Neighborhood Collective Estimation for Noisy Label Identification and Correction

Jichang Li[1,2][0000−0001−5778−2232], Guanbin Li[1*][0000−0002−4805−0926], Feng Liu[3][0000−0002−4811−7828], and Yizhou Yu[2*][0000−0002−0470−5548]

[1] Sun Yat-sen University, Guangzhou 510006, China
[2] The University of Hong Kong, Hong Kong
[3] Deepwise AI Lab, Beijing, China
csjcli@connect.hku.hk, liguanbin@mail.sysu.edu.cn, liufeng@deepwise.com, yizhouy@acm.org

Abstract. Learning with noisy labels (LNL) aims at designing strategies to improve model performance and generalization by mitigating the effects of model overfitting to noisy labels. The key success of LNL lies in identifying as many clean samples as possible from massive noisy data, while rectifying the wrongly assigned noisy labels. Recent advances employ the predicted label distributions of individual samples to perform noise verification and noisy label correction, easily giving rise to confirmation bias. To mitigate this issue, we propose Neighborhood Collective Estimation, in which the predictive reliability of a candidate sample is re-estimated by contrasting it against its feature-space nearest neighbors. Specifically, our method is divided into two steps: 1) Neighborhood Collective Noise Verification to separate all training samples into a clean or noisy subset, 2) Neighborhood Collective Label Correction to relabel noisy samples, and then auxiliary techniques are used to assist further model optimization. Extensive experiments on four commonly used benchmark datasets, i.e., CIFAR-10, CIFAR-100, Clothing-1M and Webvision-1.0, demonstrate that our proposed method considerably outperforms state-of-the-art methods.

Keywords: Learning with noisy labels, Neighborhood collective estimation, Confirmation bias.

1 Introduction

Deep neural networks (DNNs) have achieved significant success in computer vision tasks, such as image classification [41,1,22,5,18,52], *etc.* However, they rely heavily on tremendous quantities of high-quality manual annotations. To alleviate the need for extensive human annotations while improving the generalization capability of deep neural networks, learning with noisy labels (LNL) has been proposed to effectively leverage large-scale yet poorly-annotated datasets while mitigating the effects of model overfitting to noisy labels.

*Corresponding Authors are Guanbin Li and Yizhou Yu.

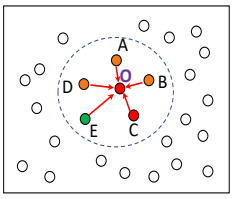


Fig. 1. An illustration to exemplify our basic idea. Samples distributed within the dotted circle, including the candidate sample, Point O, and its nearest neighbors, *i.e.*, Point A, B, C, D and E are close to each other in the feature-space neighborhood. Different colors indicate different labels (either predicted label or given groundtruth label). In the noise verification stage, a given label of the candidate (Point O) is considered noisy if there is a huge inconsistency between the label distributions of the candidate and its nearest neighbors; and otherwise, the candidate is considered as a clean sample. Likewise, in the noise correction stage, a noisy sample discards the given noisy label and is relabeled through a neighborhood collective estimation process involving its contrastive neighbors

To tackle the challenges imposed by LNL, previous works have proposed massive strategies [10,39,19,32,47], including noisy label correction [3,24], noisy label or sample rejection[19,47,15,14], and noisy sample reweighing [42,35,12]. The mainstream pipeline first uses noise verification strategies to separate the original training set into a clean set and a noisy set, which contain training samples with clean labels and noisy labels respectively, in order to diminish the effect of noisy labels during model training. Then, (un)supervised learning or semi-supervised learning (SSL) based techniques are adopted to correct noisy labels and further optimize the classification model by regarding the clean set and noisy set as labeled and unlabeled samples respectively. In this scheme, original noisy labels are simply discarded for their high chances to be incorrect, avoiding the negative effect of noisy label memorization in the trained model.

In the context of learning with noisy labels, there may exist classes with imbalanced noisy or clean samples, especially in real-world noisy datasets such as Clothing-1M [45] and Webvision-1.0 [23]. For instance, there might be a relatively high proportion of noisy labels in some hard-to-annotate classes; on the other hand, a trained model may produce low-confident predictions on a relatively high proportion of hard-to-learn clean samples in some classes, making existing noise identification algorithms incorrectly identify them as noisy samples. As a result, noise accumulation may take place implicitly in such classes, making the trained model produce unreliable label predictions. The above scenarios could make an LNL algorithm fall into the so-called confirmation bias [40,2], which causes the algorithm to favor incorrect training labels that have been confirmed with predicted labels in earlier training iterations. In this context, relying too much on the potentially biased label predictions for individual training samples would increase the risk of incorrectly identifying noisy labels in the noise verification stage. Moreover, confirmation bias also exists in the subsequent noise correction stage, where SSL or other methods, such as label-guessing [19,30,51] and label

re-assignment [47], construct pseudo-labels for unlabeled samples in the noisy set using potentially biased label predictions. Apparently, model training in the optimization stage would strengthen this bias as more confident but incorrect predictions would defy new changes, and subsequently even deteriorate model performance in high noise ratio scenarios.

We are inspired by the premise of contrastive learning that samples from the same class should have higher similarity in the feature space than those from different classes [29,31,9]. Therefore, we approach learning with noisy labels from a different perspective and propose Neighborhood Collective Estimation (NCE), in which we re-estimate the predictive reliability of a candidate sample by contrasting it against its feature-space nearest neighboring samples. Herein, we borrow the concept from contrastive learning, and then name such neighboring samples of the candidate as contrastive neighbors. Leveraging contrastive neighbors enriches the predictive information associated with the candidate and also makes such information relatively unbiased, thereby improving the accuracy of noisy label identification and correction. Fig. 1 displays the basic idea of the proposed method.

Specifically, to abide by the mainstream LNL pipeline, we divide our method into two steps: 1) Neighborhood Collective Noise Verification (NCNV) to separate all training samples into a clean set and a noisy set, 2) Neighborhood Collective Label Correction (NCLC) to relabel noisy samples. In the NCNV stage, a candidate sample is considered noisy when there is a huge inconsistency between the one-hot vector of the given label of the candidate and the label distributions of its contrastive neighbors predicted using the trained model. In the NCLC stage, we only relabel noisy samples whose predicted label distribution is sufficiently similar to the given labels of neighboring clean samples, and the corrected label of a noisy sample is related to a weighted combination of the given labels of neighboring clean samples. Once we have identified clean samples and relabeled noisy ones, we leverage off-the-shelf and well-established techniques, such as mixup regularization [50] and consistency regularization [36], to perform further SSL-based model training.

In summary, the main contributions are as follows.

- We propose Neighborhood Collective Estimation for learning with noisy labels, which leverages contrastive neighbors to obtain richer and relatively unbiased predictive information for candidate samples and thus mitigates confirmation bias.
- Concretely, we design two steps called Neighborhood Collective Noise Verification and Neighborhood Collective Label Correction to identify clean samples and relabel noisy ones respectively.
- We evaluate our method on four widely used LNL benchmark datasets, *i.e.*, CIFAR-10 [16], CIFAR-100 [16], Clothing-1M [45] and Webvision-1.0 [23], and the results demonstrate that our proposed method considerably outperforms state-of-the-art LNL methods.

2 Related Work

In this section, we focus on noise verification and label correction that are means involved in current dominant pipeline to address the LNL problem.

2.1 Noise Verification

Noise verification involves sample selection to choose and remove noisy labels within the training datasets. Proper noise verification strategies are necessary and several earlier works [10,48,15] have shown that samples with smaller cross-entropy loss are prone to hold clean labels, assuming that deep neural networks prefer to memorize simple patterns first rather than overfit to noisy labels. Also, some recently superior methods made efforts to model per-sample loss distributions with Beta Mixture Models (BMM) [26] or Gaussian Mixture Models (GMM) [34] to separate noisy labels from all the training samples [3,19,30,51,13,46]. However, based on the predicted label distributions of individual candidate samples to identify the training samples, the above-stated noise verification strategies tend to fall into confirmation bias. Previous works have also attempted to identify noisy labels by leveraging neighborhood information. They either use neighborhood samples to remove noisy labels or re-weight them [43,4,32,53,44]. For example, Bahri *et al.* [4] proposed to identify noisy label by searching nearest neighbors based on the model predictions of a KNN classifier, while Zhu *et al.* [53] uses feature-space neighbors to help estimate a noise transition matrix. In our work, we employ neighborhood collective estimation to realize both the identification and correction of noise labels, and make the two promote each other, to achieve better noise label learning.

2.2 Label Correction

To alleviate the effect of noisy memorization, noisy labels are discarded simply, and then label correction is adopted to relabel unlabeled samples [25,37,19,30,47,51]. This aims to give reliable pseudo-labels and support subsequent model training so as to achieve better performance. For example, "SELFIE" proposed by Song *et al.* [37] tried to perform label correction by considering model predictions from past selecting clean labels. Also, Li *et al.* [19] "co-guessed" pseudo-labels for unlabeled (noisy) samples via ensembling predictions of coupled networks, while Yao *et al.* [47] employed label re-assignment to provide pseudo-labels with the predictions of a temporally averaged model. Different from those as mentioned above, we correct noisy labels with the aid of neighboring labeled samples. This can relatively avoid confirmation bias that derives from model predictions at individual samples.

3 The Proposed Method

Problem formulation. Learning with noisy labels seeks an optimal model trained with a large-scale noisy dataset $\mathcal{D}_{\mathbf{train}} = \{(x_i, y_i)\}_{i=1}^{N}$, where N is the

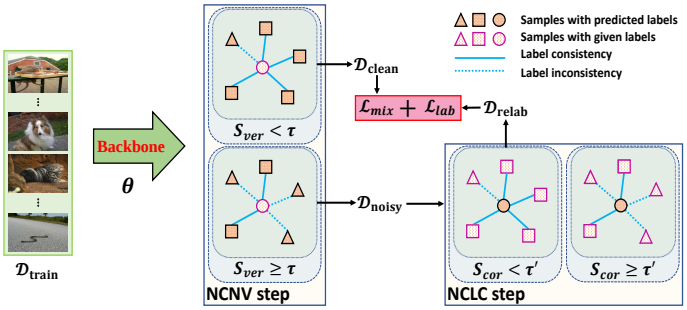


Fig. 2. Our proposed steps for learning with noisy labels. Triangles and squares represent contrastive neighbors from two different classes while circles denote the candidate samples in various steps. We assume the candidates belong to the class represented by the squares. In this work, we design two steps called Neighborhood Collective Noise Verification (NCNV) and Neighborhood Collective Label Correction (NCLC) to identify clean samples and relabel noisy ones respectively. Both steps leverage contrastive neighbors to obtain richer and relatively unbiased predictive information for candidate samples and thus mitigate confirmation bias

number of sample-label pairs and each pair consists of a training sample x_i and its associated label y_i over C classes while whether the given label is noisy or clean is unknown. During the training process, a sample is fed into a model being trained, that is parameterized by θ and contains a feature extractor Φ and a classifier with a softmax layer, to obtain its corresponding feature representation $\Phi(x_i)$ and class probabilities $p(y|x_i)$ respectively.

Contrastive neighbors. We contrast a candidate sample against its feature-space nearest neighbors to enrich and diversify predictive information of the candidate. Such nearest neighbors are called contrastive neighbors in this paper. First, to compute feature similarity between a candidate sample x_i and one of its feature-space neighbors x_j, we define a similarity function:

$$d(x_i, x_j) = \frac{\Phi(x_i)^\top \Phi(x_j)}{\|\Phi(x_i)\| \|\Phi(x_j)\|}, \tag{1}$$

where $d(\cdot, \cdot)$ denotes the cosine distance metric. Then, we set up a pairwise connection between the two samples and quantify the discrepancy between their label distributions through the Jensen-Shannon (JS) divergence as follows,

$$J(p_i, p_j) = \frac{1}{2} KL(p_i || \frac{p_i + p_j}{2}) + \frac{1}{2} KL(p_j || \frac{p_i + p_j}{2}), \tag{2}$$

where $KL(\cdot||\cdot)$ represents the Kullback-Leibler (KL) divergence, and for sample x_i (or x_j), in different contexts, p_i (or p_j) represents either its probabilistic label distribution predicted using a trained model or its given ground-truth label. $J(\cdot, \cdot)$ returns values in the range of [0,1], and the use of JS divergence allows us to measure the discrepancy between the probabilistic label distributions of different samples. $J(p_i, p_j) \to 0$ indicates that the label distributions of p_i and p_j are very similar while $J(p_i, p_j) \to 1$ means the label distributions of these two samples are of great difference.

Overview. In this paper, we propose Neighborhood Collective Estimation (NCE) to tackle learning with noisy labels. In detail, we first propose Neighborhood Collective Noise Verification (NCNV) to identify noisy labels in $\mathcal{D}_{\mathbf{train}}$ and divide $\mathcal{D}_{\mathbf{train}}$ into clean subset $\mathcal{D}_{\mathbf{clean}}$ and noisy subset $\mathcal{D}_{\mathbf{noisy}}$. Then, we propose Neighborhood Collective Label Correction (NCLC) to relabel selected samples from $\mathcal{D}_{\mathbf{noisy}}$ and form a new subset $\mathcal{D}_{\mathbf{relab}}$. Finally, we leverage auxiliary techniques to perform model fine-tuning so as to further optimize our model. The diagram and the training procedure of our proposed model have been summarized in Fig. 2 and Algorithm 1, respectively.

Algorithm 1: Learning with Noisy Labels based on Neighborhood Collective Estimation

Input: Dataset $\mathcal{D}_{\mathbf{train}}$; Number of training epochs T_{tr}; Number of warm-up epochs T_{wu}; Learning rate η

Output: Optimal model parameter θ

for $t \to 1 \cdots T_{tr}$ **do**
  if $t < T_{wu}$ **then**
    /* The warm-up step. */
    WarmUp($\mathcal{D}_{\mathbf{train}}$; θ). // Initialize the model with a "WarmUp" function.
  else
    /* The NCNV step. */
    Use Eq. (5) to split $\mathcal{D}_{\mathbf{train}}$ into clean samples $\mathcal{D}_{\mathbf{clean}}$ and noisy ones $\mathcal{D}_{\mathbf{noisy}}$.
    /* The NCLC step. */
    Use Eq. (9) to relabel a subset of samples from $\mathcal{D}_{\mathbf{noisy}}$ and form a new subset $\mathcal{D}_{\mathbf{relab}}$.
    /* The model fine-tuning step. */
    Randomly sample mini-batches from $\mathcal{D}_{\mathbf{clean}}$ and $\mathcal{D}_{\mathbf{relab}}$.
    Update model parameter θ by applying SGD with η to Eq. (13).

3.1 Neighborhood Collective Noise Verification

In an effort to identify label noise for the task of LNL, most recent research establish sample selection criteria on the basis of predicted label distributions of individual samples [10,48,15,3,19], thus it is hard for them to avoid confirmation bias. Aiming at mitigating such bias, we formulate a novel noise verification function that determines whether a candidate is a noisy sample or not through the estimation of its label inconsistency score, which measures the degree of inconsistency between the label distributions of the candidate sample and its contrastive neighbors. Specifically, given a candidate sample-label pair $\left(x^{(c)}, y^{(c)}\right) \in \mathcal{D}_{\mathbf{train}}$, we first find its K nearest neighbors in the feature space using the cosine similarity in Eq. (1) and then declare them as contrastive neighbors, as formulated below.

$$\{x_k^{(c)}\}, k = 1, \cdots, K \leftarrow \mathbf{KNN}(x^{(c)}; \mathcal{D}_{\mathbf{train}}; K), \tag{3}$$

where $\mathbf{KNN}(x^{(c)}; \mathcal{D}_{\mathbf{train}}; K)$ is a function that returns K most similar samples in $\mathcal{D}_{\mathbf{train}}$ for the candidate sample $x^{(c)}$. Note that $x^{(c)}$ is temporarily removed from $\mathcal{D}_{\mathbf{train}}$ at this moment.

Then, the neighborhood-based label inconsistency score for the given label of the candidate can be defined as follows,

$$S_{ver}(x^{(c)}, y^{(c)}) = \frac{1}{K} \sum_{k=1}^{K} J(p_y(y^{(c)}), p(y|x_k^{(c)})), \tag{4}$$

where $p_y(y^{(c)})$ is the one-hot vector for the given ground-truth label $y^{(c)}$ of the candidate sample and $p(y|x_k^{(c)})$ stands for the probabilistic label distribution of the k-th contrastive neighbor predicted using a classification model trained with all original samples including both clean and noisy ones. Here, instead of the model prediction at the candidate sample, we make use of model predictions at its contrastive neighbors, implicitly diversifying the predictive information of the candidate sample and making it relatively unbiased.

After computing the label inconsistency score for every candidate sample, we observe that if the given ground-truth label of a candidate sample is significantly different from the model prediction of its contrastive neighbor samples, *i.e.*, of large inconsistency, then the given label is very likely to be a noisy label. Therefore, by setting a threshold τ, we can classify candidate sample $x^{(c)}$ as a noisy sample if $S_{ver}(x^{(c)}, y^{(c)}) \geq \tau$, and otherwise, a clean one. To this end, we can obtain $\mathcal{D}_{\mathbf{clean}}$ and $\mathcal{D}_{\mathbf{noisy}}$ as follows,

$$\begin{aligned} \mathcal{D}_{\mathbf{clean}} &\leftarrow \{(x_i, y_i) \,|S_{ver}(x_i, y_i) < \tau, \forall\, (x_i, y_i) \in \mathcal{D}_{\mathbf{train}}\}, \\ \mathcal{D}_{\mathbf{noisy}} &\leftarrow \{(x_i,) \,|S_{ver}(x_i, y_i) \geq \tau, \forall\, (x_i, y_i) \in \mathcal{D}_{\mathbf{train}}\}. \end{aligned} \tag{5}$$

3.2 Neighborhood Collective Label Correction

After the neighborhood collective noise verification (NCNV) stage, we treat samples from $\mathcal{D}_{\mathbf{clean}}$ and $\mathcal{D}_{\mathbf{noisy}}$ as labeled and unlabeled samples respectively by simply discarding noisy labels to prevent noise memorization in the resulted classification model. To leverage the unlabeled samples, some studies have taken pseudo-labeling based methods to mine discriminative cues for model training [19,47,32], yet all of them resort to model predictions at individual unlabeled samples, again tracing back to the unavoidable bias. On the contrary, we set up neighborhood collective label correction (NCLC) stage, which corrects noisy labels by relying on neighboring clean samples to obtain more reliable and relatively unbiased pseudo-labels.

As in the NCNV stage, we first find K contrastive neighbors for each noisy sample $x^{(u)} \in \mathcal{D}_{\mathbf{noisy}}$ according to the ranked feature similarities between $x^{(u)}$ and its neighbors, as formulated below. At this time, we require all its contrastive neighbors to belong to the clean set $\mathcal{D}_{\mathbf{clean}}$.

$$\{(x_k^{(u)}, y_k^{(u)})\}, k = 1, \cdots, K \leftarrow \mathbf{KNN}(x^{(u)}; \mathcal{D}_{\mathbf{clean}}; K), \tag{6}$$

where $(x_k^{(u)}, y_k^{(u)})$ is a sample-label pair from $\mathcal{D}_{\mathbf{clean}}$. Unlike the NCNV stage, the ground-truth label information of contrastive neighbors is required in this stage.

Afterwards, we perform the following label consistency check between each candidate sample and its contrastive neighbors to mine those noisy samples that are similar to their neighboring samples in both the feature and label space,

$$S_{cor}(x^{(u)}) = \frac{1}{K}\sum_{k=1}^{K} J(p(y|x^{(u)}), p_y(y_k^{(u)})), \tag{7}$$

where $J(p(y|x^{(u)}), p_y(y_k^{(u)}))$ computes the discrepancy between the probabilistic label distribution of the candidate sample $x^{(u)}$ predicted using the trained classification model, and the one-hot vector for the given ground-truth label of its k-th contrastive neighbor. A large $S_{cor}(x^{(u)})$ indicates that the predicted label of the candidate sample is highly dissimilar to the clean and definite labels of its contrastive neighbors, suggesting that the candidate sample may lie near the decision boundary of the model. To be safe, we drop such candidate noisy samples if $S_{cor}(x^{(u)}) \geq \tau'$, where a second threshold τ' is used. In contrast, a candidate sample that satisfies $S_{cor}(x^{(u)}) < \tau'$ is more likely to be farther away from the decision boundary and could derive a more reliable pseudo-label from its contrastive neighbors. Therefore, we define a label correction function to generate a new label for such a noisy sample as follows,

$$\mathbf{Correct}(x^{(u)}) = \arg\max_{c} \sum_{k=1}^{K} w(x^{(u)}; k) \cdot p_y(y_k^{(u)}), \tag{8}$$

where we use $w(x^{(u)}; k) = 1 - J(p(y|x^{(u)}), p_y(y_k^{(u)}))$ to approximate the probability that the candidate sample belongs to the same class as its k-th contrastive neighbor, and $c = 1, \cdots, C$ indicates the c-th component of a label distribution vector has the maximum value. For convenience, we set $\hat{y}^{(u)} = \mathbf{Correct}(x^{(u)})$.

Finally, we define a new sample collection that contains all relabeled noisy samples as follows,

$$\mathcal{D}_{\mathbf{relab}} \leftarrow \{(x_i, \hat{y}_i) \,|\hat{y}_i = \mathbf{Correct}(x_i), S_{cor}(x_i) < \tau', \forall x_i \in \mathcal{D}_{\mathbf{noisy}}\}. \tag{9}$$

3.3 Training Objectives

Once we have the clean set $\mathcal{D}_{\mathbf{clean}}$ and relabeled set $\mathcal{D}_{\mathbf{relab}}$ respectively from the NCNV and NCLC steps, we use both datasets together to further optimize the classification model through fine-tuning. Auxiliary techniques are incorporated during model optimization. Since the initial classification model trained using both clean and noisy samples memorizes noisy labels during its training process and Mixup [50] can effectively attenuate such noise memorization, we first employ the mixup regularization to construct augmented samples through linear combinations of existing samples from $\mathcal{D}_{\mathbf{clean}}$.

Given two existing samples (x_i, y_i) and (x_j, y_j) from $\mathcal{D}_{\mathbf{clean}}$, an augmented sample $(\tilde{x}, \tilde{y})$ can be generated as follows,

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j, \tilde{y} = \lambda p_y(y_i) + (1-\lambda)p_y(y_j), \tag{10}$$

where $\lambda \sim Beta(\alpha)$ is a mixup ratio and α is a scalar parameter of Beta distribution. The cross-entropy loss applied to B augmented samples in each mini-batch is defined as follows,

$$\mathcal{L}^{mix} = -\sum_{b=1}^{B} \tilde{y}_b \log p(y|\tilde{x}_b). \tag{11}$$

In the NCLC stage, more reliable pseudo-labels are assigned to noisy samples farther away from the decision boundary. To leverage these relabeled samples during model optimization, we apply consistency regularization to them to further enhance the robustness of the model [8]. Label consistency is a good choice to achieve this goal because it encourages the fine-tuned model to produce the same output when there are minor perturbations in the input [36]. In practice, we enforce label consistency through the following loss:

$$\mathcal{L}^{lab} = -\sum_{b'=1}^{B'} p_y(y_{b'}) \log p(y|\mathbf{Aug}(x_{b'})), \tag{12}$$

where B' relabeled samples $(x_{b'}, y_{b'}) \in \mathcal{D}_{\mathbf{relab}}$ are chosen in each iteration, $p_y(y_{b'})$ is the one-hot vector of the pseudo-label of $x_{b'}$, $\mathbf{Aug}(\cdot)$ denotes the function that perturbs the chosen samples using Autoaugment technique proposed in [7], and $p(y|\mathbf{Aug}(x_{b'}))$ is the predicted label distribution of the perturbed sample. Proved by our experiments, this label consistency loss can be also applied to the selected clean samples from $\mathcal{D}_{\mathbf{clean}}$, especially under low noise ratios, to better boost the performance of the model.

As stated above, the overall loss function for final model fine-tuning is a combination of the cross-entropy and label consistency losses,

$$\mathcal{L}^{overall} = \mathcal{L}^{mix} + \gamma \mathcal{L}^{lab}, \tag{13}$$

where γ is a trade-off scalar to balance those two loss terms.

4 Experiments

4.1 Experimental Setup

Implementation. We highlight the effectiveness of our proposed NCE method on four standard LNL benchmark datasets: CIFAR-10 [16], CIFAR-100 [16], Clothing-1M [45] and Webvision-1.0 [23]. To be fair, we follow most details of the training and evaluation processes from the previous work "DivideMix" [19], such as network architectures, confidence penalty for asymmetric noise, and so on. Our code is publicly available at https://github.com/lijichang/LNL-NCE.

Table 1. Test accuracy (%) of our method (NCE) and existing state-of-the-art methods on the CIFAR-10 and CIFAR-100 datasets. (Mean accuracy and 95% confidence interval over 3 trails)

Dataset	CIFAR-10					CIFAR-100			
Noise type	Symmetric				Assymetric	Symmetric			
Method/Noise ratio	0.2	0.5	0.8	0.9	0.4	0.2	0.5	0.8	0.9
Cross-Entropy [19]	86.8	79.4	62.9	42.7	85.0	62.0	46.7	19.9	10.1
F-correction [33]	86.8	79.8	63.3	42.9	87.2	61.5	46.6	19.9	10.2
Co-teaching+ [49]	89.5	85.7	67.4	47.9	-	65.6	51.8	27.9	13.7
PENCIL [17]	92.4	89.1	77.5	58.9	88.5	69.4	57.5	31.1	15.3
LossModelling [3]	94.0	92.0	86.8	69.1	87.4	73.9	66.1	48.2	24.3
DivideMix [19]	96.1	94.6	93.2	76.0	93.4	77.3	74.6	60.2	31.5
ELR [24]	95.8	94.8	93.3	78.7	93.0	77.6	73.6	60.8	33.4
ProtoMix [21]	95.8	94.3	92.4	75.0	91.9	79.1	74.8	57.7	29.3
NGC [44]	95.9	94.5	91.6	80.5	90.6	79.3	75.9	62.7	29.8
NCE(best)	**96.2** ±0.09	**95.3** ±0.12	**93.9** ±0.22	**88.4** ±0.98	**94.5** ±0.70	**81.4** ±0.37	**76.3** ±0.28	**64.7** ±0.56	**41.1** ±0.54
NCE(last)	96.0 ±0.22	95.2 ±0.23	93.6 ±0.30	88.0 ±1.21	94.2 ±0.96	81.0 ±0.27	75.3 ±0.07	64.5 ±0.86	40.7 ±0.42

Table 2. Test accuracy (%) of our method (NCE) and existing state-of-the-art methods on the Clothing-1M dataset.

Meta-L. [20]	DivideMix [19]	ELR [24]	ELR+ [24]	NestedCoT. [6]	AugDesc [30]	NCE
73.5	74.8	72.9	74.8	74.9	75.1	**75.3**

CIFAR-10 and CIFAR-100 are two classic synthetic datasets for the LNL problem. We follow "DivideMix" [19] to create the noisy types, *i.e.*, "Symmetry" and "Asymmetry", and to set noise ratios, namely "0.20", "0.50", "0.80" and "0.90" for "Symmetry", and "0.40" for "Asymmetry". Similar to existing works [24,19,44], we also select PreAct Resnet [11] as the model backbone for CIFAR-10/CIFAR-100. Then we train it using a SGD optimizer with a momentum of 0.9 and a weight decay of 5×10^{-4} respectively. To better initialize our model, we set a warm-up step to perform supervised training on the model over all available samples using a standard cross-entropy loss. For effectiveness, this step is assigned a training period $T_{wu} = 10$ (or 30) for CIFAR-10 (or CIFAR-100). For adapting to diverse scenarios, we empirically set τ to 0.75 on CIFAR-10 or 0.90 on CIFAR-100, while τ' are usually set as 2×10^{-3} and 1×10^{-2} on CIFAR-10 and CIFAR-100, respectively. With respect to other hyper-parameters that are involved in NCE on CIFAR-10/CIFAR-100, we set $K = 20, T_{tr} = 300, \gamma = 1.0, \eta = 0.02$, $B = 128$, $B' = 128$ and $\alpha = 4$.

Clothing-1M and Webvision-1.0 are two large-scale real-world noisy datasets. Clothing-1M contains one million samples grabbed from the online shopping websites and Webvision-1.0 only uses top-50 classes originating from the Google image Subset of Webvision [23]. For Webvision-1.0, the results are reported from testing our model on both the WebVision validation set and the ImageNet ILSVRC12 validation set [38].

Baselines. We compare NCE with the following state-of-the-art algorithms to address the LNL problem on CIFAR-10 and CIFAR-100: "Cross-Entropy" [19], "F-correction" [33], "Co-teaching+" [49], "PENCIL" [17], "LossModelling" [3], "DivideMix" [19], "ELR" [24], "ProtoMix" [21] and "NGC" [44]. Herein, "Cross-

Table 3. Top-1 and top-5 test accuracy (%) of our method (NCE) and existing state-of-the-art methods on the Webvision and ImageNet ILSVRC12 validation sets. The models are trained on the training set of the Webvision-1.0 dataset

Method	WebVision		ILSVRC12	
	top-1	top-5	top-1	top-5
F-correction [33]	61.1	82.7	57.4	82.4
Decoupling [28]	62.5	84.7	58.3	82.3
MentorNet [15]	63.0	81.4	57.8	79.9
Co-teaching [10]	63.6	85.2	61.5	84.7
DivideMix [19]	77.3	91.6	75.2	90.8
ELR [24]	76.3	91.3	68.7	87.8
ELR+ [24]	77.8	91.7	70.3	89.8
NGC [44]	79.2	91.8	74.4	91.0
NCE	**79.5**	**93.8**	**76.3**	**94.1**

Entropy" trains the model only with a supervised cross-entropy loss over training samples along with given noisy labels, and its results are copied from "DivideMix". Besides methods stated above, we perform our comparison on Clothing-1M with previous methods, including "Meta-Learning" [20], "ELR+" [24], "NestedCoTeaching" [6] and "AugDesc" [30], where the augmentation strategy of our method on this dataset refers to that of "AugDesc" for comparison fairness. Moreover, we evaluate the proposed approach on Webvision-1.0 by newly adding "Decoupling" [28], "MentorNet" [15], and "Co-teaching" [10].

4.2 Comparisons with the State of the Art

Synthetic noisy datasets. CIFAR-10 and CIFAR-100 are two representative synthetic LNL benchmark datasets and we report results on these datasets in Table 1. For fair comparison, we follow all the settings in [19,44]. We can see that our NCE outperforms all existing state-of-the-art methods on CIFAR-10 and CIFAR-100 under all settings of symmetric (from 20% to 90%) and asymmetric (40% only) label noise ratio. In particular, on CIFAR-10, our method surpasses the best performing baselines by 7.9% and 1.1% at the highest symmetric and asymmetric noise ratios, respectively. In addition, in comparison to the performance of existing algorithms on CIFAR-100, NCE achieves the highest classification accuracy under all four noise ratio settings by exceeding the second best by 2.1%, 0.4%, 2.0% and 7.7%, respectively.

Real-world noisy datasets. To further verify the effectiveness of the proposed NCE method, we also conduct experiments on real-world noisy datasets, namely Clothing-1M and Webvision-1.0. Table 2 and Table 3 show performance comparisons between NCE and existing algorithms when these two are respectively used as the training set. We can observe that NCE achieves the highest accuracy on Clothing-1M and an improvement of 0.2% over "AugDesc", the best performing method among existing ones. Likewise, on the challenging Webvision-1.0, NCE again achieves higher performance than most existing methods in terms of top-1 and top-5 accuracy. These results further verify that our proposed approach can effectively perform well on the real-world noisy datasets.

Table 4. Ablation study of our method (NCE) on the CIFAR-10 and CIFAR-100 datasets under multiple label noise ratios. "repl." is an abbreviation for "replaced", and $\mathcal{L}^{ce}$ means the model is trained on the clean samples using a cross-entropy loss. (Only one of three trails is selected for comparison in our NCE method)

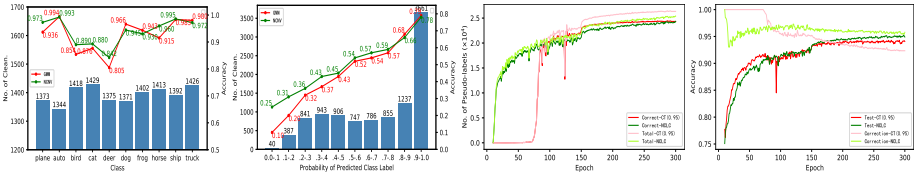
M-(#)	Dataset / Noise type / Method/Noise ratio	CIFAR-10 Symmetric 0.5	CIFAR-10 Symmetric 0.8	CIFAR-10 Assymetric 0.4	CIFAR-100 Symmetric 0.5	CIFAR-100 Symmetric 0.8	Mean
1	NCE	95.3	94.1	94.6	76.1	65.2	85.1
2	NCE repl. NCNV w/ GMM	94.8	79.0	89.7	75.8	56.8	79.2
3	NCE repl. NCLC w/ CT(0.95)	94.3	86.1	90.1	76.0	58.7	81.0
4	NCE repl. NCNV w/ GMM & w/o $\mathcal{L}^{lab}$	91.2	78.8	87.3	71.4	49.7	75.7
5	NCE w/o $\mathcal{L}^{lab}$	92.5	86.7	92.6	74.4	57.9	80.8
6	NCE repl. $\mathcal{L}^{mix}$ w/ $\mathcal{L}^{ce}$	93.3	78.5	89.0	73.2	55.2	77.8
7	NCE repl. perturbed w/ unperturbed in Eq. (12)	93.6	89.4	90.5	72.5	56.1	80.4

4.3 Analysis

To provide insights on how effectively each component of our algorithm works, we conduct an ablation study by removing or replacing individual components. Results of this ablation study are summarized in Table 4 and Fig. 3. Also, as displayed in and Fig. 4, we perform feature visualization to further analyze the proposed algorithm. All experiments are performed on both CIFAR-10 and CIFAR-100 datasets.

Effectiveness of NCNV step. To examine the effectiveness of the NCNV step in identifying clean/noisy labels, we replace NCNV with a well-known GMM-based strategy proposed in "DivideMix" [19]. In Table 4, a comparison between row M-(1) and row M-(2) reveals that our NCNV step significantly outperforms the GMM-based strategy because the former is capable of identifying clean labels of harder samples. Specifically, Fig. 3(a) and (b) show the power of our NCNV step in handling "hard" classes and "hard" samples in the clean subset. A class is considered "hard" when multiple methods have an overall low clean sample identification accuracy in the class, while a "hard" sample has a low probability (confidence) associated with its predicted class label. As Fig. 3(a) shows, our method achieves higher sensitivity on "hard" classes, *i.e.* "cat", "bird" and "deer", where both methods have the lowest identification accuracy. In addition, Fig. 3(b) also shows that our NCNV step works significantly better on "hard" samples, whose predicted class labels are associated with a low probability (confidence).

Effectiveness of NCLC step. To better understand the performance of the NCLC step in label correction, we replace NCLC with an existing label correction scheme, called Confidence Thresholding (CT) [36], which relabels such samples whose pseudo-labels have a confidence value exceeding a predefined threshold, *e.g.*, 0.95. According to row M-(3) of Table 4, NCLC clearly outperforms CT under all noise ratio settings. In detail, Fig. 3(c) and (d) reveal that CT works with few pseudo-labels in the early epochs. This is because, at that moment, the model cannot fit the training samples well and thus unlabeled samples with low-confidence predictions (< 0.95) would not be assigned pseudo-labels. Afterwards,



(a) (b) (c) (d)

Fig. 3. Analysis of ablation study results. **(a)** The accuracy of clean sample identification in various classes. **(b)** The accuracy of clean sample identification vs. the probability (confidence) of predicted class label. **(c)** The evolution of the numbers of pseudo-labels and correct pseudo-labels over epochs. **(d)** The evolution of label correction accuracy and test classification accuracy over epochs. The experiments for (a) and (b) are performed on CIFAR-10 and CIFAR-100 respectively with the same noise profile (Noise ratio: 0.80; Noise type: Symmetric). The blue bars represent the distribution of clean samples. (c) and (d) describe the same experiment, where we analyze the label correction performance of NCLC and Confidence Thresholding (*i.e.*, CT(0.95)) on CIFAR-10 (Noise ratio: 0.50; Noise type: Symmetric)

although plenty of unlabeled samples are given pseudo-labels as model training goes on, the label correction accuracy drops at the same time. Ultimately, it leads to lower performance than NCLC, which, on the other hand, obtains more reliable pseudo-labels for unlabeled (noisy) points.

Necessity of mixup regularization. To verify the importance of the mixup regularization, we remove it from our algorithm and then perform standard supervision over clean samples. As shown in row M-(6) of Table 4, this change causes very serious performance degradation, indicating that the mixup regularization is able to effectively attenuate noise memorization.

Necessity of consistency regularization. To investigate the effectiveness of consistency regularization over unlabeled (noisy) samples, we conduct two experiments. First, we disable $\mathcal{L}^{lab}$, meaning that the model is only trained over all clean samples. By comparing row M-(5) with row M-(1) in Table 4, we observe that the performance under all noise ratios drops by 1.7% to 7.6%, suggesting that this consistency loss is important for the performance of the model, especially when the noise ratio is high. In the second experiment, we replace the perturbed samples used in Eq. (12) with unperturbed ones to examine the need of sample perturbations. As shown in row M-(7) of Table 4, the average accuracy drops considerably by 4.7%. This demonstrates that sample perturbations in Eq. (12) play a significant role in realizing the full potential of consistency regularization.

Feature visualization. We use t-SNE [27] to visualize the feature distributions in both NCNV and NCLC steps. In Fig. 4(a)-(c), we show how the distributions of misidentified samples across diverse classes evolve in the model training process. It can be observed that as model training proceeds, the number of misclassifications in the training data decreases gradually. The misclassifications are distributed near the boundaries of the clusters corresponding to the classes, showing a good noise verification effect. Furthermore, in the NVLC step, as illustrated in Fig. 4(d), most well-relabeled samples are located in the core re-

(a) NCNV step, Epoch=30 **(b)** NCNV step, Epoch=60 **(c)** NCNV step, Epoch=300 **(d)** NCLC step, Epoch=300

Fig. 4. Feature visualization using t-SNE. We choose 10 representative classes on CIFAR-100 (Noise ratio: 0.80; Noise type: Symmetric). (**a**)-(**c**) show how the distributions of misidentifications in the NCNV step evolve during model training. They are involved in samples from $\mathcal{D}_{\mathbf{train}}$ corresponding to each representative class. In these subfigures, points in black are misclassified samples, such as clean (or noisy) samples misclassified as noisy (or clean) ones, in the training data, while samples in purple are correctly identified ones. The accuracy of training sample identification in (a)-(c) is 82.2%, 94.7% and 95.2%, respectively. (**d**) shows the feature distributions of unlabeled (noisy) samples in $\mathcal{D}_{\mathbf{noisy}}$ corresponding to 10 classes in the NCLC step, and points in bright colors, black and grey respectively denote correctly relabeled samples, mis-relabeled ones and dropped ones

gions of the clusters, while the mis-relabeled points and dropped ones are closer to the boundaries of the clusters or peripheral areas between different clusters. This meets our assumption stated in Section (3.2) that a candidate sample in the NVLC step that satisfies Eq. (9) is more likely to be farther away from the decision boundary of the model and could derive a more reliable pseudo-label.

5 Conclusions

In this paper, we have introduced a novel method called Neighborhood Collective Estimation (NCE) to tackle the problem of learning with noisy labels. In this method, we re-estimate the predictive reliability of a candidate sample by contrasting it against its feature-space nearest neighbors. This can enrich and diversify predictive information associated with the candidate and also makes such information relatively unbiased. The accuracy of noisy label identification and correction can thus be improved, facilitating subsequent model training. In detail, NCE consists of two steps, 1) Neighborhood Collective Noise Verification (NCNV) for separating all training data into clean samples and noisy ones, and 2) Neighborhood Collective Label Correction (NCLC) for relabeling noisy samples. Extensive experiments and a thorough ablation study have confirmed the superiority of our proposed method.

Acknowledgements

This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation (No.2020B1515020048), in part by the National Natural Science Foundation of China (No.61976250, No.U1811463), in part by the Guangzhou Science and technology project (No.202102020633), and in part by Hong Kong Research Grants Council through Research Impact Fund (Grant R-5001-18).

References

1. Algan, G., Ulusoy, I.: Image classification with deep learning in the presence of noisy labels: A survey. Knowledge-Based Systems **215**, 106771 (2021)
2. Arazo, E., Ortego, D., Albert, P., O'Connor, N.E., McGuinness, K.: Pseudo-labeling and confirmation bias in deep semi-supervised learning. In: 2020 International Joint Conference on Neural Networks (IJCNN). IEEE (2020)
3. Arazo, E., Ortego, D., Albert, P., O'Connor, N., McGuinness, K.: Unsupervised label noise modeling and loss correction. In: International Conference on Machine Learning. pp. 312–321. PMLR (2019)
4. Bahri, D., Jiang, H., Gupta, M.: Deep k-NN for noisy labels. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 540–550. PMLR (13–18 Jul 2020)
5. Bendre, N., Marín, H.T., Najafirad, P.: Learning from few samples: A survey. arXiv preprint arXiv:2007.15484 (2020)
6. Chen, Y., Shen, X., Hu, S.X., Suykens, J.A.: Boosting co-teaching with compression regularization for label noise. In: CVPR Learning from Limited and Imperfect Data (L2ID) workshop (2021)
7. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 113–123 (2019)
8. Englesson, E., Azizpour, H.: Consistency regularization can improve robustness to label noise. arXiv preprint arXiv:2110.01242 (2021)
9. Gunel, B., Du, J., Conneau, A., Stoyanov, V.: Supervised contrastive learning for pre-trained language model fine-tuning. In: International Conference on Learning Representations (2020)
10. Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels. In: NeurIPS. pp. 8535–8545 (2018)
11. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: European conference on computer vision. pp. 630–645. Springer (2016)
12. Huang, L., Zhang, C., Zhang, H.: Self-adaptive training: beyond empirical risk minimization. Advances in Neural Information Processing Systems **33** (2020)
13. Huang, Z., Niu, G., Liu, X., Ding, W., Xiao, X., Wu, H., Peng, X.: Learning with noisy correspondence for cross-modal matching. Advances in Neural Information Processing Systems **34**, 29406–29419 (2021)
14. Jiang, L., Huang, D., Liu, M., Yang, W.: Beyond synthetic noise: Deep learning on controlled noisy labels. In: International Conference on Machine Learning. pp. 4804–4815. PMLR (2020)
15. Jiang, L., Zhou, Z., Leung, T., Li, L.J., Fei-Fei, L.: Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In: International Conference on Machine Learning. pp. 2304–2313. PMLR (2018)
16. Krizhevsky, A.: Learning multiple layers of features from tiny images. Master's thesis, University of Tront (2009)
17. Kun, Y., Jianxin, W.: Probabilistic End-to-end Noise Correction for Learning with Noisy Labels. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
18. Li, J., Li, G., Shi, Y., Yu, Y.: Cross-domain adaptive clustering for semi-supervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2505–2514 (June 2021)

19. Li, J., Socher, R., Hoi, S.C.: Dividemix: Learning with noisy labels as semi-supervised learning. arXiv preprint arXiv:2002.07394 (2020)
20. Li, J., Wong, Y., Zhao, Q., Kankanhalli, M.S.: Learning to learn from noisy labeled data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5051–5059 (2019)
21. Li, J., Xiong, C., Hoi, S.C.: Learning from noisy data with robust representation learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9485–9494 (October 2021)
22. Li, W., Li, F., Luo, Y., Wang, P., et al.: Deep domain adaptive object detection: A survey. In: 2020 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1808–1813. IEEE (2020)
23. Li, W., Wang, L., Li, W., Agustsson, E., Gool, L.V.: Webvision database: Visual learning and understanding from web data. Arxiv Preprint (2017)
24. Liu, S., Niles-Weed, J., Razavian, N., Fernandez-Granda, C.: Early-learning regularization prevents memorization of noisy labels. Advances in Neural Information Processing Systems **33** (2020)
25. Ma, X., Wang, Y., Houle, M.E., Zhou, S., Erfani, S., Xia, S., Wijewickrema, S., Bailey, J.: Dimensionality-driven learning with noisy labels. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 3355–3364. PMLR (10–15 Jul 2018)
26. Ma, Z., Leijon, A.: Bayesian estimation of beta mixture models with variational inference. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(11), 2160–2173 (2011)
27. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
28. Malach, E., Shalev-Shwartz, S.: Decoupling" when to update" from" how to update". Advances in Neural Information Processing Systems **30**, 960–970 (2017)
29. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 26. Curran Associates, Inc. (2013), https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf
30. Nishi, K., Ding, Y., Rich, A., Hollerer, T.: Augmentation strategies for learning with noisy labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8022–8031 (2021)
31. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
32. Ortego, D., Arazo, E., Albert, P., O'Connor, N.E., McGuinness, K.: Multi-objective interpolation training for robustness to label noise. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6606–6615 (2021)
33. Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., Qu, L.: Making deep neural networks robust to label noise: A loss correction approach. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1944–1952 (2017)
34. Permuter, H., Francos, J., Jermyn, I.: A study of gaussian mixture models of color and texture features for image classification and segmentation. Pattern recognition **39**(4), 695–706 (2006)

35. Ren, M., Zeng, W., Yang, B., Urtasun, R.: Learning to reweight examples for robust deep learning. In: International Conference on Machine Learning. pp. 4334–4343. PMLR (2018)
36. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Advances in Neural Information Processing Systems **33** (2020)
37. Song, H., Kim, M., Lee, J.G.: Selfie: Refurbishing unclean samples for robust deep learning. In: International Conference on Machine Learning. pp. 5907–5915. PMLR (2019)
38. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI conference on artificial intelligence (2017)
39. Tanaka, D., Ikami, D., Yamasaki, T., Aizawa, K.: Joint optimization framework for learning with noisy labels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5552–5560 (2018)
40. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 1195–1204 (2017)
41. Van Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. Machine Learning **109**(2), 373–440 (2020)
42. Wang, Y., Liu, W., Ma, X., Bailey, J., Zha, H., Song, L., Xia, S.T.: Iterative learning with open-set noisy labels. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8688–8696 (2018)
43. Wu, P., Zheng, S., Goswami, M., Metaxas, D.N., Chen, C.: A topological filter for learning with label noise. Advances in neural information processing systems **33** (2020)
44. Wu, Z.F., Wei, T., Jiang, J., Mao, C., Tang, M., Li, Y.F.: Ngc: A unified framework for learning with open-world noisy data (2021)
45. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2691–2699 (2015)
46. Yang, M., Huang, Z., Hu, P., Li, T., Lv, J., Peng, X.: Learning with twin noisy labels for visible-infrared person re-identification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14308–14317 (2022)
47. Yao, Y., Sun, Z., Zhang, C., Shen, F., Wu, Q., Zhang, J., Tang, Z.: Jo-src: A contrastive approach for combating noisy labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5192–5201 (2021)
48. Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., Sugiyama, M.: How does disagreement help generalization against label corruption? In: International Conference on Machine Learning. pp. 7164–7173. PMLR (2019)
49. Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., Sugiyama, M.: How does disagreement help generalization against label corruption? In: International Conference on Machine Learning. pp. 7164–7173. PMLR (2019)
50. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
51. Zheltonozhskii, E., Baskin, C., Mendelson, A., Bronstein, A.M., Litany, O.: Contrast to divide: Self-supervised pre-training for learning with noisy labels. arXiv preprint arXiv:2103.13646 (2021)

52. Zhou, H.Y., Chen, X., Zhang, Y., Luo, R., Wang, L., Yu, Y.: Generalized radiograph representation learning via cross-supervision between images and free-text radiology reports. Nature Machine Intelligence **4**(1), 32–40 (2022)
53. Zhu, Z., Song, Y., Liu, Y.: Clusterability as an alternative to anchor points when learning with noisy labels. In: International Conference on Machine Learning. pp. 12912–12923. PMLR (2021)