Expert Systems With Applications 255 (2024) 124672

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Multi-stage ensemble with refinement for noisy labeled data classification

Chihyeon Choi [a,b], Woojin Lee [c], Youngdoo Son [a,b,*]

[a] Department of Industrial and Systems Engineering, Dongguk University-Seoul, 30, Pildong-ro 1-gil, Jung-gu, Seoul, 04620, Republic of Korea
[b] Data Science Laboratory (DSLAB), Dongguk University-Seoul, 30, Pildong-ro 1-gil, Jung-gu, Seoul, 04620, Republic of Korea
[c] School of AI Convergence, Dongguk University-Seoul, 30, Pildong-ro 1-gil, Jung-gu, Seoul, 04620, Republic of Korea

ARTICLE INFO

Keywords:
Noisy label
Dataset refinement
Re-labeling
Ensemble
Robust loss function

ABSTRACT

Deep neural networks (DNNs) have made remarkable progress in image classification. However, since DNNs can memorize all the label information in the training dataset due to their excellent feature learning ability, the generalization performance deteriorates when they are trained on the noisy labeled dataset that can be easily found in real-world problems. In this paper, we propose a multi-stage ensemble method with label refinement to build an effective classification model under noisy labels. The proposed method iteratively refines the dataset by re-labeling the samples at the end of each stage, which enables the models trained at each stage to learn different features. By integrating these models, the proposed multi-stage ensemble method exerts powerful generalization performance. Also, we suggest a novel dataset refinement method, demonstrating the effectiveness of a robust function in distinguishing corrupted samples. Experimental results on the benchmark and real-world datasets show that the proposed method outperforms the existing methods on the noisy labeled dataset classification.

1. Introduction

Recent advances in deep neural networks (DNNs) have enabled remarkable successes in image classification (Harkat, Nascimento, Bernardino, & Ahmed, 2023; Lee & Son, 2022; Mehnatkesh, Jalali, Khosravi, & Nahavandi, 2023). However, they easily overfit on noisy labels due to their large capacity (Arpit et al., 2017; Zhang, Bengio, Hardt, Recht, & Vinyals, 2021). This phenomenon worsens the generalization performance of the model when it is trained on the dataset containing corrupted labels. Since datasets with incorrect labels can be easily found in the real-world (Fu, Zhang, & Wang, 2024; Liu, Li, Chai, & Zheng, 2024; Song, Kim, & Lee, 2019; Xiao, Xia, Yang, Huang, & Wang, 2015), learning algorithms robust to noisy labels are getting more attention.

In response, various approaches have recently been proposed in several directions, such as sample selection, label correction, and robust loss function-based methods. First, the sample selection methods are the approaches that train the model with samples estimated to have clean labels. Most of these methods regard instances with small losses as clean samples. These approaches take advantage of the DNN's property, which first learns to classify clean labeled instances and then overfits noisy labeled ones (Chen, Liao, Chen, & Zhang, 2019; Han et al., 2018; Jiang, Zhou, Leung, Li, & Fei-Fei, 2018; Shen & Sanghavi, 2019). Some studies extract clean samples by noise filtering algorithms that use the information of both loss and latent features (Kim et al., 2021; Wu, Zheng, Goswami, Metaxas, & Chen, 2020). Although these methods showed promising performance, they use only a subset of training samples. Therefore, when the dataset is highly contaminated, the performance can be severely weakened as the number of clean samples available for training the model becomes too small. Moreover, some methods require additional information on the ratio of the noisy label, i.e., noise ratio, which may not be known in the real-world (Han et al., 2018; Yu et al., 2019).

Next, the label correction methods are the approaches that try to improve the quality of raw labels. Some previous works corrected noisy labels using a noise model that represents the relationship between noisy and clean labels (Li et al., 2017; Vahdat, 2017; Xiao et al., 2015). Although they proposed to construct noise models in various ways, building a model that can reflect a complex nature of noise requires a complicated process. Also, it struggles to accurately estimate the true label when the noise ratio is high. To address these limitations, recent studies adopted a re-labeling approach by exploiting the representation learning ability of the DNNs (Mandal, Bharadwaj, & Biswas, 2020; Yuan, Chen, Zhang, Tai, & McMains, 2018). Despite such attempts, however, there were no significant advantages in classification performance. In addition, some of the existing methods have a limitation in

* Corresponding author at: Department of Industrial and Systems Engineering, Dongguk University-Seoul, 30, Pildong-ro 1-gil, Jung-gu, Seoul, 04620, Republic of Korea.

E-mail addresses: choich0509@dgu.ac.kr (C. Choi), wj926@dgu.ac.kr (W. Lee), youngdoo@dongguk.edu (Y. Son).

https://doi.org/10.1016/j.eswa.2024.124672
Received 14 February 2024; Received in revised form 27 June 2024; Accepted 29 June 2024
Available online 4 July 2024

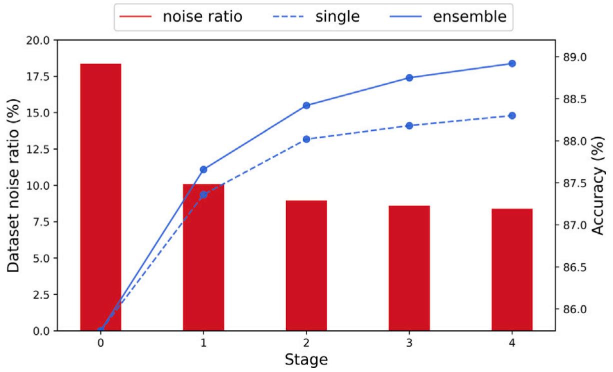


Fig. 1. The result of repeatedly training the model and refining the dataset on the CIFAR-10 with asymmetric-40% noise. The red bar represents the noise ratio of each stage dataset. The blue dashed line and solid line represent the accuracy of the single model and ensemble model at each stage, respectively.

that they work only when some validated clean data are provided (Lee, He, Zhang, & Yang, 2018).

Robust loss function-based approaches handle noisy labels by designing loss functions that are robust to noisy labels. Since the cross entropy (CE) loss is not designed for training on datasets with noisy labels, various researchers have tried to replace it with a robust loss function (Ghosh, Kumar, & Sastry, 2017; Wang et al., 2019; Zhang & Sabuncu, 2018). For example, Ma et al. (2020) proposed active passive loss (APL), which has a sufficient learning property that previous robust loss functions did not have, making it possible to build a model robust to various types and levels of noise. Unlike the sample selection and label correction approaches that intentionally reduce the proportion of noisy labels in the training dataset, the robust loss function-based methods train the model on the raw noisy labeled dataset. Although this direction achieved remarkable success only by simply using a robust loss function, there still remains room to be improved by reducing the noise ratio of the training dataset.

In this paper, we propose a multi-stage ensemble method with label refinement for the noisy labeled dataset. Our method is based on two key ideas: (1) the ensemble model shows powerful generalization ability when combining models that learn different features, and (2) each model can learn different features with iteratively refined datasets. Thus, in this study, we propose a novel ensemble method under the noisy labeled dataset by repeating the procedure of training the model and refining the corrupted dataset with a trained model. In each stage, we train the model with a robust loss function (Ma et al., 2020) to make each model robust on noisy labels. After the iterative process, we combine all the trained models to construct an effective ensemble model. In Fig. 1, we conducted a simple experiment to support our ideas, repeating the process of the training model and refining the dataset for four stages. As the stage repeats, an ensemble model was built by combining the trained models up to the stage where the training was completed. As depicted in Fig. 1, the noise ratio of the dataset gradually decreases due to the repetitive refinement. Trained on the different datasets, each model shows different performance, that is, they learn different features. Although there is a tendency that the model trained with a more refined dataset shows better performance, the ensemble model constructed by combining all the single models outperforms the best single model because it could reflect various features in classification.

In the proposed ensemble method, we inherit the property of the robust loss function so that each model is trained to be robust to noisy labels. However, unlike the existing robust loss function-based approaches, we propose to train individual models with different datasets, derived by our dataset refinement method. Through our dataset refinement process, we enable multiple models to learn diverse features, thereby achieving noteworthy generalization performance when they are integrated. Furthermore, utilizing the distribution of robust loss values, our refinement method effectively corrects the contaminated samples without requiring any additional information, such as validated clean samples and noise ratio of a dataset, which are difficult to obtain in practice. Therefore, the proposed method can be applied to real-world problems in a straightforward manner.

The main contributions of this paper can be summarized as follows:

- We propose an effective way to construct an ensemble model under the noisy labeled dataset by learning and integrating the heterogeneous models through dataset refinement, leading to powerful performance.
- We propose a novel dataset refinement method to construct a high-quality dataset exploiting the robust loss function, which enables several models to capture diverse features based on distinct refined datasets.
- Our re-labeling approach does not require any additional information on the contaminated dataset, such as validated clean samples and noise ratio of the dataset.
- The proposed method showed state-of-the-art performances on noise-injected benchmark datasets and real-world noisy labeled datasets.

The remainder of this paper is organized as follows: Section 2 describes literature reviews regarding robust classification methods on noisy labels. Section 3 details the proposed multi-stage ensemble method. Section 4 presents experimental results and analysis, followed by the conclusion in Section 5.

2. Related work

In this section, we briefly review previous works that attempted to construct a robust classification model on noisy labels. Most previous approaches can be summarized into the following three categories.

Sample selection methods. These methods improved the robustness by training the model only with the samples that are estimated to have clean labels. Most of them utilize the sample selection criterion called the small loss trick, which selects instances with small losses as clean samples. This trick exploits the memorization effect, a property that DNNs learn the clean pattern first and then overfit to the noisy pattern (Arpit et al., 2017; Zhang et al., 2021). MentorNet (Jiang et al., 2018) selected clean labeled samples with pre-trained networks and Co-teaching (Han et al., 2018) trained two networks simultaneously to minimize the sample selection bias that can occur when estimating clean samples using a single model. JoCoR (Wei, Feng, Chen, & An, 2020) proposed a joint loss function based on the Co-teaching network, which encourages the predictions from two networks closer. There have also been proposed several methods by modifying the Co-teaching concept (Chen, Shen, Hu, & Suykens, 2021; Mandal et al., 2020; Yu et al., 2019). Recently, various sample selection methods with a substantial theoretical background have been proposed (Kim et al., 2021; Mirzasoleiman, Cao, & Leskovec, 2020; Wu et al., 2020). CRUST (Mirzasoleiman et al., 2020) selected clean samples by identifying subsets of input instances that closely cluster in the gradient space. TopoFilter (Wu et al., 2020) filtered noisy samples using the k-nearest neighborhood algorithm and Euclidean distance in the latent space. The effective filtering algorithm FINE (Kim et al., 2021) was also proposed, which filters noisy samples by assessing the alignment between class-representative features, obtained through the eigen decomposition, and the features of individual samples.

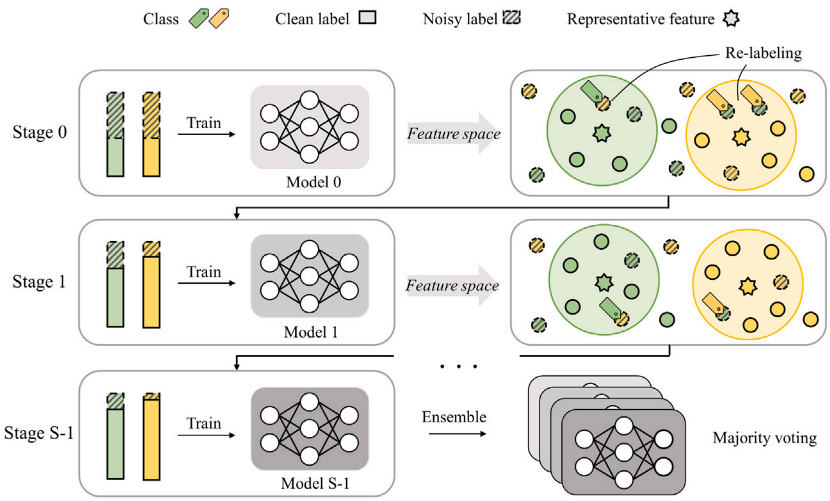


Fig. 2. Overview of the proposed multi-stage ensemble method. Each color represents a different class; sold patterns and diagonal lines indicate clean and noisy labels, respectively. Star shapes denote representative feature vectors. In the initial stage, a model is trained using a given noisy labeled dataset. Subsequently, training samples close enough to a representative feature vector from another class are re-labeled. As iteratively repeating the process of training the model and re-labeling the dataset, the models are trained to learn distinct features at each stage. Finally, an ensemble model is constructed by integrating all the trained models.

Label correction methods. Label correction methods aim to improve the quality of the dataset by revising the incorrect labels. Most studies in this category infer the true labels of noisy samples with a complicated noise model and replace the labels with them. There have been various attempts to construct the noise models (Li et al., 2017; Vahdat, 2017; Xiao et al., 2015); recent approaches mainly exploited DNNs (Lee et al., 2018; Veit et al., 2017). Tanaka, Ikami, Yamasaki, and Aizawa (2018) proposed a joint framework to optimize the parameters of the classifier and labels. Yuan et al. (2018) trained multiple networks with different subsets of the dataset and corrected the labels for the samples with mismatched predictions. Also, there is an approach performed re-labeling based on the Co-teaching. Mandal et al. (2020) extracted small loss and large loss samples from each mini-batch based on the Co-teaching network and constructed per class mean feature with small loss samples. Then, re-labeling was performed on the large loss samples based on the distance between the feature of each sample and the per class mean features. This re-labeling process seems similar to ours. However, unlike our approach, their work is specialized to the Co-teaching network and requires a noise ratio of the dataset. Moreover, our approach is not affected by the randomness of the mini-batch of the dataset because the loss distribution of the entire sample is considered in the proposed re-labeling process.

Robust loss function. This direction designs robust loss functions to address noisy labels. Ghosh et al. (2017) proved that the cross entropy (CE) loss, generally used for classification models, is not robust to noisy labels, whereas the mean absolute error (MAE) is robust. However, Zhang and Sabuncu (2018) showed that the model trained with MAE converges slowly due to the gradient saturation. To address this limitation, symmetric cross entropy (SCE) loss, which combines CE and reverse cross entropy (RCE) together, and generalized cross entropy (GCE) loss were suggested (Wang et al., 2019; Zhang & Sabuncu, 2018). Ma et al. (2020) found these loss functions are only partially robust and proved that any losses could be robust through simple normalization. They proposed active passive loss (APL), combining two losses with different optimization behaviors based on the theoretical background. Moreover, Ye et al. (2024) introduced the active negative loss (ANL), which combines the normalized active loss in the APL (Ma et al., 2020) with the proposed negative loss function. Additionally, Wei et al. (2023) proposed LogitClip, which clamps to the norm of the logit vector. They demonstrated that the models become more tolerant to noisy labels when LogitClip is integrated with existing losses.

3. Proposed method

This section introduces our multi-stage ensemble with refinement for noisy labeled data classification. Section 3.1 introduces problem setting, which describes the problem definition and the robust loss function employed in the proposed method. Section 3.2 elaborates on the dataset refinement method based on the initially trained model. Then, in Section 3.3, we present the iterative process involving dataset refinement and model training to develop an effective ensemble model under the noisy labeled dataset. Finally, in Section 3.4, we discuss the computational complexity of the entire pipeline of our proposed method. Fig. 2 represents an overall process of the proposed method.

3.1. Problem setting

Problem definition. We consider a noisy labeled dataset of N samples, denoted as $\mathcal{D}_{noisy} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, where $\boldsymbol{x}_i$ is the i-th sample and $y_i \in \{1, \dots, C\}$ is the assigned label of $\boldsymbol{x}_i$, which may not correctly match with the true label y_i^*. To indicate the noise level, we denote the class-wise noise ratio from class j to class k as η_{jk}, and the overall noise ratio of $\mathcal{D}_{noisy}$ as $\eta \in [0, 1]$. In addition, we represent the number of samples in the class c as $|\mathcal{D}_c|$, where $\sum_{c=1}^{C} |\mathcal{D}_c| = N$.

The objective of this study is to construct an ensemble model comprised of S different models, Stage 0 to $S-1$ models, where S is the total number of stages. At Stage s, we train the classification model f_s on the iteratively refined dataset $\mathcal{D}_{noisy}^{s}$, and extract a feature representation of a sample using a feature extractor F_s. Note that, at Stage 0, the classification model f_0 is trained on the initially given noisy labeled dataset $\mathcal{D}_{noisy}^{0}$, which has not been refined. Additionally, F_0 denotes the feature extractor in f_0, utilized to extract feature representations for each sample at this stage.

Active passive loss. Given a noisy labeled dataset, we train the classification model with the NCE+RCE loss suggested in Ma et al. (2020), which combines two robust losses, normalized cross entropy (NCE) and reverse cross entropy (RCE). Let $\boldsymbol{p}$ be the predictive distribution of a classification model and $\boldsymbol{q}$ be the one-hot label distribution over different classes. NCE is calculated by dividing the CE by the sum of CE of all possible label distributions and the predictive distribution of the model, as shown in Eq. (1).

$$\ell^{NCE} = \frac{\sum_{c=1}^{C} q(c \mid \boldsymbol{x}) \log p(c \mid \boldsymbol{x})}{\sum_{j=1}^{C} \sum_{c=1}^{C} q(y = j \mid \boldsymbol{x}) \log p(c \mid \boldsymbol{x})}. \tag{1}$$

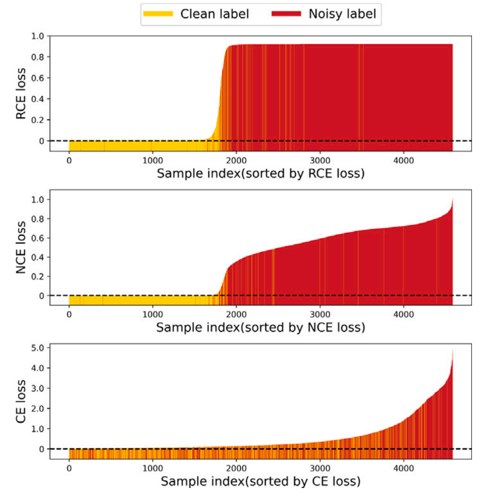


Fig. 3. NCE, RCE, and CE loss distributions sorted in ascending order for the first class of the CIFAR-10. After injecting sym-60% noisy labels into the CIFAR-10, we trained two models with NCE+RCE loss and CE loss. The first two rows show the values of NCE and RCE losses using the model trained with NCE+RCE loss. The last row shows the values of CE loss using the model trained with CE loss. In the figure, the yellow and red lines represent the clean and noisy labels, respectively.

Table 1
The average clean ratio (%) in small loss samples for RCE, NCE, and CE losses across all classes in the CIFAR-10 with sym-60% noise. The first two rows represent the results of RCE and NCE loss cases using a model trained with NCE+RCE loss. The last row represents the result of CE using a model trained with CE loss.

Loss type	Small loss ratio			
	10%	20%	30%	40%
RCE	99.31	99.26	98.57	93.22
NCE	99.26	99.22	98.51	93.08
CE	56.41	56.97	56.87	56.55

Algorithm 1 Dataset refinement with a trained model

```
Input: trained model $f_s$ using $\ell^{APL}$, noisy labeled dataset $\mathcal{D}^s_{noisy}$
Parameter: small loss ratio $\rho$, confidence threshold $\gamma$
Output: refined dataset $\mathcal{D}^{s+1}_{noisy}$
for $c = 1, \cdots, C$ do
    $\mathcal{L}^{RCE}_c \leftarrow \ell^{RCE}$ of the $c$-th class samples using the model $f_s$
    Sort the values in $\mathcal{L}^{RCE}_c$ in ascending order
    $\mathcal{D}^{Small}_c \leftarrow \lfloor |\mathcal{D}_c| \times \rho \rfloor$ smallest $\ell^{RCE}$ values in the $\mathcal{L}^{RCE}_c$
end for
$\{\mu\}^C_{c=1} \leftarrow$ Compute representative features by Eq. (4)
$\mathcal{D}^{s+1}_{noisy} \leftarrow$ Refine $\mathcal{D}^s_{noisy}$ by re-labeling criterion Eq. (5)
```

RCE is calculated by flipping the label distribution and the predictive distribution in CE, as shown in Eq. (2)

$$\ell^{RCE} = -\sum_{c=1}^{C} p(c \mid \boldsymbol{x}) \log q(c \mid \boldsymbol{x}). \tag{2}$$

The NCE+RCE loss ℓ^{APL} is calculated by a weighted sum of ℓ^{NCE} and ℓ^{RCE}, as depicted in Eq. (3)

$$\ell^{APL} = w_1 \times \ell^{NCE} + w_2 \times \ell^{RCE}, \tag{3}$$

where w_1 and w_2 are weights for balancing ℓ^{NCE} and ℓ^{RCE}. As a weighted sum of two robust losses, ℓ^{APL} is also proven to be robust to the noisy label (Ma et al., 2020).

3.2. Dataset refinement with an initially trained model

The proposed multi-stage ensemble method includes several dataset refinement processes. For the dataset refinement, we re-label the noisy labeled dataset after constructing a representative feature for each class with a trained model. Here, we provide a detailed description of the re-labeling method based on the initially trained model and present a property of the RCE we empirically found in the context of the re-labeling.

In the beginning, we train a Stage 0 model f_0 on the provided noisy labeled dataset $\mathcal{D}^0_{noisy}$ using the robust loss ℓ^{APL}. Subsequently, we estimate probably clean samples in the noisy dataset and form a representative feature for each class using the estimated samples. To effectively encode uncorrupted semantic information in each representative feature, the estimated clean samples should comprise a high proportion of clean instances. To estimate probably clean samples, we utilize the magnitude of the RCE values derived from the trained model f_0.

Fig. 3 shows the sorted values of ℓ^{RCE}, ℓ^{NCE}, and CE values of the first class in the CIFAR-10 training dataset, where 60% of labels are corrupted. The first two rows show the ℓ^{RCE} and ℓ^{NCE} values from the model trained with ℓ^{APL}. The last row shows the CE values from the model trained with CE. When the model is trained with CE, a large number of noisy samples have low loss values because the model is overfitted to the noisy labels. That is, the model fails to distinguish whether a training sample is clean or not by the magnitudes of loss values. However, when the model is trained with the robust loss ℓ^{APL}, both ℓ^{RCE} and ℓ^{NCE} can easily distinguish the samples with noisy labels. Similar observations are found in the other classes. In addition, Table 1 shows the average clean sample ratio in small loss samples across all classes. In the ideal case, where all the clean samples exhibit smaller loss values than the noisy ones, 40% of the small loss samples are expected to consist of clean samples since there are 60% noisy samples in the dataset. Although it is not ideal, small ℓ^{NCE} and ℓ^{RCE} values could be leveraged to extract samples with high purity. Among them, ℓ^{RCE} is the most effective in estimating probably clean samples, consistently assigning small loss values to clean samples compared to other losses, irrespective of the number of samples considered as small loss ones.

Motivated by the above results, we regard samples with small ℓ^{RCE} as probably clean samples. To identify such samples, using f_0, we construct a RCE loss set for each class, $\mathcal{L}^{RCE}_c = \{\ell^{RCE}_{c,n}\}^{|\mathcal{D}_c|}_{n=1}$ $(c = 1, \dots, C)$ by sorting the loss values in ascending order, i.e., $\ell^{RCE}_{c,n}$ is the n-th small RCE loss in the class c. Then, by extracting samples with $\lfloor |\mathcal{D}_c| \times \rho \rfloor$ smallest ℓ^{RCE} values for the class c, we obtain a small loss dataset for each class $\{\mathcal{D}^{Small}_c\}^C_{c=1}$, which determines probably clean samples. Here, ρ $(0 \leq \rho \leq 1)$ is a small loss ratio, which is a hyperparameter that controls the size of the small loss dataset.

Next, we construct a representative feature for each class based on the small loss dataset and re-label the noisy labeled dataset $\mathcal{D}^0_{noisy}$. The representative features $\{\mu\}^C_{c=1}$ are computed by the Eq. (4) using a feature extractor F_0 and the small loss datasets $\{\mathcal{D}^{Small}_c\}^C_{c=1}$.

$$\mu_c = \frac{1}{|\mathcal{D}^{Small}_c|} \sum_{\boldsymbol{x}_i \in \mathcal{D}^{Small}_c} F_0(\boldsymbol{x}_i), \tag{4}$$

As the average features of the probably clean samples within each class, the representative features can effectively reflect the class-specific features with minimal interference from noisy samples.

Dataset refinement is performed by re-labeling a training sample when its label does not match the class of the closest representative feature. To figure out samples for re-labeling, we compute a distance vector $d_i \in \mathbb{R}^C$, where each component indicates the negative Euclidean distance between an i-th sample and the representative feature of each class in the feature space. A larger c-th component of d_i means a closer distance between the $F_0(\boldsymbol{x}_i)$ and the μ_c. Then, by passing d_i to a softmax function, we derive $\hat{d}_i \in \mathbb{R}^C$ $(\sum_{j=1}^{C} \hat{d}_i[j] = 1)$. Finally, we re-label y_i

Algorithm 2 Multi-stage ensemble with refinement

Input: noisy labeled dataset $\mathcal{D}^0_{noisy}$
Parameter: total number of stages S
Output: ensemble model $H(\boldsymbol{x})$

```
for $s = 0, ..., S-1$ do
    train $f_s$ on the noisy labeled dataset $\mathcal{D}^s_{noisy}$
    if $s$ not $S-1$ then
        $\mathcal{D}^{s+1}_{noisy}$ ← dataset refinement by Algorithm 1
    end if
end for
$H(\boldsymbol{x})$ ← ensemble $\{f_0, \cdots, f_{S-1}\}$
```

according to Eq. (5) and construct a refined dataset $\mathcal{D}^1_{noisy}$, which will be used for the training model in the next stage (Stage 1).

$$y_i = \begin{cases} \underset{j}{argmax}\ \hat{d}_i[j], & \text{if } \underset{j}{max}\ \hat{d}_i[j] > \gamma \\ y_i, & \text{otherwise}, \end{cases} \tag{5}$$

where γ $(0 \le \gamma \le 1)$ is a confidence threshold that can be tuned as a hyperparameter. As the γ increases, only samples closer to one of the representative feature vectors can be re-labeled. In this way, we can adjust the re-labeling strategy to be either conservative or aggressive.

The proposed dataset refinement method can be expanded to apply in any stage if there is a trained model using a robust loss ℓ^{APL}. The Algorithm 1 represents the dataset refinement process.

3.3. Iterative refinement and training for ensemble

The key of a successful ensemble is that each model therein needs to be independent and learn different features (Allen-Zhu & Li, 2023). To construct an effective ensemble model, we aim to acquire multiple models that capture distinctive features by repeating the process of refining the dataset and training the subsequent model for a pre-defined total number of stages.

Specifically, at Stage s, we train a model f_s on a dataset $\mathcal{D}^s_{noisy}$ and construct a refined dataset $\mathcal{D}^{s+1}_{noisy}$ by the re-labeling method demonstrated in Section 3.2. As the stage proceeds, we train each model in a progressive manner. In other words, except for stage 0, we use the trained model from the previous stage as a pre-trained model to train the subsequent stage model. At Stage 0, the model f_0 is randomly initialized and trained on the provided dataset $\mathcal{D}^0_{noisy}$. Meanwhile, for Stage s $(s \ge 1)$, the model continues training from the weights of the trained model at Stage $s-1$. In this way, each model can learn different features compared to the previous model even without losing previously obtained knowledge. In the perspective of dataset refinement, some noisy samples may not get a chance to be corrected in the early stages. However, they can be corrected in the later stages by the model that learns different features from the refined dataset.

Here, we also present that the expected loss of the trained model with the noisy dataset decreases compared to the previous stage as the re-labeling is repeated, providing that the noise ratio of a certain stage is not greater than the previous one and some other assumptions. In other words, the generalization performance of the classifier is expected to be improved as the stage proceeds as in Fig. 1. The proof is provided in Appendix.

Proposition 1. *Let $\ell^{norm}(f(x), y)$ be a normalized loss function, that is $\sum_{y'} \ell^{norm}(f(x), y') = 1$, and $R(f) = \mathbb{E}_{x,y^*}\ell^{norm}$ and $R^{\eta^s}(f) = \mathbb{E}_{x,y^*}\ell^{norm}$ be the risk of the classifier f under the clean labels and the noisy labels at Stage s, respectively. η^s_{jk} denotes the class-wise noise ratio from class j to k at the Stage s.*

Let f^* and $f^*_{\eta^s}$ be global minimizers of $R(f)$ and $R^{\eta^s}(f)$, respectively. Given

1. $R(f^*) = 0$,
2. $0 \le \ell^{norm}(f^*(x), j) \le \frac{1}{C-1}, \forall j$,
3. $\eta^s_{jk} < 1 - \sum_{l \ne j} \eta^s_{jl}, \forall j, k$, *and*
4. $\eta^{s+1}_{jk} \le \eta^s_{jk}, \forall s \ge 0$, *then*

$$R^{\eta^{s+1}}(f^*_{\eta^{s+1}}) \le R^{\eta^s}(f^*_{\eta^s}). \tag{6}$$

It is also noteworthy that the claim holds for the symmetric noise case without the conditions 1 and 2.

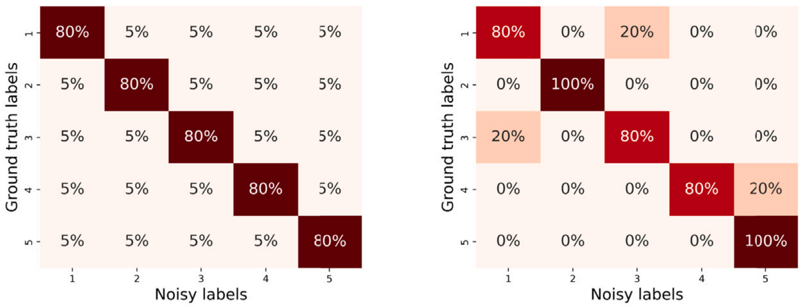
Finally, we construct an ensembled classification model by combining all the models trained in each stage as represented in Algorithm 2. In the proposed method, each stage model is trained on the dataset with distinct label information. Thus, the models trained in each stage become heterogeneous. If the models learn the same feature, we cannot expect performance improvement no matter how many models are ensembled. However, since we combine models that have learned different features, the proposed ensemble model reflects various features of the input sample so that it can perform robust classification and achieve superior performance to each stage model. Although there might be a few defective models derived from our method, the combined model is expected to work well due to the robust nature of ensemble models (Polikar, 2006; Yan, Liu, Jin, & Hauptmann, 2003). If additional techniques are required for effective performance, we can employ additional techniques, such as a weighted ensemble (Bonab & Can, 2019; Kim, Son, Lee, & Lee, 2016), to diminish the negative impact of defective models.

3.4. Complexity of the multi-stage ensemble method

The proposed multi-stage ensemble method involves an iterative process of model training and dataset refinement. Since we perform dataset refinement in each stage, the proposed method requires additional computations compared to conventional ensemble methods. To encourage a comprehensive understanding of these computations, we provide the computational complexity of the training and inference processes of our method separately.

In each stage, the time complexity of model training is $O(NMe)$, where N represents the number of samples, M is the number of learnable parameters in the model, and e is the training epochs in one stage. Subsequent to model training in each stage, the complexity of dataset refinement is upper bounded by $O(NM + NlogN + N\rho d + NCd)$, where ρ is a small loss ratio, d is the dimension of a feature vector, and C is the number of classes. Indeed, it can be considered into two parts, $O(NM + NlogN + N\rho d)$ and $O(NCd)$, which involve the construction of representative feature vectors and computation of negative Euclidean distances, respectively. In the first part, the term $O(NM)$ is from the computing RCE losses for N samples. After that, based on the Quick Sort algorithm, sorting the loss values requires a complexity of $O(NlogN)$ in the worst case when the dataset is highly imbalanced. Note that, for the best case when the number of instances for each class are equal to N/C, it could be reduced to $O(Nlog(N/C))$. Next, to construct the representative feature vectors using small loss samples in each class requires $O(N\rho d)$, as $N\rho$ samples are involved in averaging d-dimensional representative feature vectors. The remaining part in the complexity of the dataset refinement, $O(NCd)$, involves the computation of the negative Euclidean distances between the feature vectors of N samples and the C representative vectors, followed by the re-labeling.

When we set S as the total number of stages, the total computational complexity of the model training and dataset refinement over all stages is bounded to $O(SNMe + (S-1)(NM + NlogN + N\rho d + NCd))$, which can be reduced to $O(SN(Me + logN + \rho d + Cd))$. Since the number of samples, N, and learnable parameters, M, are much larger than the other factors in general, $O(SN(Me + logN))$ dominates the total complexity. In general cases, the time complexity of training S models for conventional ensemble methods is $O(SNMe)$, the additional



(a) symmetric-20%

(b) asymmetric-20%

Fig. 4. Example of clean and noisy label distributions for symmetric and asymmetric noise, respectively.

computation required by the dataset refinement process in the proposed method is $O(SN\log N)$, mainly incurred by the alignment of loss values over S stages.

Meanwhile, during inference on k input samples using our ensemble model integrating S models, the complexity of the proposed method is $O(kSM)$, consistent with conventional ensemble models. While training multiple models with our method causes additional complexity, once the training process based on the proposed method is completed, we can persistently utilize our effective ensemble model in inference without incurring extra computation costs.

4. Experiments

In this section, we validate our multi-stage ensemble with refinement using the benchmark and real-world datasets with a widely adopted setting for verifying learning algorithms under noisy labels (Han et al., 2018; Kim et al., 2021; Ma et al., 2020; Xia et al., 2020). Then, we perform an ablation study to analyze the effectiveness of relabeling and the total number of stages. Finally, we investigate the sensitivity on the small loss ratio (ρ) and confidence threshold (γ) in the proposed method. In all experiments, the proposed method is denoted as Ours.[1]

4.1. Experiment setting

Datasets. We validate the proposed method on three benchmark datasets, label noise injected MNIST (Deng, 2012), F-MNIST (Xiao, Rasul, & Vollgraf, 2017), and CIFAR-10 (Krizhevsky et al., 2009), and two real-world noisy labeled datasets, Animal-10N (Song et al., 2019) and Clothing1M (Xiao et al., 2015).

We considered symmetric and asymmetric noise to inject label noise into the benchmark datasets. To generate symmetric noisy labels, following Ma et al. (2020) and Wei et al. (2023), we selected $\frac{N\eta}{C}$ training samples from each class and randomly flipped their labels to other classes, excluding the original ground-truth labels. The asymmetric noise is generated by flipping labels within a specific set of similar classes, following previous works (Han et al., 2018; Xia et al., 2020): for MNIST, 2→7, 3→8, 5↔6, and for F-MNIST, T-SHIRT→SHIRT, PULLOVER→COAT, SANDALS→SNEAKERS, and for CIFAR-10, TRUCK→AUTOMOBILE, BIRD→AIRPLANE, DEER→HORSE, CAT↔DOG. Fig. 4 shows examples of the label distributions for symmetric and asymmetric noise-injected datasets.

Also, we used Animal-10N (Song et al., 2019) and Clothing1M (Xiao et al., 2015) datasets to verify that our method can work effectively on real-world datasets, which intrinsically have noisy labels. Animal-10N is a real-world dataset that contains human-labeled online images of 10 animals of similar appearance, where the noisy labels are naturally generated by human mistakes. The dataset provides 50k training images with noisy labels and 5k testing images only with clean labels, respectively. We used the same experimental setting following the setup of Song et al. (2019). Clothing1M contains one million clothing images with inherent noisy labels obtained from online shopping websites with 14 classes. There are also sets of images, with the sizes of 50k, 14k, and 10k, manually verified as clean samples for training, validation, and testing. Following Kim et al. (2021), we used a randomly sampled pseudo-balanced subset with 120k images as a training set instead of only using the clean one. For evaluation, we compute the classification accuracy on the 10k clean instances for testing.

Implementation. Most previous studies on learning with noisy labeled datasets employ various network architectures for different datasets (Han et al., 2018; Kim et al., 2021; Ma et al., 2020; Wei et al., 2020, 2023; Xia et al., 2020; Ye et al., 2024), leading to a lack of a standard experimental setting we can follow. For instance, they utilized a range of models from 4-layer CNN to ResNet-50 for benchmark datasets, and from ResNet-18, VGG-19 to ResNet-50 for real-world datasets. However, higher capacity models are consistently employed as the dataset difficulty increases. Hence, following this convention, we adopted higher-capacity models when we deal with more challenging datasets. Additionally, to ensure fair comparisons, we conducted extensive hyperparameter searches for each comparison method across various datasets. For MNIST and F-MNIST, we trained a 4-layer CNN and an 8-layer CNN, respectively, for 50 epochs. For CIFAR-10, we trained ResNet-34 (He, Zhang, Ren, & Sun, 2016) for 200 epochs. Regarding real-world datasets, we used the same experimental setup following previous works in Kim et al. (2021) and Song et al. (2019). Specifically, for Animal-10N, we trained VGG-19 (Simonyan & Zisserman, 2015) for 200 epochs, and for Clothing1M, we employed ResNet-50 pre-trained with the ImageNet (Deng et al., 2009) and fine-tuned for 10 epochs.

To set the balancing weights of ℓ^{APL}, we conducted a hyperparameter search, and we found the tendency that as the classification task becomes more challenging, the optimal weight of ℓ^{NCE} increases and the weight of ℓ^{RCE} decreases as follows: MNIST-{1.0, 10.0}, F-MNIST-{5.0, 0.1}, CIFAR-10-{5.0, 0.1}, Animal-10N-{5.0, 0.1} and Clothing1M-{10.0, 0.1}. Finally, we vary the small loss ratio ρ and confidence threshold γ with the difficulty of classifying the dataset. For the case of MNIST and F-MNIST, which are relatively easy datasets, we set ρ and γ as 0.1 and 0.7, respectively. Conversely, for more challenging datasets such as CIFAR-10, Animal-10N, and Clothing1M, we set ρ and γ to 0.4 and 0.99, respectively. To compare the proposed method and previous works, we set the total number of stages S to five in all experiments. Consequently, we combined five models, namely, $f_0, \ldots, f_4$, to construct the proposed ensemble model for the comparison.

[1] The code is available at https://github.com/chihyeon-choi/multi-stage-ensemble.

Table 2
Test accuracies (%) of the proposed method and *single* model for comparison methods on **symmetric noise** $\in \{0.2, 0.4, 0.6, 0.8\}$ or **asymmetric noise** $\in \{0.1, 0.2, 0.3, 0.4\}$ injected benchmark datasets. The average accuracies and standard deviations over three trials are reported. Ours denote the proposed multi-stage ensemble method. The best and second-best results are denoted as **boldfaced** and underlined, respectively.

Datasets	Methods	Sym-20%	Sym-40%	Sym-60%	Sym-80%	Asym-10%	Asym-20%	Asym-30%	Asym-40%
MNIST	CE	96.72 ± 0.09	92.20 ± 0.70	79.81 ± 1.41	41.19 ± 0.93	97.85 ± 0.65	95.83 ± 0.66	90.51 ± 1.88	83.55 ± 3.47
	Co-teach	98.56 ± 0.12	97.99 ± 0.11	97.36 ± 0.27	86.23 ± 0.70	99.25 ± 0.05	98.95 ± 0.03	98.62 ± 0.17	97.11 ± 0.31
	FINE	98.57 ± 0.06	97.99 ± 0.12	97.36 ± 0.17	81.83 ± 1.92	98.92 ± 0.08	98.75 ± 0.16	98.40 ± 0.34	97.54 ± 0.81
	CDR	98.20 ± 0.01	97.57 ± 0.17	95.91 ± 0.17	75.30 ± 1.46	98.79 ± 0.03	98.34 ± 0.17	97.37 ± 0.20	90.37 ± 1.56
	APL	99.05 ± 0.12	98.62 ± 0.07	98.07 ± 0.11	86.43 ± 1.05	99.17 ± 0.07	99.06 ± 0.04	98.82 ± 0.18	87.65 ± 1.74
	JoCoR	98.39 ± 0.02	97.67 ± 0.08	96.26 ± 0.11	86.52 ± 5.85	99.07 ± 0.03	98.71 ± 0.02	98.41 ± 0.05	97.15 ± 0.14
	LogitClip	98.80 ± 0.05	98.48 ± 0.15	97.06 ± 0.18	84.25 ± 2.70	99.07 ± 0.06	99.01 ± 0.08	98.76 ± 0.14	97.35 ± 0.16
	ANL	98.88 ± 0.12	98.55 ± 0.06	98.25 ± 0.28	86.32 ± 1.37	99.05 ± 0.12	98.82 ± 0.12	98.79 ± 0.11	98.35 ± 0.31
	Ours	**99.30 ± 0.02**	**99.09 ± 0.07**	**98.80 ± 0.07**	**93.95 ± 0.11**	**99.37 ± 0.04**	**99.32 ± 0.09**	**99.18 ± 0.03**	**98.67 ± 0.21**
F-MNIST	CE	86.78 ± 0.11	80.11 ± 0.51	65.34 ± 0.49	31.14 ± 0.30	90.59 ± 0.37	87.86 ± 0.12	83.63 ± 0.32	79.56 ± 0.58
	Co-teach	89.55 ± 0.16	88.68 ± 0.11	87.07 ± 0.46	73.64 ± 0.39	90.23 ± 0.16	89.59 ± 0.12	87.86 ± 0.39	83.22 ± 1.11
	FINE	89.67 ± 0.26	87.96 ± 0.26	86.26 ± 0.12	74.12 ± 1.05	90.91 ± 0.28	89.75 ± 0.51	88.59 ± 0.20	84.03 ± 0.85
	CDR	89.66 ± 0.22	87.90 ± 0.24	85.75 ± 0.46	74.90 ± 0.31	91.18 ± 0.18	90.08 ± 0.15	87.71 ± 0.18	82.07 ± 0.77
	APL	91.11 ± 0.21	89.71 ± 0.06	87.87 ± 0.18	77.50 ± 0.01	91.81 ± 0.04	90.70 ± 0.22	88.76 ± 0.30	80.79 ± 1.09
	JoCoR	90.77 ± 0.08	89.91 ± 0.16	88.26 ± 0.20	77.06 ± 2.54	91.97 ± 0.08	91.15 ± 0.19	90.60 ± 0.07	88.96 ± 1.22
	LogitClip	90.72 ± 0.30	89.68 ± 0.35	87.24 ± 0.71	78.47 ± 0.90	92.19 ± 0.24	91.40 ± 0.27	91.17 ± 0.30	88.23 ± 0.67
	ANL	91.50 ± 0.20	90.36 ± 0.23	88.07 ± 0.27	77.44 ± 1.28	91.86 ± 0.31	91.31 ± 0.52	89.68 ± 0.51	87.68 ± 1.39
	Ours	**93.18 ± 0.12**	**92.06 ± 0.14**	**89.56 ± 0.49**	**81.41 ± 0.17**	**93.65 ± 0.17**	**93.10 ± 0.22**	**92.27 ± 0.38**	**90.46 ± 0.48**
CIFAR-10	CE	79.17 ± 0.43	59.85 ± 1.85	39.49 ± 1.77	18.19 ± 0.45	91.48 ± 0.33	86.99 ± 0.43	84.74 ± 0.51	80.96 ± 0.92
	Co-teach	88.89 ± 0.29	85.12 ± 0.21	68.93 ± 4.36	19.17 ± 3.13	92.64 ± 0.16	91.57 ± 0.16	89.36 ± 0.07	70.91 ± 0.23
	FINE	91.13 ± 0.16	88.05 ± 0.24	81.07 ± 0.74	42.68 ± 2.65	92.72 ± 0.15	92.42 ± 0.27	91.81 ± 0.41	**90.14 ± 0.71**
	CDR	89.47 ± 0.16	83.93 ± 0.36	83.24 ± 0.31	35.64 ± 1.35	92.10 ± 0.22	90.84 ± 0.03	89.71 ± 0.32	85.92 ± 0.80
	APL	92.01 ± 0.07	89.17 ± 0.13	83.64 ± 0.29	48.39 ± 4.21	93.04 ± 0.20	91.94 ± 0.24	90.03 ± 0.25	86.07 ± 0.25
	JoCoR	89.85 ± 0.23	87.34 ± 0.19	63.22 ± 1.14	35.60 ± 1.79	89.56 ± 0.19	89.24 ± 0.44	87.46 ± 0.19	82.87 ± 0.65
	LogitClip	84.76 ± 0.11	80.21 ± 0.30	71.41 ± 0.09	43.02 ± 0.57	85.11 ± 0.31	84.49 ± 0.05	83.38 ± 0.38	77.89 ± 0.74
	ANL	91.04 ± 0.13	87.66 ± 0.17	80.47 ± 0.20	40.21 ± 0.81[a]	91.83 ± 0.11	90.93 ± 0.30	88.40 ± 0.12	82.19 ± 0.98
	Ours	**94.01 ± 0.12**	**92.01 ± 0.16**	**87.09 ± 0.16**	**65.40 ± 0.41**	**94.28 ± 0.19**	**93.78 ± 0.12**	**92.52 ± 0.11**	89.58 ± 0.38

[a] In the original reference (Ye et al., 2024), the accuracy for the case of CIFAR-10 Sym-80% was reported as 61.27%.

Comparison methods. To demonstrate the effectiveness of the proposed method, we compared Ours with the following methods.

- *CE* is a model trained with CE loss without any further techniques for noisy labels.
- *Co-teach* (Han et al., 2018) is a sample selection method that selects probably clean samples based on two networks with the same architecture to minimize the selection bias. It requires a noise ratio to select small loss samples.
- *FINE* (Kim et al., 2021) is a sample selection method that filters noisy samples utilizing the class-representative features obtained through the eigen decomposition of latent representations for each class.
- *CDR* (Xia et al., 2020) is a regularization-based method that imposes penalties only on the parameters of the model that overfit to noisy labels. It requires a noise ratio to impose penalties.
- *APL* (Ma et al., 2020) is a robust loss function-based method that introduced active passive loss, which has a sufficient learning property.
- *JoCoR* (Wei et al., 2020) is a sample selection method that employs a joint loss to select low-loss data. It requires a noise ratio to select small loss samples.
- *LogitClip* (Wei et al., 2023) is a regularization-based method that clamps the norm of the logit vector to ensure it is upper bounded by a constant.
- *ANL* (Ye et al., 2024) is a robust loss function-based method that utilizes a new class of robust passive loss functions to focus on memorized clean samples and speed up the convergence of the model.

Since the proposed method performs classification based on the ensemble model, we experimented with both *single* and *ensemble* models of the existing methods and compared them with the proposed method for a fair comparison. For the *ensemble* model of the comparison methods, we combined five individual models trained with randomly initialized weights in order to enhance the independence of the models.

Note that we do not include semi-supervised technique based methods (Li, Socher, & Hoi, 2020; Nguyen et al., 2020; Zhou, Wang, & Bilmes, 2021) for comparison because they took advantage of existing powerful representation learning approaches, such as consistency regularization with various image augmentation techniques. We focused on validating the efficacy of the proposed ensemble method based on the dataset refinement without such techniques.

4.2. Experiment results

Results on benchmark datasets. Table 2 shows the classification accuracy of the proposed method and *single* models of comparison methods for the benchmark datasets with noisy labels. When generating label noise, we considered diverse degrees of noise ratios to evaluate various scenarios. After training each model on the noisy labeled training dataset, we evaluated the classification accuracy on the clean test dataset. Table 2 shows that the proposed method achieved the best classification performance in all cases except for Asym-40% of CIFAR-10. In particular, for all benchmark datasets, Ours significantly outperformed other approaches in the Sym-80% case, which contains the largest number of noisy labels in each dataset.

Further, as the proposed method is based on the ensemble model, we compared the proposed method with the *ensemble* versions of the other methods in Table 3 to show that the performance gain of the proposed method is not simply derived from the ensemble. As demonstrated in Table 3, Ours achieved the best performance in all symmetric noise cases except for F-MNIST Sym-60% and Sym-80%. For MNIST, which is a relatively less challenging dataset, most existing methods achieved promising performance, even with a considerable amount of noisy labels. However, in Sym-80%, all of the methods showed a notable performance drop compared to Sym-60%. Among them, Ours showed the least degradation as the accuracy decreased by 4.85%. For F-MNIST and CIFAR-10, Ours achieved superior performance to the other methods in most cases similar to the results of MNIST. In particular, in the Sym-80% case of CIFAR-10, where almost all

Table 3
Test accuracies (%) of the proposed method and *ensemble* model for comparison methods on **symmetric noise** $\in \{0.2, 0.4, 0.6, 0.8\}$ or **asymmetric noise** $\in \{0.1, 0.2, 0.3, 0.4\}$ injected benchmark datasets. The average accuracies and standard deviations over three trials are reported. Ours denote the proposed multi-stage ensemble method. The best and second-best results are denoted as **boldfaced** and underlined, respectively.

Datasets	Methods	Sym-20%	Sym-40%	Sym-60%	Sym-80%	Asym-10%	Asym-20%	Asym-30%	Asym-40%
MNIST	CE	98.45 ± 0.06	97.17 ± 0.03	91.44 ± 0.40	52.04 ± 0.59	99.11 ± 0.03	98.01 ± 0.18	93.93 ± 1.30	87.62 ± 1.72
	Co-teach	98.97 ± 0.05	98.56 ± 0.02	98.15 ± 0.08	89.93 ± 0.09	**99.42 ± 0.04**	99.10 ± 0.12	99.03 ± 0.05	98.46 ± 0.10
	FINE	98.78 ± 0.06	98.61 ± 0.11	98.69 ± 0.10	84.75 ± 0.59	99.22 ± 0.05	99.10 ± 0.04	98.93 ± 0.10	98.52 ± 0.22
	CDR	98.68 ± 0.09	98.29 ± 0.02	97.09 ± 0.11	84.48 ± 0.69	99.15 ± 0.05	98.93 ± 0.05	98.35 ± 0.06	95.23 ± 1.17
	APL	99.08 ± 0.33	99.03 ± 0.06	98.73 ± 0.08	88.22 ± 0.06	99.35 ± 0.05	99.18 ± 0.06	99.03 ± 0.09	93.29 ± 2.59
	JoCoR	98.98 ± 0.02	98.58 ± 0.01	97.84 ± 0.07	89.92 ± 0.11	99.32 ± 0.02	99.16 ± 0.02	98.92 ± 0.02	98.34 ± 0.04
	LogitClip	98.14 ± 0.02	98.79 ± 0.02	97.74 ± 0.03	87.12 ± 0.35	99.26 ± 0.04	99.30 ± 0.03	99.01 ± 0.03	98.34 ± 0.15
	ANL	99.17 ± 0.03	99.04 ± 0.04	98.69 ± 0.04	87.96 ± 0.11	99.32 ± 0.02	99.22 ± 0.02	99.13 ± 0.04	98.63 ± 0.55
	Ours	**99.30 ± 0.02**	**99.09 ± 0.07**	**98.80 ± 0.07**	**93.95 ± 0.11**	99.37 ± 0.04	**99.32 ± 0.09**	**99.18 ± 0.03**	**98.67 ± 0.21**
F-MNIST	CE	91.01 ± 0.21	88.02 ± 0.11	78.03 ± 0.15	40.87 ± 0.32	92.36 ± 0.25	90.54 ± 0.11	87.02 ± 0.46	81.33 ± 0.43
	Co-teach	90.97 ± 0.00	89.86 ± 0.00	88.42 ± 0.00	79.31 ± 0.00	92.14 ± 0.00	91.07 ± 0.00	89.60 ± 0.00	85.80 ± 0.00
	FINE	91.16 ± 0.17	89.45 ± 0.13	87.64 ± 0.30	80.14 ± 0.26	92.14 ± 0.10	91.48 ± 0.09	90.52 ± 0.08	88.20 ± 0.20
	CDR	90.90 ± 0.08	90.02 ± 0.14	88.24 ± 0.36	79.98 ± 0.37	92.36 ± 0.07	91.50 ± 0.15	90.20 ± 0.17	84.61 ± 0.51
	APL	92.04 ± 0.14	90.66 ± 0.29	88.74 ± 0.38	78.83 ± 1.49	92.48 ± 0.12	91.55 ± 0.08	90.04 ± 0.35	84.19 ± 0.13
	JoCoR	92.05 ± 0.08	91.23 ± 0.02	**90.00 ± 0.07**	**81.66 ± 1.00**	93.28 ± 0.03	92.40 ± 0.10	91.16 ± 0.16	87.41 ± 0.51
	LogitClip	92.23 ± 0.03	90.94 ± 0.07	89.43 ± 0.16	80.90 ± 0.18	93.28 ± 0.07	92.95 ± 0.02	91.99 ± 0.17	89.16 ± 0.45
	ANL	92.64 ± 0.08	91.24 ± 0.02	89.36 ± 0.20	79.98 ± 0.28	93.19 ± 0.09	92.68 ± 0.03	90.78 ± 0.01	88.36 ± 0.34
	Ours	**93.18 ± 0.12**	**92.06 ± 0.14**	89.56 ± 0.49	81.41 ± 0.17	**93.65 ± 0.17**	**93.10 ± 0.22**	**92.27 ± 0.38**	**90.46 ± 0.48**
CIFAR-10	CE	88.87 ± 0.32	76.29 ± 0.76	53.95 ± 0.47	22.78 ± 0.27	93.74 ± 0.07	90.34 ± 0.56	87.97 ± 0.11	83.34 ± 0.89
	Co-teach	92.02 ± 0.00	88.72 ± 0.00	77.74 ± 0.00	28.29 ± 0.00	94.04 ± 0.00	92.78 ± 0.00	90.35 ± 0.00	71.67 ± 0.00
	FINE	92.46 ± 0.17	89.78 ± 0.13	83.60 ± 0.30	45.01 ± 0.26	94.07 ± 0.10	93.73 ± 0.09	**93.18 ± 0.08**	**92.10 ± 0.20**
	CDR	91.64 ± 0.08	87.80 ± 0.14	78.26 ± 0.36	41.68 ± 0.37	93.50 ± 0.07	92.66 ± 0.15	91.54 ± 0.17	88.47 ± 0.51
	APL	93.37 ± 0.14	90.97 ± 0.29	85.56 ± 0.38	49.14 ± 1.49	94.11 ± 0.12	93.28 ± 0.08	91.43 ± 0.35	87.57 ± 0.13
	JoCoR	91.63 ± 0.15	88.87 ± 0.08	67.21 ± 0.29	40.48 ± 0.49	90.84 ± 0.04	90.64 ± 0.15	88.97 ± 0.07	83.76 ± 0.61
	LogitClip	87.28 ± 0.12	83.53 ± 0.21	75.81 ± 0.15	46.02 ± 0.27	87.64 ± 0.06	86.92 ± 0.20	85.68 ± 0.13	80.74 ± 1.47
	ANL	92.67 ± 0.08	89.35 ± 0.11	83.39 ± 0.14	42.49 ± 0.40	93.23 ± 0.16	92.22 ± 0.11	90.05 ± 0.14	84.56 ± 0.05
	Ours	**94.01 ± 0.12**	**92.01 ± 0.16**	**87.09 ± 0.16**	**65.40 ± 0.41**	**94.28 ± 0.19**	**93.78 ± 0.12**	92.52 ± 0.11	89.58 ± 0.38

Table 4
Test accuracies (%) of the proposed method and *ensemble* model for comparison methods on Animal-10N and Clothing1M dataset. The average accuracies and standard deviations over three trials are reported. Ours denote the proposed multi-stage ensemble method. The best and second-best results are denoted as **boldfaced** and underlined, respectively. The value in parentheses denotes the *single* model performance.

Methods	CE	Co-teach	FINE	CDR	APL	JoCoR	LogitClip	ANL	Ours
Animal-10N	83.65 ± 0.13 (80.80 ± 0.38)	78.05 ± 0.29 (74.88 ± 0.46)	84.17 ± 0.24 (81.14 ± 0.03)	84.41 ± 0.08 (81.24 ± 0.55)	84.05 ± 0.24 (81.72 ± 0.02)	76.96 ± 0.09 (74.13 ± 0.70)	84.59 ± 0.27 (81.91 ± 0.15)	84.19 ± 0.06 (82.38 ± 0.25)	**85.47 ± 0.28**
Clothing1M	72.46 ± 0.20 (70.15 ± 0.38)	65.34 ± 0.15 (65.17 ± 0.10)	72.30 ± 0.09 (70.73 ± 0.57)	72.27 ± 0.10 (71.73 ± 0.28)	73.06 ± 0.13 (71.73 ± 0.17)	66.75 ± 0.13 (66.47 ± 0.03)	72.45 ± 0.14 (71.76 ± 0.18)	73.26 ± 0.23 (71.47 ± 0.50)	**73.79 ± 0.06**

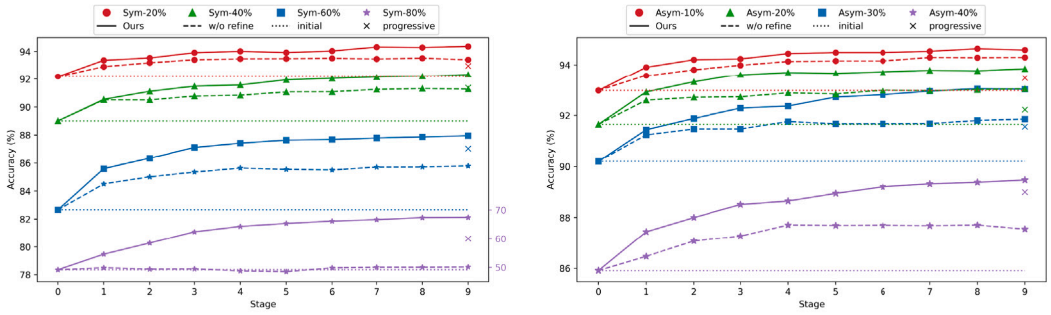
comparison methods struggled, Ours significantly outperformed the others. In that case, none of the comparison methods achieved an accuracy over 50%, while the proposed method exhibited a 65.40% accuracy, demonstrating a remarkable improvement of 16.26% compared to the second-best performance. Also, in the cases of asymmetric noise, Ours showed effective performance as it achieved the best or second-best performances in most cases. Since we forced each model to learn different features through our dataset refinement method, our ensemble model still works better than simple ensemble versions of the existing methods. In summary, the proposed method showed superior performance compared to both single and ensemble models of the existing methods, and our multi-stage ensemble method with refinement is validated to be effective in classification under various types and levels of noisy labels.

Results on real-world datasets. To verify the proposed method on the real-world datasets, we employed two datasets with intrinsic noisy labels, Animal-10N (Song et al., 2019) and Clothing1M (Xiao et al., 2015), in which the estimated noise ratios are around 8% and 38.5%, respectively. Since Co-teach, CDR, and JoCoR require the noise ratio of the dataset for model training, we set the noise ratios for them to match the estimated noise ratios of the real-world datasets. Table 4 shows the classification accuracy of the *ensemble* models on the clean test dataset after training each model on the noisy training dataset. Similar to the results of the benchmark datasets, Ours showed better classification accuracy than the comparison methods in both datasets. Also, unlike the methods that require the noise ratio of the dataset in advance, such as Co-teach, CDR, and JoCoR, the proposed method achieved the best performance without any prior information on noise ratio, which is challenging to identify in practice. Therefore, we confirmed that our method could work effectively not only on synthetic noisy labeled datasets but also on the real-world ones.

4.3. Ablation study

Using the CIFAR-10 dataset, we performed an ablation study to examine the effects of the refinement process and the total number of stages, and the results are demonstrated in Fig. 5. In Fig. 5, the *y*-axis on the right and left sides represent the accuracy of the Sym-80% case and that of the others, respectively. In this experiment, Ours denotes the ensemble model constructed by the proposed method, while *w/o refine* represents the ensemble model without the refinement process. Meanwhile, *initial* and *progressive* indicate the single model trained in the initial stage and progressively trained until the last stage, respectively.

First, when we compare the performance of Ours with *w/o refine*, Ours shows better performance than *w/o refine* in every case as the stage proceeds. In particular, in both symmetric and asymmetric noise cases, the higher the label noise, the larger the effect of re-labeling. Despite Sym-20% and Asym-10% being less corrupted cases among the noise-injected datasets, the proposed refinement and ensemble approach still enables Ours to outperform the *w/o refine*. Moreover, for the Sym-80% and Asym-40%, where the datasets contain a large



(a) Symmetric noise

(b) Asymmetric noise

Fig. 5. Results of the ablation study on the effect of the refinement process and the total number of stages. x and y-axes represent the number of stages and classification performance, respectively. Noise ratios and types are indicated by colors and markers on lines. The solid line depicts the results of Ours, while the dashed and dotted lines represent those of the ensemble model without refinement and the single model, respectively. The cross mark represents the result of the single model trained in a progressive manner until the last stage.

portion of noisy labels, Ours significantly improves *w/o refine* and *initial* immediately after stage 0. However, in the Sym-80%, even though *w/o refine* uses an ensemble approach, it performs worse than *initial* in some stages. Thus, we can infer that it is difficult to learn appropriate features without the refinement process when the dataset is severely corrupted. We also analyze the performance of the *progressive*, which is a trained model in a progressive manner until the last stage. With the help of the dataset refinement, *progressive* achieves significantly improved performance compared to *single*. Especially when there are considerable noisy labels in the dataset, *progressive* performs better than the *w/o refine*, which is an ensemble model. Although the proposed method brings more benefits by integrating progressively trained models in every stage that have learned diverse features, the results of *progressive* also support the importance of the dataset refinement. Based on these results, we found that the refinement process is essential in learning different features for each stage model and constructing an effective ensemble classifier under the noisy labeled dataset.

Next, we analyze the effect of the total number of stages in the proposed method. In both symmetric and asymmetric noises, we can see that the performance of Ours and *w/o refine* continues to increase as the stage proceeds. For Ours, when the noise ratio is small, the performance increases slowly at the beginning and achieves a reasonable performance in the early stages. However, when the noise ratio is large, the performance increases rapidly at the beginning and continues to increase until almost the last stage set to be 10. Although we can expect performance improvement regardless of the noise type and ratios as the stage proceeds, our approach can achieve decent performance with substantial performance gain over *w/o refine* unless the total number of stages is too small.

4.4. Hyperparameter sensitivity analysis

The proposed method includes two important hyperparameters, the small loss ratio (ρ) and the confidence threshold (γ). Here, we conducted sensitivity analyses on these hyperparameters using the benchmark datasets. The small loss ratio controls the proportion of the small RCE loss samples for each class, which consist of representative feature vectors. A lower small loss ratio allows us to construct a representative feature vector using samples with lower RCE loss, that is, a small number of samples with a large proportion of clean labels, as demonstrated in Table 1. Conversely, a higher small loss ratio enables the construction of a representative feature vector using a larger number of samples with small RCE loss, even though the proportion of clean labels might be reduced compared to the former case. The confidence threshold adapts the strategy of the re-labeling in the refinement phase. With a high confidence threshold, we can re-label only for noisy samples very close to the representative feature vector for a particular class. Such a conservative strategy enables re-labeling only for limited samples with high accuracy. In contrast, a low confidence threshold allows for the re-labeling of a larger number of samples. However, such an aggressive strategy may cause more samples with incorrectly re-labeled. For the sensitivity analysis, we constructed an ensemble model by combining the models of the stages 0 to 4 and investigated the performance by varying each hyperparameter as follows: $\rho \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and $\gamma \in \{0.7, 0.8, 0.9, 0.95, 0.99\}$. Fig. 6 depicts sensitivity analysis results on both simple datasets MNIST and F-MNIST, as well as the challenging dataset CIFAR-10.

Regardless of the difficulty of the dataset, the variations in the small loss ratio and confidence threshold do not significantly influence on the classification accuracy of the proposed method in most cases. However, in the case of Sym-80% and Asym-40%, where the datasets contain a substantial number of noisy samples, the hyperparameter should be carefully chosen, taking into account the difficulty of the dataset.

For MNIST and F-MNIST datasets, the model performance tends to improve as we set the lower ρ and γ within the range explored during the hyperparameter search. As shown in Fig. 6, the optimal hyperparameter of the small loss ratio and confidence threshold was identified to be 0.1 and 0.7, respectively. When we deal with a simple dataset with noisy labels, it is found to be effective to form a representative feature vector with limited small loss samples and re-label with an aggressive strategy for effective classification.

For the CIFAR-10 dataset, relatively high ρ and γ improve the classification performance in the Sym-80% and Asym-40% case. We observed that the best performance is achieved for most noise-injected datasets derived from CIFAR-10 when we set ρ and γ as 0.4 and 0.99, respectively. The results indicate that, for the challenging dataset, constructing representative feature vectors with a relatively higher proportion of small loss samples and adopting a conservative re-labeling strategy is effective.

5. Conclusion

We introduced a novel multi-stage ensemble with refinement for noisy label data classification. Our main idea is to force the models trained in each stage to learn diverse features to construct an effective ensemble classifier based on the noisy labeled dataset. To achieve the goal, we proposed an iterative process that involves refining the dataset and training the model progressively with the refined dataset. The proposed approach enables each stage model to learn distinctive features with the iteratively refined datasets, ultimately constructing the successful ensemble classifier by combining them. Through the experiments on both benchmark and real-world datasets, we validated that the proposed method outperformed the ensemble models of the existing methods. Moreover, the results of the ablation study supported

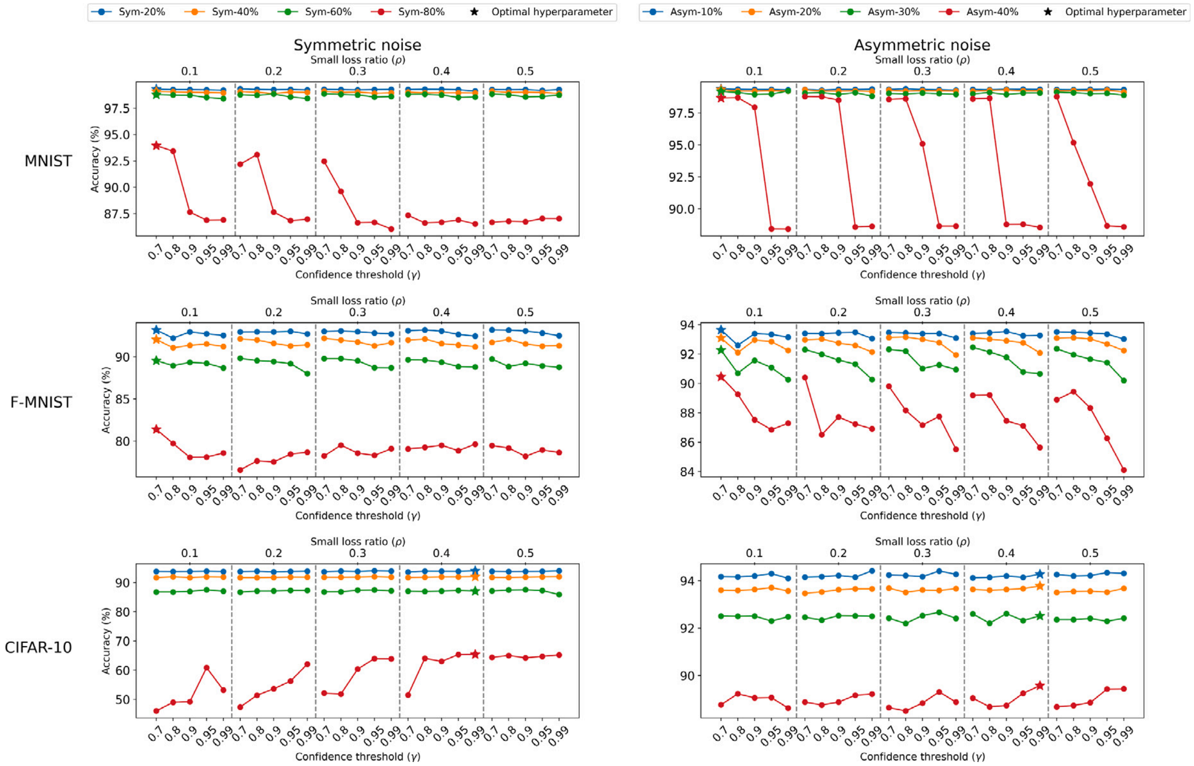


Fig. 6. Sensitivity analysis results for the small loss ratio (ρ) and confidence threshold (γ) on benchmark datasets. The first and second columns present the results for datasets injected with symmetric and asymmetric label noise, respectively. In each figure, the top and bottom of the x-axis indicate the small loss ratio and confidence threshold, respectively. Different colors in each figure indicate various levels of noise ratio. Asterisk marks denote optimal hyperparameters identified for each dataset.

that our key ideas are suitable for being applied to corrupted datasets. Finally, we showed that the proposed method is not sensitive to the hyperparameters in most cases. While some variations in classification performance may occur as the selection of hyperparameters in extremely corrupted datasets, we found that the effective ensemble model can be constructed with an appropriate hyperparameter choice according to the difficulty of the dataset.

Although the proposed multi-stage ensemble method shows promising results, there are some limitations and future research directions for further investigations. First, we validated the proposed multi-stage ensemble method on balanced datasets with similar levels of noisy labels across different classes. Therefore, additional analysis might be required for the datasets with different noise levels across the classes or those exhibiting long-tailed class distributions, as well as other practical challenges, to expand the applicability of the proposed method in various real-world scenarios. Second, the proposed method entails a time-consuming task of searching for appropriate hyperparameters, especially for highly contaminated datasets to achieve effective generalization performance. Since the optimal ones might differ depending on the dataset, careful hyperparameter tuning is necessary. If the guidelines for hyperparameter selection are provided depending on the difficulty of the datasets in further study, it can facilitate the construction of the effective ensemble model under the corrupted datasets, avoiding the need for labor-intensive adjustment.

CRediT authorship contribution statement

Chihyeon Choi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Woojin Lee:** Methodology, Validation, Investigation, Writing – original draft, Writing – review & editing, Supervision, Funding acquisition. **Youngdoo Son:** Methodology, Validation, Formal analysis, Investigation, Resources, Writing – original draft, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All datasets used in this study are publicly available.

Acknowledgments

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT (MSIT) of Korea (No. RS-2023-00208412, 2022R1F1A1074393).

Appendix. Proof of Proposition 1

Proof. According to Lemma 2 in Ma et al. (2020), $f^*_{\eta^s} = f^*_{\eta^{s+1}} = f^*, \forall s \geq 0$ and

$$R^{\eta^s}(f) = \mathbb{E}_{\boldsymbol{x},y^*}(1-\sum_{j=1}^{C}\eta^s_{y^*j}) - \mathbb{E}_{\boldsymbol{x},y^*}[\sum_{k\neq y^*}(1-\eta^s_{y^*k}-\sum_{j=1}^{C}\eta^s_{y^*j})\ell^{norm}(f(\boldsymbol{x}),k)] \tag{A.1}$$

where y^* is the true label of $\boldsymbol{x}$. Then, from the definition of the normalized loss function, $R(f^*) = 0$, and $0 \leq \ell^{norm}(f^*(x), j) \leq \frac{1}{C-1}$, we have $\ell^{norm}(f^*(\boldsymbol{x}), k) = \frac{1}{C-1}, \forall k \neq y^*$. Thus,

$$R^{\eta^s}(f^*_{\eta^s}) - R^{\eta^{s+1}}(f^*_{\eta^{s+1}})$$

$$= \mathbb{E}_{\boldsymbol{x},y^*}(1-\sum_{j=1}^{C}\eta^s_{y^*j}) - \mathbb{E}_{\boldsymbol{x},y^*}[\sum_{k\neq y^*}(1-\eta^s_{y^*k}-\sum_{j=1}^{C}\eta^s_{y^*j})\ell^{norm}(f^*_{\eta^s}(\boldsymbol{x}),k)]$$

$$
-\ \mathbb{E}_{\boldsymbol{x},y^*}(1-\sum_{j=1}^{C}\eta^{s+1}_{y^*j})+\mathbb{E}_{\boldsymbol{x},y^*}[\sum_{k\neq y^*}(1-\eta^{s+1}_{y^*k}-\sum_{j=1}^{C}\eta^{s+1}_{y^*j})\ell^{norm}(f^*_{\eta^{s+1}}(\boldsymbol{x}),k)]
$$

$$
=\mathbb{E}_{\boldsymbol{x},y^*}[\frac{1}{C-1}\sum_{k\neq y^*}(\eta^{s}_{y^*k}-\eta^{s+1}_{y^*k})]\geq 0, \tag{A.2}
$$

which completes the proof. □

References

Allen-Zhu, Z., & Li, Y. (2023). Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In *The eleventh international conference on learning representations*. URL: https://openreview.net/forum?id=Uuf2q9TfXGA.

Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., et al. (2017). A closer look at memorization in deep networks. In *International conference on machine learning* (pp. 233–242). PMLR.

Bonab, H., & Can, F. (2019). Less is more: A comprehensive framework for the number of components of ensemble classifiers. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(9), 2735–2745.

Chen, P., Liao, B. B., Chen, G., & Zhang, S. (2019). Understanding and utilizing deep neural networks trained with noisy labels. In *International conference on machine learning* (pp. 1062–1070). PMLR.

Chen, Y., Shen, X., Hu, S. X., & Suykens, J. A. (2021). Boosting co-teaching with compression regularization for label noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2688–2692).

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, *29*(6), 141–142.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). Ieee.

Fu, H., Zhang, K., & Wang, J. (2024). An adaptive self-correction joint training framework for person re-identification with noisy labels. *Expert Systems with Applications*, *238*, Article 121771.

Ghosh, A., Kumar, H., & Sastry, P. S. (2017). Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence, vol. 31, no. 1*.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., et al. (2018). Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in Neural Information Processing Systems*, *31*.

Harkat, H., Nascimento, J. M., Bernardino, A., & Ahmed, H. F. T. (2023). Fire images classification based on a handcraft approach. *Expert Systems with Applications*, *212*, Article 118594.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., & Fei-Fei, L. (2018). Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning* (pp. 2304–2313). PMLR.

Kim, T., Ko, J., Choi, J., Yun, S.-Y., et al. (2021). Fine samples for learning with noisy labels. *Advances in Neural Information Processing Systems*, *34*, 24137–24149.

Kim, N., Son, Y., Lee, Y., & Lee, J. (2016). Self-correcting ensemble using a latent consensus model. *Applied Soft Computing*, *47*, 262–270.

Krizhevsky, A., Hinton, G., et al. (2009). *Learning multiple layers of features from tiny images*. Citeseer.

Lee, K.-H., He, X., Zhang, L., & Yang, L. (2018). Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5447–5456).

Lee, S., & Son, Y. (2022). Multitask learning with single gradient step update for task balancing. *Neurocomputing*, *467*, 442–453.

Li, J., Socher, R., & Hoi, S. C. (2020). DivideMix: Learning with noisy labels as semi-supervised learning. In *International conference on learning representations*.

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., & Li, L.-J. (2017). Learning from noisy labels with distillation. In *Proceedings of the IEEE international conference on computer vision* (pp. 1910–1918).

Liu, S., Li, Y., Chai, Q.-w., & Zheng, W. (2024). Region-scalable fitting-assisted medical image segmentation with noisy labels. *Expert Systems with Applications*, *238*, Article 121926.

Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., & Bailey, J. (2020). Normalized loss functions for deep learning with noisy labels. In *International conference on machine learning* (pp. 6543–6553). PMLR.

Mandal, D., Bharadwaj, S., & Biswas, S. (2020). A novel self-supervised re-labeling approach for training with noisy labels. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision* (pp. 1381–1390).

Mehnatkesh, H., Jalali, S. M. J., Khosravi, A., & Nahavandi, S. (2023). An intelligent driven deep residual learning framework for brain tumor classification using MRI images. *Expert Systems with Applications*, *213*, Article 119087.

Mirzasoleiman, B., Cao, K., & Leskovec, J. (2020). Coresets for robust training of deep neural networks against noisy labels. *Advances in Neural Information Processing Systems*, *33*, 11465–11477.

Nguyen, D. T., Mummadi, C. K., Ngo, T. P. N., Nguyen, T. H. P., Beggel, L., & Brox, T. (2020). SELF: Learning to filter noisy labels with self-ensembling. In *International conference on learning representations*. URL: https://openreview.net/forum?id=HkgsPhNYPS.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, *6*(3), 21–45.

Shen, Y., & Sanghavi, S. (2019). Learning with bad training data via iterative trimmed loss minimization. In *International conference on machine learning* (pp. 5739–5748). PMLR.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *3rd international conference on learning representations* (pp. 1–14). Computational and Biological Learning Society.

Song, H., Kim, M., & Lee, J.-G. (2019). Selfie: Refurbishing unclean samples for robust deep learning. In *International conference on machine learning* (pp. 5907–5915). PMLR.

Tanaka, D., Ikami, D., Yamasaki, T., & Aizawa, K. (2018). Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5552–5560).

Vahdat, A. (2017). Toward robustness against label noise in training deep discriminative neural networks. *Advances in Neural Information Processing Systems*, *30*.

Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., & Belongie, S. (2017). Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 839–847).

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., & Bailey, J. (2019). Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 322–330).

Wei, H., Feng, L., Chen, X., & An, B. (2020). Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 13726–13735).

Wei, H., Zhuang, H., Xie, R., Feng, L., Niu, G., An, B., et al. (2023). Mitigating memorization of noisy labels by clipping the model prediction. In *International conference on machine learning* (pp. 36868–36886). PMLR.

Wu, P., Zheng, S., Goswami, M., Metaxas, D., & Chen, C. (2020). A topological filter for learning with label noise. *Advances in Neural Information Processing Systems*, *33*, 21382–21393.

Xia, X., Liu, T., Han, B., Gong, C., Wang, N., Ge, Z., et al. (2020). Robust early-learning: Hindering the memorization of noisy labels. In *International conference on learning representations*.

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747.

Xiao, T., Xia, T., Yang, Y., Huang, C., & Wang, X. (2015). Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2691–2699).

Yan, R., Liu, Y., Jin, R., & Hauptmann, A. (2003). On predicting rare classes with SVM ensembles in scene classification. In *2003 IEEE international conference on acoustics, speech, and signal processing, 2003. Proceedings, vol. 3* (pp. III–21). IEEE.

Ye, X., Li, X., Liu, T., Sun, Y., Tong, W., et al. (2024). Active negative loss functions for learning with noisy labels. *Advances in Neural Information Processing Systems*, *36*.

Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., & Sugiyama, M. (2019). How does disagreement help generalization against label corruption? In *International conference on machine learning* (pp. 7164–7173). PMLR.

Yuan, B., Chen, J., Zhang, W., Tai, H.-S., & McMains, S. (2018). Iterative cross learning on noisy labels. In *2018 IEEE winter conference on applications of computer vision* (pp. 757–765). IEEE.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, *64*(3), 107–115.

Zhang, Z., & Sabuncu, M. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in Neural Information Processing Systems*, *31*.

Zhou, T., Wang, S., & Bilmes, J. (2021). Robust curriculum learning: from clean label detection to noisy label self-correction. In *International conference on learning representations*.