VRLVMix: Combating Noisy Labels with Sample Selection based on Loss Variation

HaiLun Wang
AHU-IAI AI Joint Laboratory, Anhui University
Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
Hefei, China
2017217789@mail.hfut.edu.cn

[*] Zhengzheng Tu
Anhui Provincial Key Laboratory of Multimodal Cognitive Computation,
School of Computer Science and Technology, Anhui University,
Hefei, China
[*] Corresponding author: zhengzhengahu@163.com

Bo Jiang
Anhui Provincial Key Laboratory of Multimodal Cognitive Computation,
School of Computer Science and Technology, Anhui University,
Hefei, China
zeyiabc@163.com

Yuhe Ding
Anhui Provincial Key Laboratory of Multimodal Cognitive Computation,
School of Computer Science and Technology, Anhui University,
Hefei, China
madao3c@foxmail.com

***Abstract*—Since deep neural networks can fully fit all data, including noisy labels, that is, mislabeled data, this will be detrimental to the robustness and generalization ability of the network. To address this problem, existing methods usually use small loss tricks to select clean samples for training. However, sample selection using only small loss techniques cannot distinguish between large loss "hard" samples and noise samples. In this work, we analyze the reasons why clean samples are misidentified as noise samples and propose a balanced selection mechanism based on loss variation. This method sorts the amount of loss change, uses variance to count the rankings of multiple epochs to describe the stability of loss variation, and selects samples with stable loss as clean samples to reduce the misclassification of "hard" samples. At the same time, semantic clustering is used to assist resampling and reweighting, thereby alleviating the negative impact of class imbalance on sample selection. We conduct comparative experiments and ablation experiments on synthetic noise datasets and real-world datasets such as CIFAR-10/100 and Clothing1M. Our VRLVMix has been shown to outperform numerous state-of-the-art methods, as evidenced by the experimental results. Moreover, it demonstrates the ability to extract clean samples even from large loss sample.**

Keywords-Noisy Label; Semi-supervised learning; Sample Selection

I. INTRODUCTION

Despite DNNs' outstanding generalization effectiveness in diverse tasks, they still face challenges in practical scenarios, primarily due to their heavy reliance on large-scale and precisely annotated training datasets. Unfortunately, obtaining such data is often difficult and expensive for many real-world tasks. Mining labeled large-scale data through alternative and cost-effective methods can lead to potentially inaccurate labels, introducing noisy labels [1]. Research [2, 3] indicates that Deep Neural Networks (DNNs) are susceptible to overfitting caused by noisy labels, which has a detrimental impact on model accuracy and generalization. Firstly, incorrect labels lead the model to learn erroneous associations and features, consequently reducing the accuracy of the model when applied to new data in practical scenarios. Secondly, the model overfits to the noisy labels present in the training set, resulting in an inability to generalize effectively to new data. As a consequence, the model's credibility and interpretability are limited, potentially constraining the application of artificial intelligence in certain domains. Lastly, mislabeled samples significantly increase the model's complexity and amplify the training difficulty. In light of these challenges, it becomes imperative to develop anti-noise algorithms that can effectively handle learning tasks with noisy labels.

Learning methods with noisy labels (LNL) primarily encompass two approaches: sample selection and label correction. Sample selection [4, 5, 6, 7] involves the careful curation of a set of clean samples from the overall sample pool to train the network. The commonly used standard is the small loss trick [5, 8], which is based on the idea that DNN tends to prioritize learning clean samples faster, resulting in clean samples often exhibiting small loss characteristics. On the other hand, label correction [9, 10, 11] usually requires estimating the noise through a random transition matrix or using the network to predict sample labels, so as to correct incorrect labels. However, in high noise situations, achieving accurate corrections can be challenging.

Currently, the most successful method for addressing the LNL problem is based on Semi-Supervised Learning methods [4, 12, 13] Unlike sample selection, this approach automatically divides the training set into clean and noisy subsets. Subsequently, this set of noisy samples is considered unlabeled, and in conjunction with the labeled clean samples, is used to jointly minimize the classification loss. By leveraging SSL techniques [14], this method effectively tackles the challenges posed by noisy labels and enhances the model's robustness and performance. However, within the labeled set, the partitioning strategy based on small loss fails to account for category

information, resulting in a class imbalance problem within the divided set. According to [13], enforcing mandatory class balancing is advantageous for network training and improves overall robustness. Nevertheless, in scenarios with uneven noise distribution, their adaptive approach of equalizing the sample count for each class might lead to further loss of valuable knowledge. We believe that their class balance method falls short of achieving optimal effectiveness in such situations.

We observe the selection results of small loss trick [12] and find that on some datasets the selected labeled set still contains a high proportion of noisy labels. For instance, when we synthesize a 40% noise rate and asymmetric noise in the CIFAR-100 dataset, modeling the loss distribution through GMM determines that approximately 40% of misclassified samples remain in the clean sample set. This presence of noise-labeled samples in the clean set negatively impacts the model's performance, leading to misclassifications of an equivalent number of clean samples. Therefore, the valuable knowledge contained in these misclassified clean samples is difficult to fully utilize.

In this paper, we consider the reasons why some samples are misclassified when the dataset is partitioned by a small loss trick and propose methods to mitigate the error. The use of GMM to partition the dataset [12] relies on the loss distribution of the entire training set, thus directly impacting the division results using the size of a single epoch loss. Consequently, factors influencing the loss size indirectly affect the accuracy of our dataset division: (1) The training results of a single epoch may not accurately reflect the overall learning trend. Under-learned "hard" clean samples are not selected due to large single epoch loss, resulting in less gain from the training process. (2) Accumulation of misclassifications due to model instability in the early stages of training and class imbalance problems during sample selection.

Overall, considering these factors is crucial for optimizing the dataset division process and ensuring robust and accurate training of the model. However, for the small loss strategy, the effect may not be significant due to the above-mentioned influencing factors during the training process, which blurs the decision boundary. The major contributions of this work are as follows:

- We propose to utilize loss variation as the sample selection criterion to alleviate the misclassification of large loss samples. At the same time, in order to avoid confirmation bias, we use the complementary advantages of small loss techniques and loss changes for sample selection.
- We use semantic clustering results to aid resampling of samples and use reweighting techniques to ensure relative balance among categories of selected sample sets. This further improves the accuracy of sample selection and the robustness of the network in high-noise environments.
- We conduct experiments on three datasets with different noise ratios to demonstrate the effectiveness of our method, even outperforming many state-of-the-art methods.

II. Related Work

In recent work, considerable research efforts have been devoted to noisy label training, encompassing three distinct groups of research methodologies.

A. Sample Selection Methods

The purpose of sample selection is to select potentially clean samples for training, thereby discarding noisy samples and fundamentally alleviating the impact of noisy labels. Coteaching [5] trains two networks simultaneously and selects samples through a small loss strategy in peer networks. SELFIE [6] leverages predictive uncertainty based on sample label prediction history to measure sample refurbishability, uses refurbishability and threshold comparison to screen samples. DivideMix [12] divides the set by modeling the loss using a two-distribution Gaussian Mixture Model (GMM), and then selects a clean sample. Jo-SRC [15] uses contrastive learning to estimate the "clean" or "outlier" probability of a sample. CNCLU [16] uses a sample selection method based on interval estimation to handle noisy labels and efficiently explore minority classes and large loss data. BARE [17] dynamically determines the selection threshold by analyzing the statistical properties of the loss values within the minibatch data.

B. Label Correction Methods

In order to correct the incorrect labels, several methods have been proposed. ALC [18] uses an active learning strategy to select and correct labels. M-correction [10] introduces a Beta mixture model for estimating and correcting noisy la-bels based on network predictions. PLC [19] progressively corrects labels and improves the model by considering the margin between the highest predicted confidence and the true label. MLC [20] employs a label rectification network as a meta-model to generate corrected labels for noisy labels, which are then used for training the main model.

C. Semi-Supervised Learning Methods

[4] proposes a two-stage semi-supervised learning method to cope with noisy labels. DivideMix [12], a typical algorithm among these works, conducts MixMatch [21] Semi-Supervised Learning on the separation results obtained by GMM to minimize the empirical victim risk of the model. Both UNICON [13] and ProMix [22] employ a uniform selection method to ensure a balanced distribution of selected samples from each category. However, the distinction lies in their approaches: UNICON [13] employs an adaptive approach to calculate the JS-divergence truncation value for determining the partition proportion, whereas ProMix [22] re-quires a manual specification of the noise rate. However, in cases where the dataset contains less noise or the distribution of noise

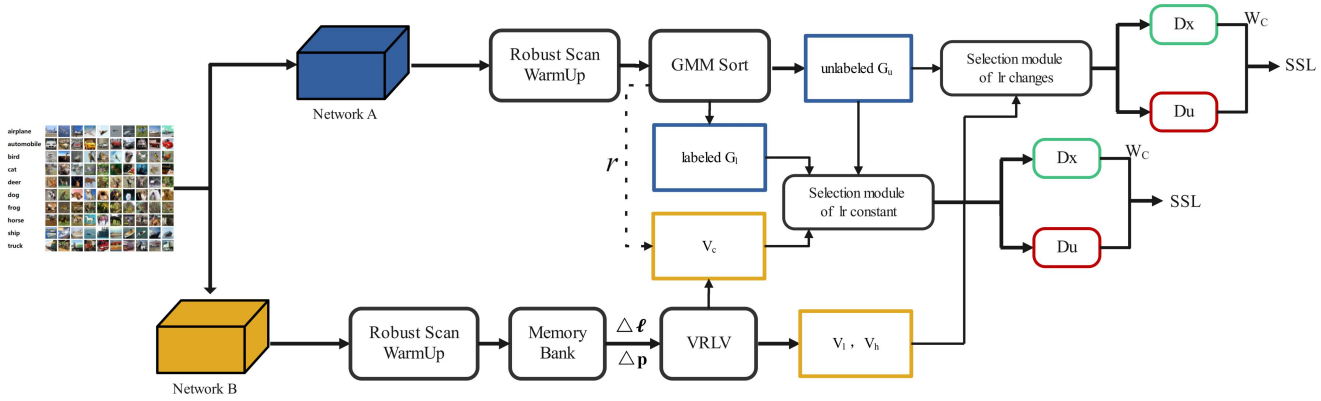

Figure 1. VRLVMix trains two networks simultaneously, referred to as A and B. During each epoch, Network A uses a two-distribution GMM to model the loss of the entire dataset to divide the two datasets G_l and G_u. Network B divides the dataset into V_l and V_c sets with stable loss changes and V_h sets with large loss changes by calculating the variance ranking of loss changes based on the losses and prediction results from the past period in MemoryBank. When the learning rate changes, we select balanced samples through the *selection module of lr changes*. When the learning rate is constant, we select balanced samples through the *selection module of lr constan*

sample categories is uneven, the selection scheme may be suboptimal. ScanMix [23] introduces unsupervised semantic clustering in semi-supervised learning, which im-proves the robustness of the model in the case of a high noise rate.

Not limited to the aforementioned three methods, there are also various robust deep learning techniques for handling noisy labels, such as designing robust loss functions [24, 25, 26, 27], estimating noise transfer matrices [28, 29, 30], and employing meta-learning [31, 32, 33, 34].

III. Background

Supervised Classification. Considering dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$. represents a training set of C classes, where x_i is the input image, y_i is the corresponding ground-truth label and satisfies the distribution $\mathcal{Y} = \{0,1\}^C$; N is the sample size of the training set. We instantiate the DNN model as $f(.;\theta)$ with parameters θ . For each input x_i , we minimize the cross-entropy loss:

$$\mathcal{L}_{ce} = \frac{1}{N}\sum_{i=1}^{N} l_{ce}(y_i, f(x_i)) = -\frac{1}{N}\sum_{i=1}^{N} y_i^T log(p_\theta(y_i \mid x_i))), \tag{1}$$

where $p_\theta(y_i \mid x_i) = argmax(softmax(f(x_i;\theta)))$.

Learning with Noisy Labels. However, when dealing with datasets containing noisy labels, certain labels may be incorrectly labeled. Let our dataset be represented as $\widetilde{D} = (x_i, \tilde{y}_i)_{i=1}^N$, with observable labels $\tilde{y}$ erroneously labeled with a probability of r , which we refer to as the noise rate. To address this, we train the entire noisy dataset on a DNN model. By using the softmax function, we obtain the conditional probability of each class and minimize the loss to achieve:

$$\mathcal{L}_{ce} = \frac{1}{N}\sum_{i=1}^{N} l_{ce}(\tilde{y}_i, f(x_i)) = -\frac{1}{N}\sum_{i=1}^{N} \tilde{y}_i^T log(p_\theta(\tilde{y}_i \mid x_i)), \tag{2}$$

The training process demonstrates a rapid convergence rate in its initial stages, driven by its ability to learn from “easy” samples. However, “hard” samples typically undergo a more gradual rate of loss reduction and are frequently memorized during the later epochs [2, 3]. Consequently, deep neural networks (DNNs) end up memorizing all noisy samples. This accumulation of memorization culminates in the phenomenon of overfitting during the later phases of training, ultimately leading to a degradation in performance.

IV. Proposed Method

Owing to the ambiguity of “hard” samples and the inherent randomness and instability of losses within a single epoch, these samples frequently manifest substantial loss values. It is difficult to distinguish whether it is a clean sample or not using only a single epoch loss, leading to incorrect classification and the inability to converge gradually to lower losses in subsequent epochs [35]. Therefore, we train two networks A and B simultaneously and proposed: Firstly, partition the dataset using the Variance Ranking of Loss Variation (VRLV) method as described in IV-A. Next, in IV-B, we introduce a Balancing Strategy involving resampling and weighting. We also employ Robust Warm-up and SSL techniques to train the network in IV-C. Our method overview is shown in Figure 1.

A. Partition by the VRLV

When partitioning the dataset D , Some methods usually use the small loss trick [12, 35, 23] to analyze the loss size of a single epoch and identify samples with higher loss values as

noise samples. However, there are some clean but challenging samples that also exhibit significant loss values, leading to mispartitioning. Once a sample is memorized, its corresponding loss will undergo a rapid reduction. In the event of a significant fluctuation in loss, it can be reasonably inferred that this sample is not a clean but challenging sample but rather a genuine noise sample.

We record the loss values in consecutive multiple epochs in the memory bank [37] to observe the change in the loss. Since the clean samples have the correct labels, their loss values will be relatively stable with less variation in loss values. Even if the loss drops sharply between adjacent epochs, it will not fluctuate frequently. However, noise samples are difficult to converge stably due to wrong labels, and their loss values vary greatly and fluctuate frequently. We use the difference between adjacent epoch losses to describe the degree of change, and sort according to the amount of change, and use the variance to count the sorting in τ epoch to describe the stability of the sample and the fluctuation of the loss:

$$\begin{aligned} &Var = \{var_i\}_{i=1}^{N}, \\ &var_i = \frac{1}{\tau}\sum_{t=t}^{t+\tau}(S_i^t - S_i^{mean})^2. \end{aligned} \tag{3}$$

where t represents epoch, $S_i^t = argsort(\ell_{t-1} - \ell_t)$ represents the ranking record of the loss changes of sample x_i between adjacent epochs, and $S_i^{mean} = \frac{1}{\tau}\sum_{t=t}^{t+\tau} S_i^t$. is the average ranking of each epoch during the τ epoch. By calculating the variance Var to describe the degree of loss variation for each sample, the aim is to distinguish between samples with frequent loss variations and stable learning, We divide the dataset into sets V_l, V_h, V_c of stable and fluctuating loss changes:

$$\begin{aligned} &V_l = \{x_i \mid var_i < \gamma_1\}, \\ &V_h = \{x_i \mid var_i > \gamma_2\}, \\ &V_c = \{x_i \mid x_i \in sort(Var) \times r\}. \end{aligned} \tag{4}$$

where r is the noise rate, which can be estimated by Gaussian Mixture Model, γ_1 and γ_2 are hyperparameters used to select stable learning and frequently changing samples, which can be manually set or calculated through noise rate. V_l is a sample set with a stable loss variation, and V_h is a sample set with a large loss variation. Similarly, we use the noise rate multiplied by the dataset size to select a certain proportion of clean samples V_c.

To better describe the loss variation and sample stability, we observed the variation trend of loss before and after the adjustment of the learning rate, as well as the model's predicted outcomes: at the same time as noise samples and "hard" samples experience significant fluctuations in loss, their prediction results also undergo corresponding changes. However, due to the difference in the ability of noise samples and clean samples to adapt to the network fitting state, noise samples often undergo continuous fluctuations or obtain new prediction results after the learning rate is reduced. On the contrary, the clean samples keep the prediction before the learning rate change after small fluctuations. We use β_i to denote the adaptability of x_i to the network:

$$\begin{aligned} \beta_i = \{&argmax(f(x_i;\theta))_{t1} \neq argmax(f(x_i;\theta))_{t2}\} \\ &\setminus \{argmax(f(x_i;\theta))_{t_*} \neq argmax(f(x_i;\theta))_{t_*+\tau}\} \end{aligned} \tag{5}$$

The former term refers to the change in prediction results, while the latter term describes the variation in prediction results when the learning rate changes (t_* epoch) and the model tends to stabilize ($t_* + \tau$ epoch) . We mark the samples that meet the above equation as $\beta_i = 1$. We interpret these samples as clean samples that adapt well to network changes. Consequently, we can identify a set of clean samples D_c based on these criteria.

$$D_c = \begin{Bmatrix} \{(x_i, y_i) \in (V_c \cap G_l) \mid \beta_i = 0\}, & if\ lr \text{ constant} \\ \{(x_i, y_i) \in (V_l \cap G_l \cup V_h) \mid \beta_i = 1\}, & if\ lr \text{ changes} \end{Bmatrix} \tag{6}$$

where G_l represents a labeled set divided by GMM. In the V_h set, when $\beta_i = 1$, both "hard" clean samples and noise samples exhibit significant loss fluctuations. However, "hard" clean samples show higher adaptability to changes in the network. In the V_l set, samples are less sensitive to changes in the learning rate. We identify samples with small losses and finally consistent predictions as clean samples. When $\beta_i = 0$ and the learning rate is constant, we select samples with loss variation stable and small loss as clean samples.

B. Resampling and Weighting

In section IV-A, we observed that selecting a more accurate set of clean samples can be achieved through the stability of loss variations. [13] selects the same lowest R partial samples for each class to resolve class imbalance due to sample selection. However, due to the uneven distribution of noise samples in different classes, this selection is suboptimal. Moreover, its adaptive partitioning results often select fewer labeled samples, and we hope to learn more valuable knowledge from clean samples. Therefore, we propose our balanced selection strategy. In addition to choosing a high-precision clean set, we utilize the results of GMM and the results of loss variation to perform mutual resampling between networks to alleviate category imbalance. This process helps fill the clean set, preventing underfitting caused by too few samples available for supervised learning.

We employ two different balanced sampling methods for the labeled set, one during periods of stable learning rate and the other when the learning rate undergoes changes. Specifically, when the learning rate is stable, we use GMM to divide the samples on network A, calculate the amount of loss change on network B, sort according to the amount of change, calculate the variance of multiple epochs of ranking, and use the obtained results to complement the labeled set partitioned by network A. In the supplementary process, we select samples in ascending order of variance. Additionally, we realize that the

unlabeled set of GMM contains a certain proportion of clean samples, which are usually not used in gradient optimization. The loss of these clean samples is still large, leading to fewer choices. So, when the pending selection set does not contain any samples of the "minority class" in order to better balance

Algorithm 1: VRLV-Mix

Input: training set $\mathcal{D}=\{x,y\}$, number of classes $\mathcal{C}$, number of epoch E, hyper-parameters $\gamma_1,\gamma_2,\tau,\mathrm{w}$

$f_\phi(x),\{\mathcal{N}_{x_i}\}_{i=1}^{|\mathcal{D}|}=PreTrain(\mathcal{D})$;
$f(.|\theta_1),f(.|\theta_2)=WarmUp(\mathcal{D},f_\phi(x))$;
for $i=\{1,2,...,|\mathcal{D}|\}$ **do**
  $f(x_i|\theta_1),f(x_i|\theta_2),\mathcal{L}_1,\mathcal{L}_2\rightarrow$MemoryBank;
$G_l=GMM(\mathcal{L}_1)$;
// Estimate the noise rate
$r=|G_l|/|\mathcal{D}|$;
while $epoch<E$ **do**
  $G_l,G_u=GMM(\mathcal{L}_1)$ and Update MemoryBank;
  $\Delta\mathcal{L}_2,p_i=f(x_i|\theta_2)\leftarrow MemoryBank$;
  // stable set V_l, unstable set V_h
  // small loss variation set V_c
  Determine V_l,V_h,V_c using (4) with $\Delta\mathcal{L}_2$;
  $\mathcal{R}=|\mathcal{D}|\times r/\mathcal{C}$;
  $D_{balance}=\{\}$;
  if *lr changes* **then**
    Determine β_i using (5) with p_i;
    $D_c=\{(x_i,y_i)\in(V_l\cap G_l\cup V_h)|\beta_i=1\}$;
    for $j=1$ *to* $\mathcal{C}$ **do**
      $D_o^j\leftarrow$ Lowest $\mathcal{R}-|D_c^j|$ portion of G_l^j;
      $D_{balance}\leftarrow D_{balance}\cup D_o^j$;
  else
    Determine β_i using (5) with p_i;
    $D_c=\{(x_i,y_i)\in(V_c\cap G_l)|\beta_i=0\}$;
    $\mathcal{R}_c=\{\mathcal{R}-|D_c^i|\},i=1,2,...,\mathcal{C}$;
    $P\leftarrow G_l\backslash D_c$ or $V_l\backslash D_c$;
    for $k=1$ *to* $\mathcal{C}$ **do**
      if $|P^k|>\mathcal{R}_c^k$ **then**
        // Data cleaning by $\mathcal{N}_{x_i}$
        $D_o^k\leftarrow ENN(P^k,\mathcal{N}_{x_i}|x_i\in D_c^k)$
      else
        // Data mixup by $\mathcal{N}_{x_i}$
        $D_o^k\leftarrow KNN(G_u,\mathcal{N}_{x_i}|x_i\in D_c^k)\cup P^k$
      $D_{balance}\leftarrow D_{balance}\cup D_o^k$
  $D_x=D_c\cup D_{balance}$;
  $D_u=D\backslash D_c$;
  $r>0.6$? Training with (11), (12) : Eq. (11)

each class and learn from the clean samples in the GMM unlabeled set, we use the semantic clustering results for sampling. We select samples from the GMM unlabeled set that satisfy the nearest neighbors of samples of this class in D_c, and apply mixup [38] interpolation on the selected samples as an oversampling target, up to the size of the selected sample set being equal to $|D_x|=N*r$:

$$D_x=D_c+D_{balance} \tag{7}$$

where, for the network B that uses the loss variation for sample selection, $D_{balance}$ is:

$$D_{balance}=|(G_l-D_c)|_{balance}\cup[KNN(G_u)] \tag{8}$$

For network A using GMM for sample selection, $D_{balance}$ is:

$$D_{balance}=|(V_c-D_c)|_{balance}\cup[KNN(V_c)] \tag{9}$$

Here, $|*|_{balance}$ refers to finding the balanced sample in the set, as denoted in (8), by selecting samples from G_l except D_c. We ensure to select the same number of samples for each category, and if any category y_i has insufficient samples, we supplement it using the K-Nearest Neighbors ($[KNN(G_u)]$) approach. This means that we add the mixup interpolation results of the Top-K nearest neighbor samples of $(x_i,y_i)\in D_c$ in the G_u set to the labeled set. Specifically, we calculate the number of additional samples $N_{over}=|D_{balance}/C|$ required for each category and perform data cleaning for categories with a sample size greater than N_{over} in the (G_l-D_c) set. For lower-ranked samples, our cleaning method follows the approach of Edited Nearest Neighbors (ENN). For categories with a sample size less than N_{over}, we select samples satisfying the K nearest neighbors from the GMM unlabeled set for interpolation and add them to $D_{balance}$.

When the learning rate changes, the clean sample set D_c we choose is obtained by (6), and most of the samples satisfy $\beta_i=1$ and come from the V_h set with large loss fluctuations. Through experimental observation, it can be seen that these samples usually have a large loss and will be misclassified as noise samples by GMM. To put it simply, most of the D_c sets we choose belong to the minority class samples in the marked set G_l selected by GMM, so we directly use the samples with small losses in each category in G_l for sampling until the total number of samples of each type reaches a balance, where for categories less than $|N*r|/C$, all samples are directly selected without additional supplements to avoid introducing noise samples. The detailed steps of our algorithm can be found in Algorithm 1, and the semi supervised training process is shown in the Figure 2.

Finally, we perform a simple weighting of the losses:

$$\mathcal{L}_{x'}=\frac{|D_x|}{|D_x^{\mathcal{C}}|\times\mathcal{C}}\mathcal{L}_x \tag{10}$$

C. SSL-Training

Using our sample selection method, we obtained partitioned datasets D_x and D_u, where D_x represents the identified clean set, and D_u contains samples more likely to be noise. The labels of D_u are removed, and it is utilized as an unlabeled set for joint learning with the labeled subset through Semi-

Supervised Learning. We conduct our experiments based on the MixMatch [21] approach used in [12]. Firstly, we perform label refinement on the labeled samples by linearly combining the ground truth y_b and the network's prediction result p_b. Additionally, a sharpening function is applied to reduce their temperature. Afterwards, we apply Mixup [38] to interpolate the samples. Finally, using MixMatch to train Mixup results, its semi-supervised loss is expressed as:

$$\mathcal{L}_{semi} = \mathcal{L}_{x'} + \lambda_u \mathcal{L}_u + \lambda_r \mathcal{L}_{reg} \tag{11}$$

Where $\mathcal{L}_{x'}$ is cross entropy loss, $\mathcal{L}_u$ is mean squared error, and $\mathcal{L}_{reg}$ is used to balance the regularization term of the classification output [39,10]. According to [23], in order to improve the robustness of the network at extremely high noise rates ($r > 0.6$), we also introduce the results of semantic clustering to assist in network training, and maximize the semantic clustering term through SCAN Loss [40]:

$$\mathcal{L}_{CLU} = \mathcal{L}_{\mathcal{N}} + \lambda_e \mathcal{L}_e \tag{12}$$

where, the term $\mathcal{L}_{\mathcal{N}}$ is used to optimize the similarity between the k-nearest neighbor sample set of x_i and x_j ($x_j \in \mathcal{N}_{x_i}$) with the same prediction result, making the model tend to produce similar outputs on similar samples. The term $\mathcal{L}_e$, determined by the λ_e weighting, predicts a uniform distribution on cluster $\mathcal{C}$.

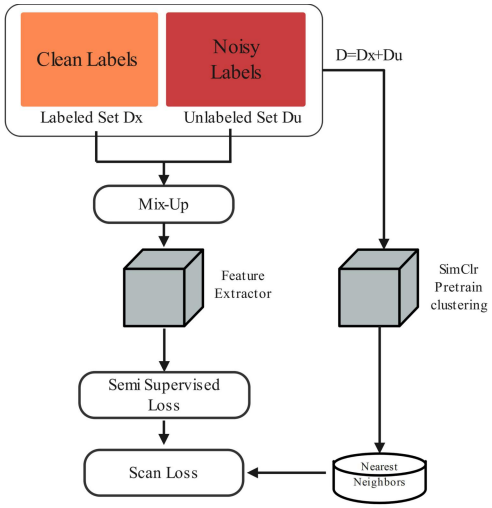

Figure 2. Semi supervised training process

V. EXPERIMENT

In this chapter, we present the results of our proposed algorithm on three datasets: CIFAR-10 [41], CIFAR-100, and Clothing1M [42]. We compare our method with state-of-the-art approaches in the field and validate its effectiveness. Detailed experimental setup information is provided, and we conduct comparative and ablation experiments analyses.

A. Experimental Setup

CIFAR-10 and CIFAR-100: The CIFAR-10 and CIFAR-100 datasets contain 50K training samples and 10K test samples, with image sizes of 32 × 32. CIFAR-10 contains 10 categories, while CIFAR-100 contains 100 categories. We use the noise Stochastic matrix method to synthesize noise in the dataset. The noise types are symmetric noise and asymmetric noise. Symmetric noise refers to the probability of randomly flipping labels into other classes being symmetric. Asymmetric noise is a more complex and realistic way of modeling noise, which simulates annotation errors with a fixed probability. For example, in CIFAR-10 [31] RUCK → AUTOMOBILE, BIRD → AIRPLANE, DEER → HORSE, CAT → DOG. In CIFAR-100, two subclasses are randomly switched within each superclass.

Clothing1M Dataset: A large-scale real-world dataset containing noisy labels. It contains 1 million clothing images from 14 categories, sourced from online shopping websites. The incorrect labeling was provided by the actual seller who posted the wrong label. The overall accuracy of the labels is 61.54% [43].

Experimental Details: We followed some previous work [12, 13, 22] and used the PreAct ResNet [44] architecture as our backbone on the CIFAR-10/100 dataset and trained it using SGD with a momentum of 0.9, a weight decay of 0.0005, and a batch size of 128. The network is trained for 300 epochs. We set the initial learning rate as 0.02 and reduce it by a factor of 10 after 150 epochs. The warm-up period is 10 epochs for CIFAR-10 and 30 epochs for CIFAR-100. For the Clothing1M dataset, we use the pre-trained ResNet-50 [44] on ImageNet. We choose an initial lr of 0.002 and a weight decay of 1e-3. The network is trained for 80 epochs. Additionally, to improve noise resistance under high noise rates, when the noise rate estimated by GMM is greater than 0.6, We adopt the standard settings of the semantic clustering pre-training network in [23], based on SimCLR [45], the SGD optimizer learning rate is 0.4, the decay rate is 0.1, the momentum is 0.9, the weight decay is 0.0001, the batch size is 512, and run for 500 epochs. For the pre-trained model mentioned above, we select K=20 nearest neighbors for each sample to form an ensemble. We then follow [42] with a batch size of 128 and $\lambda_E = 2$. We apply an SGD optimizer with a momentum of 0.9, a weight decay of 0.0005, and a learning rate of 0.001 for training.

B. Experimental Results

We use [12] as our baseline to compared with some recent Semi-Supervised Learning and other excellent works, such as ELR [46], MOIT [47], UNICON [13], ScanMix [23], JSmix [48], NCR [49], Sel-Cl [50], TCL [51]. And we obtained the results of earlier work from [13] articles, including PENCIL [52], JPL [53] etc.

CIFAR10 and CIFAR100: Table I. shows the average training results of our method on the CIFAR-10 and CIFAR-100 datasets with synthesized noise labels (Symmetric noise r=0.2, 0.5, 0.8, 0.9). It can be seen that our method generally achieves better testing accuracy on datasets with symmetric noise at different noise rates. Compared with DivideMix [12], it has achieved a comprehensive and significant improvement.

From the perspective of class balance affecting network learning and sample selection, UNICON [13] has selected all kinds of samples in a balanced way. We have achieved similar results on the CIFAR-10 dataset with only 10 categories, but UNICON performs slightly lower than DivideMix at a low noise rate (20%), probably because there are more clean samples available for supervised learning at low noise rates

TABLE I. TEST ACCURACIES (%) UNDER SYMMETRIC NOISE.

Method	CIFAR-10				CIFAR-100			
	20%	*50%*	*80%*	*90%*	*20%*	*50%*	*80%*	*90%*
LADMI [54]	88.3	81.2	43.7	36.9	58.8	51.8	27.9	13.7
DivideMix [12]	96.1	94.6	92.9	76.0	77.3	74.6	60.2	31.5
NCR+ [49]	95.2	94.3	91.6	75.1	76.6	72.5	58.0	30.8
ELR [46]	95.8	94.8	93.3	78.7	77.6	73.6	60.0	33.4
SeL-Cl+[50]	95.5	93.9	89.2	81.9	76.5	72.4	59.6	48.8
TCL [51]	95.0	93.9	92.5	89.4	78.0	73.3	65.0	54.5
MOIT [47]	94.1	91.1	75.8	70.1	75.9	70.1	51.4	24.5
UNICON [13]	96.0	95.6	93.9	90.8	78.9	77.6	63.9	44.8
ScanMix [23]	96.0	94.5	93.5	91.0	77.0	75.7	66.0	58.5
VRLVMix	**96.3**	**95.8**	**94.8**	**92.5**	**79.0**	**78.1**	**69.8**	**58.9**

($D_{clean} >> D_{noise}$). The former method of equally selecting each sample loses a large number of clean samples, and its comparative feature learning is not very effective. The difference is that we chose a labeled set with higher accuracy and relative balance, achieving better results than DivideMix without increasing additional comparison loss. And our method is more effective on the CIFAR-100 dataset with 100 categories. The labeled set we choose is closer to the real clean sample, which reduces the overfitting of the supervised learning process to the noise samples with wrong classification.

TABLE II. TEST ACCURACIES (%) UNDER ASYMMETRIC NOISE.

Method	CIFAR-10			CIFAR-100		
	10%	*30%*	*40%*	*10%*	*30%*	*40%*
SELF [6]	93.8	92.4	89.1	72.5	65.1	53.8
JPL [53]	94.2	92.5	90.7	72.0	68.1	59.5
DivideMix [12]	93.8	92.5	91.7	71.6	69.5	55.1
MIOT [47]	94.2	94.1	93.2	77.4	75.1	74.0
ELR[46]	95.4	94.7	92.0	77.3	74.6	73.2
JSMix [48]	95.2	92.8	90.7	77.7	74.6	74.3
Sel-Cl+ [50]	95.6	94.5	93.4	78.7	76.4	74.2
UNICON [13]	95.3	94.6	94.1	78.2	75.6	**74.8**
VRLVMix	**96.4**	**95.2**	**94.9**	**78.6**	**77.5**	73.0

We also conducted experiments in the presence of asymmetric noise ($r = 0.1, 0.3, 0.4$), and the experimental results are shown in Table II. Here, we did not train on the CIFAR-10 dataset using Scan-Loss [40] as this would corrupt our model and overfitting would occurred during training process. The overfitting situation is listed in the next section. But in CIFAR-100 ($r = 0.4$), we used SimCLR pretraining and Scan Loss [40] for auxiliary training when the early model prediction ability was poor. We believe that the semantic clustering method ensures the lower limit of our model, and this unsupervised clustering result is more helpful when it is difficult to distinguish between clean and noisy samples. Our accuracy has generally improved, but in more difficult situations ($r = 0.4$, CIFAR-100), our method is 1.8% lower than UNICON [13]. From the sample selection results, it may be because UNICON selected fewer labeled set samples of 25k, while our result of 30k introduced more noise labels. We did not present the results of ScanMix [23] here because its method only has a significant improvement in high noise rates. According to the results of their paper, the accuracy at CIFAR10-0.4 is 93.7%, and we achieve a higher accuracy of 1.2% compared to their result.

Clothing1M: We also tested our method on a real-world dataset that was mistakenly labeled. As shown in Table III, compared to the baseline DivideMix we chose, we achieved an improvement of 0.4%. Additionally, our results showed a 0.16% improvement compared to UNICON [13]. Due to the noise rate of the Clothing1M dataset being less than 0.4, we did not compare it with the ScanMix [23] method, as we believe its improvement is not significant at low noise rates and may even have negative effects compared to DivideMix [12]. Overall, the effectiveness of our method has been validated in real-world datasets.

TABLE III. EXPERIMENTAL RESULTS ON CLOTHING1M DATASET.

Method	Test Accuracy
GCE [26]	69.75
MLNT [31]	73.47
PENCIL [52]	73.49
JSMix [48]	74.15
JPL [53]	74.15
DivideMix[12]	74.76
ELR[46]	74.81
UNICON[13]	74.98
O2UMIX	**75.14**

C. Ablation Studys

In this section, we conduct ablation experiments on our method in different training environments to verify the effectiveness of each key component: (1) Utilizing Variance Ranking of Loss Changes; (2) Class-Balanced Sample Replenishment Strategy; (3) Combining with Semantic Clustering. We quantify the impact of each component on the overall performance of CIFAR-10/100 datasets with asymmetry levels of 40% and symmetry levels of 80% and 90%. We evaluate the performance by removing each

TABLE IV. ABLATION STUDY WITH DIFFERENT TRAINING SETTINGS. TEST ACCURACY (%) OF DIFFERENT SETTINGS ON CIFAR-10 AND CIFAR100 WITH DIFFERENT NOISE RATES (80% - 90% FOR SYM. AND 40% FOR ASYM.). SHOW THE BEST EPOCH, WHILE ALSO DISPLAYING THE PRECISION LAST EPOCH.

Dataset	**CIFAR-10**			**CIFAR-100**		
Noise Rate	***Asym 40%***	***Sym 80%***	***Sym 90%***	***Asym 40%***	***Sym 80%***	***Sym 90%***
Method	***Best***	***Last***	***Best***	***Last***	***Best***	***Last***
VRLVMix w/o VRLV	92.74	92.49	92.39	92.23	91.82	91.80
VRLVMix w/o Balancing	93.67	93.59	92.92	92.74	91.59	91.38
V-Mix w/o Balanceing+Scan	93.83	93.55	92.40	92.38	88.72	88.39
VRLVMix w/o Scan	**94.97**	**94.80**	93.51	92.89	91.61	91.47
VRLVMix	94.90	94.68	**94.83**	**94.76**	**92.51**	**92.46**

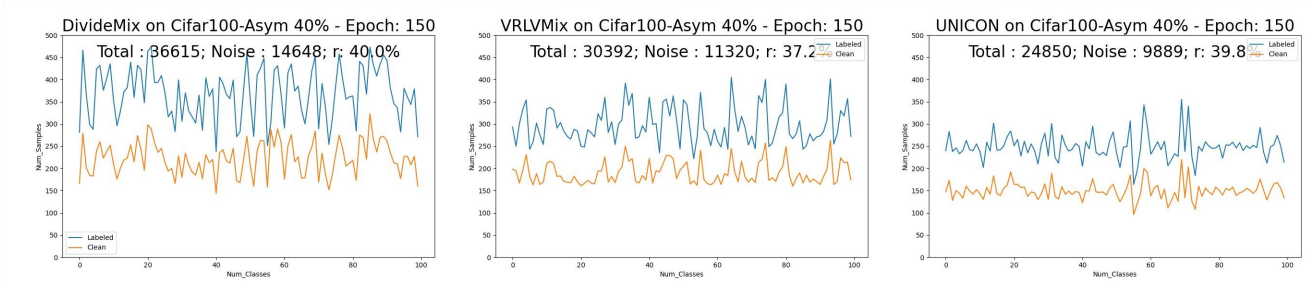

Figure 3. The left figure shows the results of small loss selection using GMM at epoch 150. The middle figure displays the selection results of our method at epoch 150. The figure on the right shows the results of UNICON's dynamic uniform selection through JS divergence.

component during training. Each of these experiments was performed in triplicate, and we list the best precision and last precision obtained from these experiments in Table IV.

First, we train on the entire training dataset without utilizing the degree of loss variation for sample selection. Both network A and network B model the loss distribution through GMM. However, to maintain sample balance between categories, we still perform ENN cleaning for the "large number" category and oversampling for the "small number" category. Additionally, we use the "k=20" nearest neighbor results from semantic clustering for auxiliary sampling. We observed a decline in accuracy, especially on the CIFAR-100 dataset, where the high noise rate limits the availability of clean sample information. Relying solely on the small loss trick to obtain more noise samples in the labeled set hampers network training. Thus, selecting a cleaner sample set to increase valuable information and reduce noise disturbance becomes crucial. Nevertheless, with the support of the Scan semantic clustering method, the network maintains relatively good accuracy.

Next, we eliminate the balancing strategy. Specifically, after obtaining a cleaner set D_c, which contains a relatively small number of samples, we need to perform over sampling on this labeled set. For network A, we will utilize the remaining portion of the labeled set V_c, which is divided based on loss variation, excluding D_c, as a supplementary labeled dataset. As for network B, we will use the remaining portion of the labeled set G_l, which is divided using GMM and does not include D_c, as a supplementary labeled dataset. We observed a more significant reduction in accuracy, especially on CIFAR-100 with more categories and asymmetric noise. These results indicate that the class balancing strategy becomes more critical in cases with more complex and challenging noise patterns. At the same time, since the semantic clustering Scan improves the lower limit of our network at high noise rates, we remove both the semantic clustering and balancing strategies simultaneously to observe the impact of the balancing strategy alone, resulting in a 13.9% drop on CIFAR100 Sym90% and a 23.9% drop on Asym40%.

Finally, we evaluated the effect of Scan on our network and concluded that Scan complements our network well at higher noise rates. However, at low noise rates (CIFAR-10 Asym40%), it negatively impacts our model, leading to a 0.1% accuracy drop. In contrast, for CIFAR-100 with 80% and 90% noise rates, it brings significant improvement.

D. Analysis of partition results

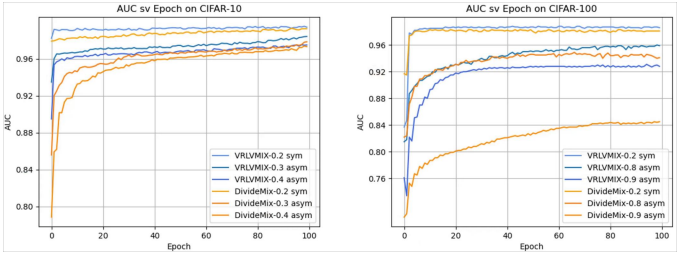

Figure 4. AUC for clean/noisy image classification on CIFAR-10 and CIFAR-100 training samples. As training increases our selection results become more accurate.

Precision performance: First, we analyze our method from the accuracy of sample selection. In Figure 4, we show the

Area Under a Curve for sample selection of the first 100 epochs on the CIFAR-10 and CIFAR-100 datasets, respectively. The blue curves represent our method, and the yellow curves represent methods using the small loss trick [12]. It can be observed that by pre-training the network through semantic clustering, our method can reduce the instability in the early stage of model training, and the score is better than [12] while steadily improving.

Balance performance: Our method builds on DivideMix [12] and utilizes loss variation to assist in dividing the dataset. As can be seen from the left side of Figure 3, our method alleviates the imbalance in the number of samples between categories to a certain extent and simultaneously increases the proportion of clean samples in the labeled set. Specifically, our division results contain 37.2% noise samples, which is 3328 fewer noise samples than DivideMix in the same epoch. Additionally, we identify a certain proportion of clean samples from the GMM unlabeled set. These samples would not be selected if we solely relied on small loss trick for selection, as they are prone to misclassification due to error accumulation.

In Table II, our method is 1.8% less accurate than UNICON [13] on CIFAR100-Asym40%. By observing the dataset division results, as shown on the right side of Figure IV, UNICON [13] selects a relatively uniform number of samples for each category, but it selects fewer training samples (25000 < 30000) overall. Although our method increases the proportion of clean samples in the labeled set, there are still more noisy samples in the divided labeled set, which we believe is one of the influencing factors. Secondly, we think that class uniformity may be more consistent with the distribution of this synthetic noise dataset, and the impact of class uniformity may be more significant on noisy datasets that are difficult to divide. We do not replace the sampling method in IV-B to make all kinds of completely balanced to test whether the experimental results are improved because our starting point is still to reduce the misclassification caused by the small loss strategy and to learn more valuable knowledge from clean samples.

VI. Conclusion

In this work, we analyze the factors that affect the differentiation between noise samples and "hard" clean samples, including the complexity and unclear features of the "hard" clean samples themselves. Due to the learning ability of the samples at fixed stages, the inability of individual epoch results to reflect the overall trend, and class imbalance leading to error accumulation, these "hard" samples are difficult to identify. We propose a complementary sample selection method that utilizes loss variation and small loss strategies, which has been proven to produce a more accurate sample set. Additionally, we mitigating the issue of class imbalance from the root by employing resampling and reweighting methods. Finally, we leverage the results of semantic clustering to improve the robustness of our model under high noise rates. We demonstrate the effectiveness of our method across various datasets.

Acknowledgment

This work was supported in part by the Anhui Provincial Natural Science Foundation under Grant 2108085MF211, in part by the Anhui Provincial Natural Science Foundation under Grant 2308085QF221, in part by the University Synergy Innovation Program of Anhui Province under Grant GXXT-2022-014. (Corresponding author: Zhengzheng Tu.)

References

[1] Benoıt Fr´enay and Michel Verleysen. "Classification in the presence of label noise: a survey". In: IEEE transactions on neural networks and learning systems 25.5 (2013), pp. 845–869.

[2] Devansh Arpit et al. "A closer look at memorization in deep networks". In: International conference on machine learning. PMLR. 2017, pp. 233–242.

[3] Chiyuan Zhang et al. "Understanding deep learning (still) requires rethinking generalization". In: Communications of the ACM 64.3 (2021), pp. 107–115.

[4] Yifan Ding et al. "A semi-supervised two-stage approach to learning from noisy labels". In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE. 2018, pp. 1215–1224.

[5] Bo Han et al. "Co-teaching: Robust training of deep neural networks with extremely noisy labels". In: Advances in neural information processing systems 31 (2018).

[6] Duc Tam Nguyen et al. "Self: Learning to filter noisy labels with self-ensembling". In: arXiv preprint arXiv:1910.01842 (2019).

[7] Hongxin Wei et al. "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. 2020, pp. 13726–13735.

[8] Xingrui Yu et al. "How does disagreement help generalization against label corruption?" In: International Conference on Machine Learning. PMLR. 2019, pp. 7164–7173.

[9] Scott Reed et al. "Training deep neural networks on noisy labels with bootstrapping". In: arXiv preprint arXiv:1412.6596 (2014).

[10] Eric Arazo et al. "Unsupervised label noise modeling and loss correction". In: International conference on machine learning. PMLR. 2019, pp. 312–321.

[11] Yangdi Lu and Wenbo He. "SELC: self-ensemble label correction improves learning with noisy labels". In: arXiv preprint arXiv:2205.01156 (2022).

[12] Junnan Li, Richard Socher, and Steven CH Hoi. "Dividemix: Learning with noisy labels as semi-supervised learning". In: arXiv preprint arXiv:2002.07394 (2020).

[13] Nazmul Karim et al. "Unicon: Combating label noise through uniform selection and contrastive learning". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, pp. 9676–9686.

[14] Yassine Ouali, C´eline Hudelot, and Myriam Tami. "An overview of deep semi-supervised learning". In: arXiv preprint arXiv:2006.05278 (2020).

[15] Yazhou Yao et al. "Jo-src: A contrastive approach for combating noisy labels". In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, pp. 5192–5201.

[16] Xiaobo Xia et al. "Sample selection with uncertainty of losses for learning with noisy labels". In: arXiv preprint arXiv:2106.00445 (2021).

[17] Deep Patel and PS Sastry. "Adaptive sample selection for robust learning under label noise". In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2023, pp. 3932–3942.

[18] Jan Kremer, Fei Sha, and Christian Igel. "Robust active label correction". In: International conference on artificial intelligence and statistics. PMLR. 2018, pp. 308–316.

[19] Yikai Zhang et al. "Learning with feature-dependent label noise: A progressive approach". In: arXiv preprint arXiv:2103.07756 (2021).

[20] Guoqing Zheng, Ahmed Hassan Awadallah, and Susan Dumais. "Meta label correction for noisy label learning". In: Proceedings of the AAAI

Conference on Artificial Intelligence. Vol. 35. 12. 2021, pp. 11053–11061.

[21] David Berthelot et al. “Mixmatch: A holistic approach to semi-supervised learning”. In: Advances in neural information processing systems 32 (2019).

[22] Haobo Wang et al. “ProMix: combating label noise via maximizing clean sample utility”. In: arXiv preprint arXiv:2207.10276 (2022).

[23] Ragav Sachdeva et al. “ScanMix: learning from severe label noise via semantic clustering and semi-supervised learning”. In: Pattern Recognition 134 (2023), p. 109121.

[24] Aritra Ghosh, Himanshu Kumar, and P Shanti Sastry. “Robust loss functions under label noise for deep neural networks”. In: Proceedings of the AAAI conference on artificial intelligence. Vol. 31. 1. 2017.

[25] Zhilu Zhang and Mert Sabuncu. “Generalized cross entropy loss for training deep neural networks with noisy labels”. In: Advances in neural information processing systems 31 (2018).

[26] Yisen Wang et al. “Symmetric cross entropy for robust learning with noisy labels”. In: Proceedings of the IEEE/CVF international conference on computer vision. 2019, pp. 322–330.

[27] Lei Feng et al. “Can cross entropy loss be robust to label noise?” In: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence. 2021, pp. 2206–2212.

[28] Giorgio Patrini et al. “Making deep neural networks robust to label noise: A loss correction approach”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, pp. 1944–1952.

[29] Dan Hendrycks et al. “Using trusted data to train deep networks on labels corrupted by severe noise”. In: Advances in neural information processing systems 31 (2018).

[30] Zhuolin Jiang et al. “Learning from noisy labels with noise modeling network”. In: arXiv preprint arXiv:2005.00596 (2020).

[31] Junnan Li et al. “Learning to learn from noisy labeled data”. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, pp. 5051–5059.

[32] Jun Shu et al. “Meta-weight-net: Learning an explicit mapping for sample weighting”. In: Advances in neural information processing systems 32 (2019).

[33] Zizhao Zhang et al. “Distilling effective supervision from severe label noise”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, pp. 9294–9303.

[34] Zhen Wang, Guosheng Hu, and Qinghua Hu. “Training noise-robust deep neural networks via meta-learning”. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020, pp. 4524–4533.

[35] Sheng Guo et al. “Curriculumnet: Weakly supervised learning from large-scale web images”. In: Proceedings of the European conference on computer vision (ECCV). 2018, pp. 135–150.

[36] Lu Jiang et al. “Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels”. In: International conference on machine learning. PMLR. 2018, pp. 2304–2313.

[37] Jinchi Huang et al. “O2u-net: A simple noisy label detection approach for deep neural networks”. In: Proceedings of the IEEE/CVF international conference on computer vision. 2019, pp. 3326–3334.

[38] Hongyi Zhang et al. “mixup: Beyond empirical risk minimization”. In: arXiv preprint arXiv:1710.09412 (2017).

[39] Daiki Tanaka et al. “Joint optimization framework for learning with noisy labels”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, pp. 5552–5560.

[40] Wouter Van Gansbeke et al. “Scan: Learning to classify images without labels”. In: European conference on computer vision. Springer. 2020, pp. 268–285.

[41] Alex Krizhevsky, Geoffrey Hinton, et al. “Learning multiple layers of features from tiny images”. In: (2009).

[42] Tong Xiao et al. “Learning from massive noisy labeled data for image classification”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, pp. 2691–2699.

[43] Xiaobo Wang et al. “Co-mining: Deep face recognition with noisy labels”. In: Proceedings of the IEEE/CVF international conference on computer vision. 2019, pp. 9358–9367.

[44] Kaiming He et al. “Deep residual learning for image recognition”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 770–778.

[45] Ting Chen et al. “A simple framework for contrastive learning of visual representations”. In: International conference on machine learning. PMLR. 2020, pp. 1597–1607.

[46] Sheng Liu et al. “Early-learning regularization prevents memorization of noisy labels”. In: Advances in neural information processing systems 33 (2020), pp. 20331–20342.

[47] Diego Ortego et al. “Multi-objective interpolation training for robustness to label noise”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, pp. 6606–6615.

[48] Zhijie Wen, Hui Xu, and Shihui Ying. “JSMix: a holistic algorithm for learning with label noise”. In: Neural Computing and Applications 35.2 (2023), pp. 1519–1533.

[49] Ahmet Iscen et al. “Learning with neighbor consistency for noisy labels”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, pp. 4672–4681.

[50] Shikun Li et al. “Selective-supervised contrastive learning with noisy labels”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, pp. 316–325.

[51] Zhizhong Huang, Junping Zhang, and Hongming Shan. “Twin contrastive learning with noisy labels”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, pp. 11661–11670.

[52] Kun Yi and Jianxin Wu. “Probabilistic end-to-end noise correction for learning with noisy labels”. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, pp. 7017–7025.

[53] Youngdong Kim et al. “Joint negative and positive learning for noisy labels”. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, pp. 9442–9451.

[54] Yilun Xu et al. “L_dmi: An information-theoretic noise-robust loss function”. In: arXiv preprint arXiv:1909.03388 (2019).