is almost negligible compared to the time for model training and feature extraction.

We further validate that the running time of DivideMix is significantly longer, particularly because our “Model training” stage does not incorporate semi-supervised modules. This distinction underscores the efficiency of our method in comparison to existing methods.

D. Evaluation With Synthetic Noisy Datasets

In this section, we compare our three variant methods (NoiseBox+SS-KNN, NoiseBox+SS-GMM and NoiseBox+SS-Minimum) to the most recent state-of-the-art methods.
methods. We show that it achieves consistent improvements in all datasets and at all noise types and ratios, even compared with our earlier version [15] in some experiments.

a) Evaluation with controlled closed-set noise: In this section, we compare our methods to the most competitive recent works. Table VIII shows results on CIFAR10 and CIFAR100 — we note again for our method this is without the use of model co-training or pre-training. It is clear that our method outperforms them, not only in the case of symmetric noise but also in the more realistic asymmetric synthetic noise settings.

Moreover, we also observe that even as a hyperparameter-free minimum sample selection method, the averaged performance of NoiseBox+SS-Minimum approaches 90% accuracy on the CIFAR10 dataset under various noise ratios. This indicates that CIFAR10, as a widely used benchmark dataset, has reached a level of saturation, and in the future, we may need more challenging benchmarks to keep up with evolving techniques.

b) Evaluation with combined open-set noise and closed-set noise: Table XI shows the performance of our method in a more complex combined noise scenario. Previous methods that are specially designed for open-set noise degrade rapidly when the open-set noise ratio is decreased from 1 to 0.5 [62], [63]. Also, the performance of the method without considering open-set noise like DivideMix [12] decreases when the open-set noise ratio is increased. EDM [25] modifies the method of DivideMix to deal with combined noise, however, reported results that are considerably lower than ours.

E. Evaluation With Real-World Noisy Datasets

Finally, in table IX, table X, table XII, table XIII and we show results on the Clothing1M, ANIMAL-10N, Red Mini-ImageNet and WebVision datasets, respectively. To summarize, our method achieves better or competitive performance compared to the current state-of-the-art in both large-scale web-crawled datasets and small-scale human-annotated noisy datasets.

F. Additional Discussion on Different Sample Selection Methods

SS-Minimum is a simple parameter-free sample selection method to validate the superior performance of our proposed NoiseBox and normally leads to sub-optimal performance compared to the other two. SS-KNN can be considered the default optimal choice in most cases, as demonstrated by its superior performance in the majority of experiments, especially when the dataset noise ratio is high. Additionally, it is noteworthy that when used with NoiseBox, both SS-KNN and SS-GMM outperformed the current state-of-the-art (SOTA) methods in most experiments.

V. CONCLUSION

To achieve efficient and effective learning with noisy labels, we aim to develop an efficient and effective method, in contrast to previous methods that integrate multiple mechanisms and regularizations. In this regard, we first propose a novel sample selection method based on neighbouring relations in the feature space. Unlike most methods that rely on model predictions, our approach reduces the dependence on dataset-specific information and demonstrates clear advantages.

Furthermore, we introduce a simple yet effective training framework called NoiseBox. By combining computational-free techniques such as subset expansion and class re-balancing,
NoiseBox does not utilize complicated mechanisms such as model co-training and model pre-training, and is shown with extensive experiments and ablation studies to be robust to the values of its few hyper-parameters and to consistently and significantly surpass the state-of-the-art in various datasets.

REFERENCES

[1] A. Ghosh, H. Kumar, and P. Sastry, "Robust loss functions under label noise for deep neural networks," in Proc. AAAI Conf. Artif. Intell., 2017, vol. 31, no. 1, pp. 1–20.

[2] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in Proc. Adv. Neural Inf. Process. Syst., vol. 31, 2018, pp. 1–12.

[3] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Ji, and J. Bai, "Symmetric cross entropy for robust learning with noisy labels," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 322–330.

[4] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," in Proc. Int. Conf. Learn. Represent., 2016, pp. 1–19.

[5] L. Wang, X. Xu, K. Guo, B. Cai, and F. Liu, "Reflective learning with label noise," IEEE Trans. Circuits Syst. Video Technol., vol. 33, no. 7, pp. 3343–3357, Jul. 2023.

[6] D. Okamura, R. Harakawa, and M. Iwahashi, "LCNME: Label correction using network prediction based on memorization effects for cross-modal retrieval with noisy labels," IEEE Trans. Circuits Syst. Video Technol., vol. 34, no. 1, pp. 590–602, Jan. 2024.

[7] Z. Zhang, F. Zhang, W. Hu, S. Feng, and J. Liu, "OHDB: An online category-aware framework for learning with noisy labels under long-tailed distribution," IEEE Trans. Circuits Syst. Video Technol., vol. 34, no. 5, pp. 3806–3818, May 2024.

[8] B. Han et al., "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in Proc. Adv. Neural Inf. Process. Syst., vol. 31, 2018, pp. 1–6.

[9] X. Yu, B. Han, L. Yao, G. Niu, I. Tsang, and M. Sugiyama, "How does disagreement help generalization against label corruption?" in Proc. Int. Conf. Mach. Learn., 2019, pp. 7164–7173.

[10] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, " MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in Proc. Int. Conf. Mach. Learn., 2018, pp. 2304–2313.

[11] I. Malach and S. Shalev-Shwartz, "Decoupling when to update," in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 1–19.

[12] J. Li, R. Socher, and S. C. Hoi, "Dividemix: Learning with noisy labels via self-reweighting from class centroids," IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 11, pp. 6275–6285, Nov. 2022.

[13] T. Zhou, S. Wang, and J. Bilmes, "Robust curriculum learning: From clean label detection to noisy label self-correction," in Proc. Int. Conf. Learn. Represent., 2020, pp. 1–7.

[14] R. Sachdeva, F. R. Cordeiro, V. Belagiannis, I. Reid, and G. Carneiro, "EvidentialMix: Learning with combined open-set and closed-set noisy labels," in Proc. IEEE Winter Conf. Comput. Vis. (WACV), Jan. 2021, pp. 3606–3614.

[15] P. Albert, D. Ortego, E. Arazo, N. E. O’Connor, and K. McGuinness, "Addressing out-of-distribution label noise in weby-labelled data," in Proc. IEEE/CVF Winter Conf. Comput. Vis. (WACV), Jan. 2022, pp. 2393–2402.

[16] F. Ma, Y. Xu, X. Yu, and Y. Yang, "Learning with noisy labels via self-reweighting from class centroids," IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 11, pp. 6275–6285, Nov. 2022.

[17] T. Kim et al., "FINE samples for learning with noisy labels," in Proc. Adv. Neural Inf. Process. Syst. (NIPS), vol. 34, 2021, pp. 24137–24149.

[18] P. Wu, S. Zheng, M. Goswami, D. Metaxas, and C. Chen, "A topological filter for learning with label noise," in Proc. Adv. Neural Inf. Process. Syst., vol. 33, 2020, pp. 21382–21393.

[19] Z.-F. Wu, T. Wei, J. Jiang, C. Mao, M. Tang, and Y.-F. Li, “NGC: A unified framework for learning with open-world noisy data,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 62–71.

[20] D. Bahri, H. Jiang, and M. Gupta, "Deep k-NN for noisy labels," in Proc. Int. Conf. Mach. Learn., vol. 30, 2019, pp. 540–550.

[21] D. Ortego et al., "Multi-objective interpolation training for robustness to label noise," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2021, pp. 6606–6615.

[22] N. Karim et al., "UNICON: Combating label noise through uniform selection and contrastive learning," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2022, pp. 9676–9686.

[23] Q. Wei, H. Sun, X. Lu, and Y. Yin, "Self-filtering: A noise-aware sample selection for label noise with confidence penalization," in Computer Vision—ECCV. Cham, Switzerland: Springer, 2022, pp. 516–532.

[24] K. Sohn et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2020, pp. 596–608.

[25] Q. Yu, D. Ikami, G. Irie, and K. Aizawa, "Multi-task curriculum framework for open-set semi-supervised learning," in Proc. Eur. Conf. Comput. Vis., 2020, pp. 438–454.

[26] K. Saito, D. Kim, and K. Saenko, "OpenMatch: Open-set consistency regularization for semi-supervised learning with outliers," 2021, arXiv:2105.14148.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.

[28] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in Proc. Adv. Neural Inf. Process. Syst., 1992, pp. 950–957.

[29] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "MixUp: Beyond empirical risk minimization," in Proc. Int. Conf. Learn. Represent., 2018, pp. 1–50.

[30] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. Hinton, "Regularizing neural networks by penalizing confident output distributions," in Proc. Int. Conf. Learn. Represent., 2017, pp. 1–8.

[31] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda, "Early-learning regularization prevents memorization of noisy labels," in Proc. Adv. Neural Inf. Process. Syst. (NIPS), vol. 33, 2020, pp. 20331–20342.

[32] Y. Chen, S. X. Hu, X. Shen, C. Ai, and J. A. K. Suykens, "Early-learning consistency and confidence," in Proc. Adv. Neural Inf. Process. Syst. (NIPS), vol. 34, 2021, pp. 24137–24149.

[33] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in Proc. IEEE Trans. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 1944–1952.

[34] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, "Using trusted data to train deep networks on labels corrupted by severe noise," in Proc. Adv. Neural Inf. Process. Syst., vol. 31, 2018, pp. 1–45.

[35] X. Xia et al., "Are anchor points really indispensable in label-noise learning?" in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 1–11.

[36] H. Song, M. Kim, and J.-G. Lee, “SELFIE: Refurbishing unclean samples for robust deep learning," in Proc. Int. Conf. Mach. Learn., 2019, pp. 5907–5915.

[37] X. Xia et al., "Sample selection with uncertainty of losses for learning with noisy labels," in Proc. Int. Conf. Learn. Represent., 2022, pp. 1–28.

[38] F. Ma, Y. Xu, X. Yu, and Y. Yang, "Learning with noisy labels via self-reweighting from class centroids," IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 11, pp. 6275–6285, Nov. 2022.
[46] W. Li, L. Wang, W. Li, E. Agustsson, and L. Van Gool, “WebVision database: Visual learning and understanding from web data,” 2017, arXiv:1708.02862.

[47] X. Chen and K. He, “Exploring simple Siamese representation learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 15745–15753.

[48] J.-B. Grill et al., “Bootstrap your own latent—A new approach to self-supervised learning,” in Proc. 34th Int. Conf. Neural Inf. Process. Syst., 2020, pp. 21271–21284.

[49] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, “Learning from massive noisy labeled data for image classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 2691–2699.

[50] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 630–645.

[51] A. Park, C. Nguyen, R. Felix, T.-T. Do, and G. Carneiro, “Instance-dependent noisy label learning via graphical modelling,” in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV), Jan. 2023, pp. 2287–2297.

[52] Y. Xu, L. Zhu, L. Jiang, and Y. Yang, “Faster meta update strategy for noise-robust deep learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 144–153.

[53] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learn. Represent., 2015, pp. 1–26.

[54] K. Nishi, Y. Ding, A. Rich, and T. Hollerer, “Augmentation strategies for learning with noisy labels,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Aug. 2021, pp. 8022–8031.

[55] C. Feng and I. Patras, “Adaptive soft contrastive learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 144–153.

[56] X. Wu, E. Dyer, and B. Neyshabur, “When do curricula work?” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 7017–7025.

[57] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V. Le, “AutoAugment: Learning augmentation strategies from data,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 113–123.

[58] X. Chen and K. He, “Exploring simple Siamese representation learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 1–36.

[59] K. Yi and J. Wu, “Probabilistic end-to-end noise correction for learning with noisy labels,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 7017–7025.

[60] J. Li, C. Xiong, and S. C. H. Hoi, “Learning from noisy data with robust representation learning,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 9465–9474.

[61] Y. Zhang, S. Zheng, P. Wu, M. Goswami, and C. Chen, “Learning with feature-dependent label noise: A progressive approach,” in Proc. Int. Conf. Learn. Represent., 2020, pp. 1–36.

[62] Y. Chen, X. Shen, S. X. Hu, and J. A. Suykens, “Boosting co-teaching with compression regularization for label noise,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Sep. 2021, pp. 2688–2692.

[63] K. Lee, S. Yun, K. Lee, H. Lee, B. Li, and J. Shin, “Robust inference via generative classifiers for handling noisy labels,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 3763–3772.

[64] Y. Wang et al., “Iterative learning with open-set noisy labels,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8688–8696.

[65] F. R. Cordeiro, R. Sachdeva, V. Belagiannis, I. Reid, and G. Carneiro, “LongReMix: Robust learning with high confidence samples in a noisy label environment,” Pattern Recognit., vol. 133, Jan. 2023, Art. no. 109013.