NGC: A Unified Framework for Learning with Open-World Noisy Data

Zhi-Fan Wu[1,2][*], Tong Wei[1][*], Jianwen Jiang[2][*], Chaojie Mao[2], Mingqian Tang[2], Yu-Feng Li[1][†]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2]Alibaba Group, China

{wuzf, weit}@lamda.nju.edu.cn, liyf@nju.edu.cn
{jianwen.jjw, chaojie.mcj, mingqian.tmq}@alibaba-inc.com

Abstract

The existence of noisy data is prevalent in both the training and testing phases of machine learning systems, which inevitably leads to the degradation of model performance. There have been plenty of works concentrated on learning with in-distribution (IND) noisy labels in the last decade, i.e., some training samples are assigned incorrect labels that do not correspond to their true classes. Nonetheless, in real application scenarios, it is necessary to consider the influence of out-of-distribution (OOD) samples, i.e., samples that do not belong to any known classes, which has not been sufficiently explored yet. To remedy this, we study a new problem setup, namely Learning with Open-world Noisy Data (LOND). The goal of LOND is to simultaneously learn a classifier and an OOD detector from datasets with mixed IND and OOD noise. In this paper, we propose a new graph-based framework, namely Noisy Graph Cleaning (NGC), which collects clean samples by leveraging geometric structure of data and model predictive confidence. Without any additional training effort, NGC can detect and reject the OOD samples based on the learned class prototypes directly in testing phase. We conduct experiments on multiple benchmarks with different types of noise and the results demonstrate the superior performance of our method against state of the arts.

1. Introduction

Deep neural networks (DNNs) have gained popularity in a variety of applications. Despite their success, DNNs often rely on the availability of large-scale labeled training datasets. In practice, data annotation inevitably introduces label noise, and it is extremely expensive and time-consuming to clean up the corrupted labels. The existence
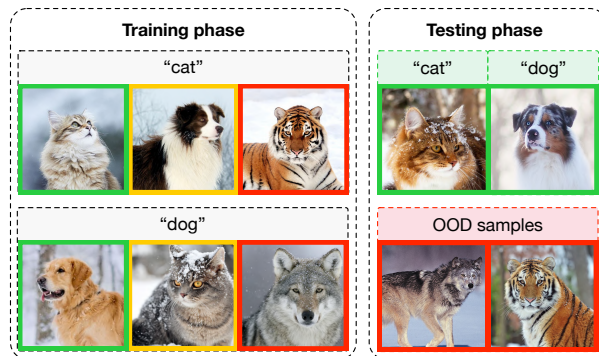
Figure 1: A demonstration of the LOND setup. We use green boxes to represent clean samples while yellow and red boxes are IND and OOD noisy samples, respectively.

of label noise can be problematic for overparameterized deep networks, as they may overfit to label noise even on randomly-assigned labels [54]. Therefore, mitigating the effects of noisy labels becomes a critical issue.

When learning with noisy labels (LNL), plenty of promising methods have been proposed to improve the generalization [39, 7, 42, 22, 40, 56, 48, 49, 47]. Many existing methods work by analyzing output predictions to identify mislabeled samples [51, 35, 21] or reweighting samples to alleviate the influence of noisy labels [36, 1]. Note that, these methods are particularly designed to deal with in-distribution (IND) label noise. Some other works also consider the existence of out-of-distribution (OOD) noise in training datasets [41, 20]. Their basic assumption is that clean samples are clustered together while OOD samples are widely scattered in the feature space.

Although significant performance improvement is achieved, most existing LNL works only take account of OOD samples in training phase, while the existence of OOD samples in testing phase is neglected, which is crucial for machine learning systems in real applications [10, 19, 38]. In this paper, we study this practical problem, i.e., the ex-

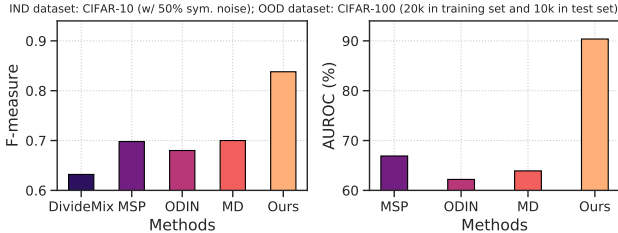IND dataset: CIFAR-10 (w/ 50% sym. noise); OOD dataset: CIFAR-100 (20k in training set and 10k in test set)

Figure 2: Performance on testing dataset with extra OOD samples. MSP, ODIN, MD are combined with DivideMix.

istence of both IND and OOD noise in training phase, as well as the presence of OOD samples in testing phase. We name this new setup as learning with open-world noisy data (LOND). An illustration of the LOND setup can be found in Figure 1. A straightforward approach to address LOND is to combine LNL methods with OOD detectors [10, 27, 19]. However, we empirically find that such direct combinations lead to unsatisfactory results as shown in Figure 2. Therefore, obtaining models that can handle IND and OOD noise in both training and testing phases remains challenging.

To address the LOND problem, we present Noisy Graph Cleaning (NGC), a unified framework for learning with open-world noisy data. Different from previous LNL methods that utilize either model predictions [21, 35, 23, 24] or neighborhood information [41, 45], where the interaction between model predictions and geometric structure of data is neglected, NGC simultaneously takes advantage of output confidence and the geometric structure. With the help of graph structure, we find that the confidence-based strategy can break the connectivity between clean and noisy samples, which significantly facilitates the geometry-based strategy. In specific, NGC iteratively constructs the nearest neighbor graph using latent representations of training samples. Given the graph structure, NGC corrects IND noisy labels by aggregating information from neighborhoods through soft pseudo-label propagation. Then, to remove the OOD and remaining obstinate IND noise, we present subgraph selection. It first degrades the connectivity between clean and noisy samples by removing samples with low-confidence predictions. Then, subgraphs corresponding to the largest connected component are constructed for each class. Moreover, NGC employs the devised contrastive losses [46, 4, 15] to refine the representations from both instance-level and subgraph-level, which in return benefits label correction and subgraph selection. At test time, NGC can readily detect and reject OOD samples by calculating distances to learned class prototypes.

The main contributions of this work are:

1. We study a new problem, that is, the training set contains both IND and OOD noise and the test set contains OOD samples, which is practical in real applications.

2. We propose a new graph-based noisy label learning framework, NGC, which corrects IND noisy labels and sieves out OOD samples by utilizing the confidence of model predictions and geometric structure of data. Without any additional training effort, NGC can detect and reject OOD samples at testing time.

3. We evaluate NGC on multiple benchmark datasets under various noise types as well as real-world tasks. Experimental results demonstrate the superiority of NGC over the state-of-the-art methods.

The rest of the paper is organized as follows. First, we introduce some related work. Then, we present the studied learning problem and the proposed framework. Furthermore, we experimentally analyze the proposed method. Finally, we conclude this paper.

2. Related Work

Learning from Noisy Labels is a heavily studied problem. Many methods attempt to rectify the loss function, which can be categorized into two types. The first type treats samples equally and rectifies the loss by either removing or relabeling noisy samples [9, 35, 50, 32]. For example, AUM [35] designs a margin-based method for detecting noisy samples by observing that clean samples have a larger margin than noisy samples. TopoFilter [45] assumes that clean data is clustered together while noisy samples are isolated. Joint-Optim [39] and PENCIL [51] treat labels as learnable variables, which are jointly optimized along with model parameters. Another type of method learns to reweight samples with higher weights for clean data points [29, 36, 16]. Instead of using a fixed weight for all samples, M-correction [1] uses dynamic hard and soft bootstrapping loss to dynamically reweight training samples. Some recent works resort to early-learning regularization [28] and data augmentation [33] to handle noisy labels.

The above methods only consider IND label noise in training datasets. Recently, some works [41, 20, 37, 24, 23] propose to handle both IND and OOD noise in training datasets. For instance, ILON [41] discriminates noise samples by density estimating. MoPro [24] and ProtoMix [23] identify IND and OOD noise according to predictive confidence. However, these approaches cannot be directly applied for detecting OOD at test time, and the performance of simply combining with existing OOD detection methods is not satisfactory. In this work, we introduce a new framework that simultaneously learns a classifier and an OOD detector from training data with both IND and OOD noise.

OOD Detection aims to identify test data points that are far from the training distribution. According to whether requiring labels during training time, OOD detection methods can be categorized into supervised learning methods [18, 27, 19, 52, 11] and unsupervised learning meth-

ods [5, 6, 38]. For example, ODIN [27] separates IND and OOD samples by using temperature scaling and adding perturbations to the input. Lee et al. [19] obtains the class conditional Gaussian distributions and calculates confidence score based on Mahalanobis distance. Recently, SSD [38] uses self-supervised learning to extract latent feature representations and Mahalanobis distance to compute the membership score between test data points and IND samples.

Compared with supervised detectors, NGC does not assume the availability of clean datasets which are often difficult to obtain in many real-world applications [43, 44]. Instead, NGC can detect OOD examples by training on noisy-labeled datasets.

3. Learning with Open-World Noisy Data

In this section, we first introduce the studied problem setup and an overview of the proposed noisy graph cleaning framework. Then, we present the proposed framework.

3.1. Problem Formulation

Given a training dataset $\mathcal{D}_{train} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$, where $\boldsymbol{x}_i$ is an instance feature representation and $y_i \in \mathcal{C} = \{1, \ldots, K\}$ is the class label assigned to it. In $\mathcal{D}_{train}$, we assume that the instance-label pair $(\boldsymbol{x}_i, y_i), 1 \leq i \leq N$ consists of three types. Denote y_i^* as the ground-truth label of $\boldsymbol{x}_i$, a **correctly-labeled sample** whose assigned label matches the ground-truth label, i.e., $y_i = y_i^*$. An **IND mislabeled sample** has an assigned label that does not match the ground-truth label, but the input matches one of the classes in $\mathcal{C}$, i.e., $y_i \neq y_i^*$ and $y_i^* \in \mathcal{C}$. An **OOD mislabeled sample** is one where the input does not match the assigned label and other known classes, i.e., $y_i \neq y_i^*$ and $y_i^* \notin \mathcal{C}$. In inference, there are two types of test samples. An **IND sample** is one where $\boldsymbol{x}$ is taken from the distribution of one of the known classes, i.e., $y_i^* \in \mathcal{C}$. An **OOD sample** is the one taken from unknown class distributions, i.e., $y_i^* \notin \mathcal{C}$.

3.2. An Overview of the Proposed Framework

To address the LOND problem, we present a graph-based framework, named Noisy Graph Cleaning (NGC), which can exploit the relationships among data and learn robust representations from reliable data. Initially, a k-NN graph is constructed, where samples are represented as vertices (nodes) in the graph with edges represent similarities between samples. Since labels of samples may be mislabeled, we refer to the resulting graph as *noisy graph*. Then, NGC accomplishes noisy graph cleaning in two steps. First, to cope with IND noise, NGC refines noisy labels using the proposed soft pseudo-label propagation based on the smoothness assumption [57, 58, 13]. Second, since OOD samples do not belong to any IND classes, soft pseudo-label propagation is not able to correct

their labels. We propose to collect a subset of clean samples to guide the learning of the network. To achieve this goal, a two-stage subgraph selection method is introduced, i.e., confidence-based and geometry-based selection. The confidence-based strategy breaks the edges between nodes with clean labels and noisy labels by removing samples with low-confidence predictions. Then the geometry-based strategy selects nodes that are likely to be clean. Figure 3 provides an illustration of the proposed method. We observe that these two selection strategies are indispensable and single application of each one leads to inferior performance. Based on that, we employ devised instance-level and subgraph-level contrastive losses to learn robust representations, which in return can benefit the construction of graph and the subgraph selection. In each training iteration, the graph is re-constructed and noise correction as well as subgraph selection are performed. Then, the selected clean samples are used for the training of DNNs.

3.3. Graph-based Noise Correction

The goal of noise correction is to propagate labels on the undirected graph $G = \langle V, E \rangle$ by leveraging similarities between data. V and E denote the set of graph vertices and edges, respectively. In graph G, the similarities between vertices are encoded by a weight matrix $\boldsymbol{W}$. For scalability, we adopt the k-NN matrix, which is obtained by:

$$\boldsymbol{W}_{ij} := \begin{cases} \left[\boldsymbol{z}_i^\top \boldsymbol{z}_j\right]_+^\gamma, & \text{if } i \neq j \wedge \boldsymbol{z}_i \in \mathbf{NN}_k\left(\boldsymbol{z}_j\right) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Here, γ is a parameter simply set as $\gamma = 1$ in our experiments. $\boldsymbol{z}_i$ is the latent representation for $\boldsymbol{x}_i$ and $\mathbf{NN}_k$ denotes the k nearest neighbors. To capture high-order graph information, researchers have designed models on the assumption that labels vary smoothly over the edges of the graph [57, 58, 13, 26]. In this work, we propose to propagate soft pseudo-labels obtained from the network. Denote $\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_N] \in \mathbb{R}^{N \times K}$ as the initial label matrix. We set $\boldsymbol{y}_i$ to the one-hot label vector of $\boldsymbol{x}_i$ if $\boldsymbol{x}_i$ is selected as a clean sample by our method introduced in Section 3.4, otherwise we use model prediction aggregated by temporal ensemble [17, 32] to initialize it. Let $\boldsymbol{D}$ be the diagonal degree matrix for $\boldsymbol{W}$ with entry $d_{ii} = \sum_j \boldsymbol{W}_{ij}$, we obtain the refined soft pseudo-labels $\tilde{\boldsymbol{Y}} = [\tilde{\boldsymbol{y}}_1, \cdots, \tilde{\boldsymbol{y}}_N] \in \mathbb{R}^{N \times K}$ by solving the following minimization problem:

$$J(\tilde{\boldsymbol{Y}}) := \frac{\alpha}{2} \sum_{i,j=1}^N \boldsymbol{W}_{ij} \left\| \frac{\tilde{\boldsymbol{y}}_i}{\sqrt{d_{ii}}} - \frac{\tilde{\boldsymbol{y}}_j}{\sqrt{d_{jj}}} \right\|^2 + (1-\alpha)\|\boldsymbol{Y} - \tilde{\boldsymbol{Y}}\|_F^2 \quad (2)$$

In Eq. (2), all nodes propagate pseudo-labels to their neighbors according to edge weights. α is used to trade-off between information from neighborhoods and vertices themselves and we simply set it to 0.5 in all experiments. This
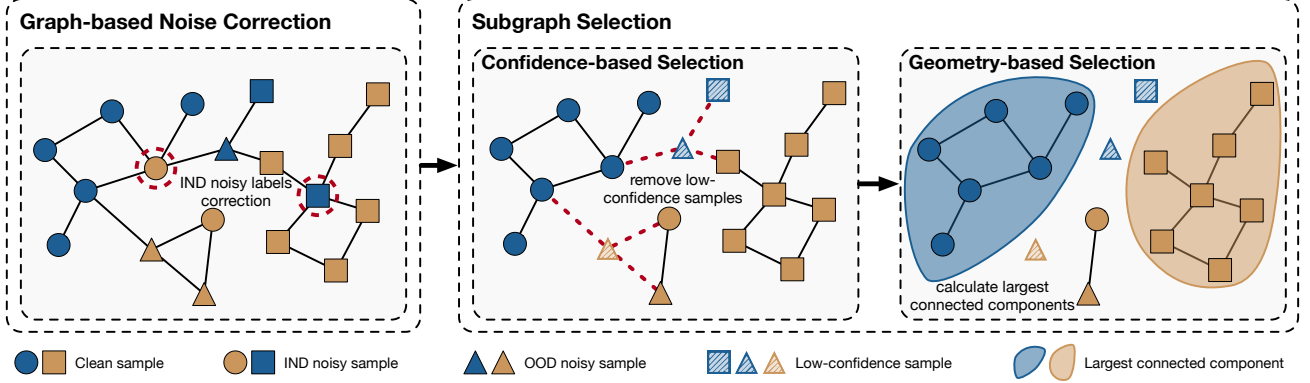
Figure 3: An illustration of graph-based noise correction and subgraph selection in binary classification case.

minimization problem can be solved by using conjugate gradient as [58, 13]. After obtaining refined soft pseudo-labels, it is common to transform $\tilde{Y}$ into hard pseudo-labels to guide the training. Specifically, in iteration t, the hard pseudo-label for the i-th data point is generated by taking the largest prediction score as $\hat{y}_i = \arg\max_k \tilde{Y}_{ik}^{(t)}$.

3.4. Subgraph Selection

When training DNNs with noisy labels, it is observed that clean samples of the same class are usually clustered together in the latent feature space, while noisy samples are pushed away from these clusters [20, 45]. This inspires us to find the connected component with the same class label in the graph for each class. Unfortunately, OOD samples can be similar to some clean samples, leading to undesirable edges in the graph such that nodes corresponding to OOD samples are included in the largest connected component (LCC). To remedy this, we introduce confidence-based selection to remove edges associated with low-confidence nodes because these edges are unreliable. After that, the geometry-based selection is employed to obtain the LCC in subgraphs of each class.

Confidence-based Sample Selection. Since low-confidence nodes are more likely to connect OOD nodes to the clusters of clean nodes, we use a sufficiently high threshold $\eta \in [0, 1]$ to select a reliable subset of nodes:

$$g_i = \begin{cases} 1, & \text{if } \tilde{Y}_{iy_i}^{(t)} > \frac{1}{K} \\ \mathbb{I}\left[\max_k \tilde{Y}_{ik}^{(t)} > \eta\right], & \text{otherwise} \end{cases} \quad (3)$$

where g_i is a binary indicator representing the conservation of node $v_i \in V$ when $g_i = 1$ and the removal of node v_i when $g_i = 0$. Note that we have another condition $\tilde{Y}_{iy_i}^{(t)} > \frac{1}{K}$ which is complementary to the high-confidence condition inspired by previous works [24, 23]. The reason is that the network may not produce confident predictions in the early phase of training, while it has been observed to

first fit the training data with clean labels [35, 28]. Therefore, we incline to treat label y_i as clean if its corresponding prediction score is higher than uniform probability $\frac{1}{K}$, and we set $\hat{y}_i = y_i$. Then we refine graph G based on the indicator g as $\tilde{V} = V \setminus \{v \mid \forall v \in V, g_v = 0\}$ and $\tilde{E} = E \setminus \{e \mid \forall e = \langle e_1, e_2 \rangle \in E, g_{e_1} + g_{e_2} < 2\}$. In this way, low-confidence nodes and their corresponding edges are removed from graph G and the resulting graph is denoted by $\tilde{G} = \langle \tilde{V}, \tilde{E} \rangle$. In the modified graph $\tilde{G}$, the connectivity between nodes are more reliable, which facilitates to the geometry-based selection.

Geometry-based Sample Selection. In graph $\tilde{G}$, we expect that nodes with same labels are connected. Since nodes with noisy labels locate far away from clean ones, more than one connected component may exist for each class. Therefore, we selected the LCC for robustness. Specifically, for the k-th class, graph nodes that possess labels of other classes, i.e., $\hat{y}_i \neq k, \forall i \in [N]$, and their adjacent edges $\tilde{G}$ are removed. We denote this as the class-specific subgraph for class k as $\tilde{G}(k)$. Let $\tilde{G}(k)_{lcc}$ be the set of nodes in the LCC of $\tilde{G}(k)$, we obtain a subset of clean samples by $\mathcal{S} = \bigcup_{k=1}^{K} \tilde{G}(k)_{lcc}$. Note that a connected component of $\tilde{G}(k)$ is a subgraph in which any two vertices are connected by edges, and which is not connected to any other vertex in the rest of the graph. In other words, we consider data points belonging to the LCC of the class-specific subgraphs for each class to be clean, since small connected components may contain noisy samples. In practice, we implement disjoint-set data structures to compute the components effectively.

In summary, we identify clean samples by using both predictive confidence and the geometric structure of data:

$$g_i = \begin{cases} \mathbb{I}[i \in \mathcal{S}], & \text{if } \tilde{Y}_{iy_i}^{(t)} > \frac{1}{K} \\ \mathbb{I}\left[\max_k \tilde{Y}_{ik}^{(t)} > \eta\right] \cdot \mathbb{I}[i \in \mathcal{S}], & \text{otherwise} \end{cases} \quad (4)$$

3.5. Subgraph-level Contrastive Learning

It is noted that exploring the similarities between samples is essentially based on meaningful feature representations. To this end, we take advantage of contrastive learning, which has been successfully used to learn good representations in many tasks [46, 4, 15, 23]. The basic idea of contrastive learning is to pull together two embeddings of the same samples, while pushing apart embeddings of other samples. Formally, the instance-level contrastive loss is obtained as follows.

$$\mathcal{L}^{\text{inst}} = -\sum_{i \in I} \log \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_{j(i)}/\tau_1\right)}{\sum_{a \in A(i)} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_a/\tau_1\right)} \quad (5)$$

Here $\boldsymbol{z}_i = \text{Proj}\left(\text{Enc}\left(\boldsymbol{x}_i\right)\right) \in \mathbb{R}^{D_P}$ denotes the l_2 normalized feature representation with dimension D_P, and τ_1 is a scalar temperature parameter. I denotes the set of training samples, I' is another augmented set, and $A(i) = (I\backslash\{i\}) \cup I'$. Different augmentation strategies can be used on I and I' as [23]. We use $j(i)$ to denote the index of the other augmented sample of $\boldsymbol{x}_i$.

However, direct optimization of the instance-level contrastive objective in Eq. (5) is ineffective, which does not leverage the label information and the geometry of data. To this end, we design a subgraph-level contrastive loss:

$$\mathcal{L}^{\text{subgraph}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_p/\tau_2\right)}{\sum_{a \in A(i)} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_a/\tau_2\right)}$$
$$(6)$$

Here $P(i) = \{p \in A(i) : \hat{y}_p = \hat{y}_i \wedge g_p + g_i = 2\}$, and $|P(i)|$ is its cardinality. In the calculation of $|P(i)|$, $g_i = 1$ indicates that only selected clean samples by NGC are used for training. τ_2 is another temperature parameter. For each class, samples belonging to the corresponding LCC are pulled together by optimizing Eq. (6). In return, it benefits the clean data selection because more samples of the same class are connected in the k-NN graph.

Considering the above definitions and denoting $\mathcal{L}^{\text{ce}}$ as conventional cross-entropy loss, the overall training objective is written as follows.

$$\mathcal{L} = \mathcal{L}^{\text{ce}} + \lambda_1 \mathcal{L}^{\text{inst}} + \lambda_2 \mathcal{L}^{\text{subgraph}}, \quad (7)$$

where hyperparameters λ_1 and λ_2 are simply set to 1 in all experiments. We adopt DNN model as feature extractor $\text{Enc}(\cdot)$ and a linear layer as projector $\text{Proj}(\cdot)$ to generate latent feature representation $\boldsymbol{z}_i$. Another linear layer following the feature extractor is used as classifier. Finally, we train the network by minimizing the total loss in Eq. (7).

3.6. OOD Detection

By far, NGC is able to learn classifiers from data with mixed IND and OOD noise. To fully achieve the goal of LOND, the framework must account for the presence of OOD samples at test time. This motivates us to design a principled way to detect OOD samples by measuring the class-conditional probability. Specifically, given a feature representation learned from NGC, the class-conditional probability is computed based on the similarity between the latent representation of input $\boldsymbol{x}$ and the class prototypes $\{\boldsymbol{c}_k\}_{k=1}^K$, where $\boldsymbol{c}_k$ is the normalized mean embedding for selected clean samples of class k, and can be obtained by:

$$\boldsymbol{c}_k = \text{Normalize}(\frac{1}{\sum_{i \in \mathcal{I}_k} g_i} \sum_{i \in \mathcal{I}_k} g_i \boldsymbol{z}_i), \quad (8)$$

where $\mathcal{I}_k$ denotes the set of samples for which the corresponding pseudo-labels $\hat{y}_i = k, \forall i \in [N]$. Then, the maximum class-wise similarity is computed as follows.

$$s\left(\boldsymbol{x}\right) := \max_{k \in [K]} \text{sim}\left(\boldsymbol{z}, \boldsymbol{c}_k\right). \quad (9)$$

Here $\boldsymbol{z} = \text{Proj}\left(\text{Enc}\left(\boldsymbol{x}\right)\right)$ and sim stands for any similarity measure. In practice, we measure cosine similarity to compute $s(\boldsymbol{x})$. When detecting OOD samples, the lower $s\left(\boldsymbol{x}\right)$ is, the more likely it is to be an OOD sample. To make hard decisions, the probability threshold ζ is used. That is, a testing point $\boldsymbol{x}$ is deemed as OOD if and only if $s(\boldsymbol{x}) < \zeta$.

4. Experiments

In this section, we investigate the performance of the proposed NGC on multiple datasets with various label noises. Specifically, we introduce our experiments in three aspects as shown in Table 1. We verify the effectiveness of our method in the proposed LOND task and learning with closed-world noisy labels (LCNL) as well as learning from real-world noisy dataset (LRND) tasks in order.

Table 1: Three types of tasks considered in our experiments.

Setup	IND noise in $\mathcal{D}_{train}$	OOD in $\mathcal{D}_{train}$	OOD in $\mathcal{D}_{test}$
LOND	✓	✓	✓
LCNL	✓	✗	✗
LRND	✓	✓	✗

Implementation details. For all CIFAR experiments, we train PreAct ResNet-18 network using SGD optimizer with momentum 0.9 and weight decay $5 \cdot 10^{-4}$. The initial learning rate is set to 0.15 and cosine decay schedule is used. The batch size is set to 512 and the dimension of projector layer is set to 64. For CIFAR-10 experiments, we use $k = 30$ for sym. noise and $k = 10$ for asym. noise, warmup with cross-entropy loss for 5 epochs. For CIFAR-100 experiments, we set $k = 200$ and warmup for 30 epochs. The network is trained for 300 epochs. Mixup [55] and AugMix [12] are used as data augmentation. We provide detailed experimental settings in the supplementary material.

Table 2: Test accuracy (%) under mixed IND and OOD noise compared with state-of-the-art LNL methods. 50% sym. IND noise is injected into dataset. We run methods three times with different seeds and report the mean and the standard deviation.

IND dataset	OOD dataset	# OOD	CE	RoG [20]	ILON [41]	DivideMix [21]	Ours
CIFAR-10	CIFAR-100	10k	$53.36_{\pm0.92}$	$63.01_{\pm0.46}$	$75.17_{\pm1.50}$	$92.73_{\pm0.27}$	$\mathbf{93.69_{\pm0.09}}$
		20k	$50.73_{\pm0.80}$	$62.56_{\pm1.76}$	$74.85_{\pm1.61}$	$92.26_{\pm0.13}$	$\mathbf{92.31_{\pm0.29}}$
	TinyImageNet	10k	$51.85_{\pm1.09}$	$61.69_{\pm1.18}$	$75.93_{\pm1.13}$	$\mathbf{94.08_{\pm0.18}}$	$93.73_{\pm0.36}$
		20k	$52.32_{\pm1.41}$	$63.15_{\pm1.13}$	$74.63_{\pm0.74}$	$\mathbf{93.83_{\pm0.08}}$	$93.54_{\pm0.21}$
	Places-365	10k	$54.06_{\pm0.53}$	$64.21_{\pm0.27}$	$76.17_{\pm0.90}$	$93.81_{\pm0.33}$	$\mathbf{94.18_{\pm0.09}}$
		20k	$55.30_{\pm1.31}$	$63.52_{\pm1.73}$	$76.36_{\pm1.26}$	$93.59_{\pm0.07}$	$\mathbf{93.67_{\pm0.22}}$
CIFAR-100	TinyImageNet	10k	$37.01_{\pm0.40}$	$52.65_{\pm0.30}$	$51.43_{\pm0.29}$	$70.38_{\pm0.09}$	$\mathbf{74.57_{\pm0.23}}$
		20k	$34.55_{\pm0.55}$	$50.40_{\pm0.44}$	$50.14_{\pm0.66}$	$69.89_{\pm0.25}$	$\mathbf{73.49_{\pm0.11}}$
	Places-365	10k	$37.53_{\pm0.54}$	$52.43_{\pm0.03}$	$50.74_{\pm0.65}$	$70.01_{\pm0.11}$	$\mathbf{74.89_{\pm0.21}}$
		20k	$34.54_{\pm0.18}$	$50.32_{\pm0.29}$	$49.87_{\pm0.46}$	$69.84_{\pm0.15}$	$\mathbf{73.44_{\pm0.35}}$

Table 3: AUROC (%) comparison with state-of-the-art OOD detectors. 50% sym. IND noise is injected into training dataset. 20k and 10k OOD samples are added into training set and test set, respectively. $^+$ indicates supervised detection methods.

IND dataset	OOD dataset	MSP[10]$^+$	ODIN[27]$^+$	MD[19]$^+$	Rot[6]	Rot[11]$^+$	SSD[38]	SSD[38]$^+$	Ours
CIFAR-10	CIFAR-100	69.91	65.40	64.45	63.84	60.25	68.42	55.88	**90.37**
	TinyImageNet	70.12	67.31	77.55	68.87	64.64	75.51	60.52	**94.18**
	Places-365	71.08	71.12	70.83	50.42	69.35	77.11	62.30	**94.31**
CIFAR-100	TinyImageNet	86.59	91.36	67.33	58.63	57.40	68.50	65.48	**94.24**
	Places-365	85.82	89.93	68.08	44.85	59.90	68.97	76.16	**91.20**

Table 4: F-measure comparison with DivideMix (DM) combined with OOD detection methods. 50% sym. IND noise is injected into training set, 20k and 10k OOD samples are added into training set and test set, respectively.

IND dataset	OOD dataset	DM	MSP	ODIN	MD	Ours
CIFAR-10	CIFAR-100	0.632	0.698	0.681	0.635	**0.838**
	TinyImageNet	0.638	0.726	0.707	0.702	**0.875**
	Places-365	0.637	0.717	0.705	0.651	**0.887**
CIFAR-100	TinyImageNet	0.516	0.687	0.705	0.526	**0.773**
	Places-365	0.519	0.685	0.696	0.541	**0.731**

4.1. Learning with Open-World Noisy Data

To investigate the effectiveness of NGC, we test it under mixed IND and OOD label noise. In this setup, we report both classification and OOD detection performance to show that NGC can learn a good classifier and OOD detector simultaneously. We use CIFAR-10 and CIFAR-100 as IND datasets, and TinyImageNet and Places-365 as the OOD datasets. We first add 50% symmetric IND noise. Then, additional samples are randomly selected from the OOD datasets to form the training dataset. It is noted that the CIFAR-100 dataset is also used as one of the OOD datasets

when CIFAR-10 is treated as the IND dataset.

First, we present the classification performance in Table 2. We compare NGC with the cross-entropy baseline and three recent methods for LNL, i.e., ILON [41], RoG [20] and DivideMix [21].

ILON reweights samples based on the outlier measurement. RoG uses an ensemble of generative classifiers built from features extracted from multiple layers of the pretrained model. DivideMix is the state-of-the-art method for LNL. We report the results of DivideMix without ensemble for a fair comparison. The number of OOD samples in training datasets is set to either 10k or 20k. We can see that NGC and DivideMix significantly outperform the other three methods. On CIFAR-10, NGC achieves better or on par performance compared with DivideMix. On CIFAR-100, NGC obtains an average performance gain of ∼4%. This demonstrates the superiority of NGC in classification.

Next, we present the OOD detection performance using AUROC in Table 3 following [10] and open-set classification performance [2] using F-measure in Table 4 as the metric. Since different OOD detectors need particularly tuned probability thresholds ζ, for fair comparison, we search the best ζ for all methods. Noted that LOND has not been studied before, we hence combine one of the best LNL methods DivideMix with leading OOD detectors including

Table 5: Test accuracy (%) under controlled IND label noise compared with state-of-the-art methods on CIFAR-10 and CIFAR-100 datasets. We run our method three times with different random seeds and report the mean and the standard deviation. Results for baseline methods are copied from [21, 23]

Dataset	CIFAR-10					CIFAR-100			
Noise type	Sym.				Asym.	Sym.			
Noise level	20%	50%	80%	90%	40%	20%	50%	80%	90%
Cross-Entropy	82.7	57.9	26.1	16.8	85.0	61.8	37.3	8.8	3.5
F-correction [34]	83.1	59.4	26.2	18.8	87.2	61.4	37.3	9.0	3.4
Co-teaching+ [53]	88.2	84.1	45.5	30.1	-	64.1	45.3	15.5	8.8
Mixup [55]	92.3	77.6	46.7	43.9	-	66.0	46.6	17.6	8.1
P-correction [51]	92.0	88.7	76.5	58.2	88.5	68.1	56.4	20.7	8.8
Meta-Learning [22]	92.0	88.8	76.1	58.3	89.2	67.7	58.0	40.1	14.3
M-correction [1]	93.8	91.9	86.6	68.7	87.4	73.4	65.4	47.6	20.5
DivideMix [21]	95.0	93.7	**92.4**	74.2	91.4	74.8	72.1	57.6	29.2
ProtoMix [23]	95.8	94.3	**92.4**	75.0	**91.9**	79.1	74.8	57.7	29.3
Ours	**95.88**±**0.13**	**94.54**±**0.35**	91.59±0.31	**80.46**±**1.97**	90.55±0.29	**79.31**±**0.35**	**75.91**±**0.39**	**62.70**±**0.37**	**29.76**±**0.85**

MSP [10], ODIN [27] and Mahalanobis distance (MD) [19] for comparisons. We also compare with recent OOD detection methods, Rot [6, 11] and SSD [38], which cannot be simply combined with DivideMix and need separate training. From the results, it can be seen that most comparison methods perform significantly worse than NGC. In terms of AUROC, NGC obtains performance gains over 17.2% on CIFAR-10 and 1.27% on CIFAR-100. Regarding F-measure, NGC outperforms other methods by at least 14% on CIFAR-10 and 3.5% on CIFAR-100. In supplementary material, we conduct comprehensive comparisons with another recent method for LNL, i.e., ProtoMix [23], due to limited space. We also provide further analysis to show that our method is robust to the selection of ζ.

4.2. Learning with Closed-World Noisy Labels

In addition to the LOND task, we test NGC in the conventional closed-world noisy label setup. We conduct experiments under controlled IND noise using the CIFAR-10 and CIFAR-100 datasets. To validate the efficacy of NGC, we compare it with many existing methods, including Meta-Learning [22], P-correction [51], M-correction [1], DivideMix [21], and ProtoMix [23]. Following commonly used LNL setups [1, 21], we run algorithms under asymmetric noise and symmetric noise with different noise levels. The noise level for symmetric noise ranges from 20% to 90% where it consists of randomly selecting labels for a percentage of the training data using all possible labels (i.e., the true label could be randomly retained). The noise level for asymmetric noise is set to 40%.

As Table 5 shown, in most cases, our method outperforms recent methods particularly designed for closed-world noisy label problems. This indicates the superiority and robustness of NGC.

4.3. Learning from Real-World Noisy Dataset

We test the performance of our method on real-world dataset WebVision [25] which contains noisy-labeled images collected from Flickr and Google. Similar to previous work [21], we perform experiments on the first 50 classes.

Table 6: Accuracy (%) on WebVision-50 and ILSVRC2012 validation sets. Results of baselines are from [3, 21, 28].

Method	WebVision		ILSVRC12	
	top-1	top-5	top-1	top-5
F-correction [34]	61.12	82.68	57.36	82.36
Decoupling [31]	62.54	84.74	58.26	82.26
D2L [30]	62.68	84.00	57.80	81.36
MentorNet [14]	63.00	81.40	57.80	79.92
Co-teaching [8]	63.58	85.20	61.48	84.70
Iterative-CV [3]	65.24	85.34	61.60	84.98
DivideMix [21]	77.32	91.64	**75.20**	90.84
ELR+ [28]	77.78	91.68	70.29	89.76
Ours	**79.16**	**91.84**	74.44	**91.04**

We report comparison results in Table 6, measuring top-1 and top-5 accuracy on WebVision validation set and ImageNet ILSVRC12 validation set. NGC consistently outperforms competing methods in most cases, which verifies the efficacy of our method on real-world noisy label task.

4.4. Ablation Studies and Discussion

To better understand NGC, we examine the impact of each component of NGC in Table 7. It can be observed that all components contribute to the efficacy of NGC. In particular, the two strategies in subgraph selection and the

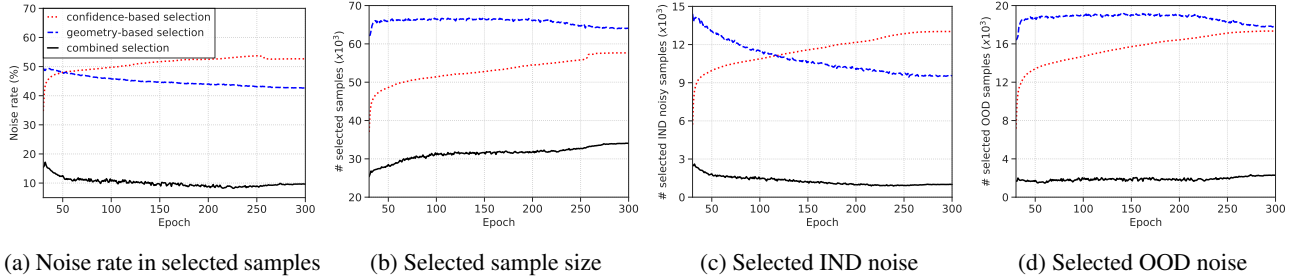| (a) Noise rate in selected samples | (b) Selected sample size | (c) Selected IND noise | (d) Selected OOD noise |

Figure 4: Analysis of subgraph selection under 50% IND noise (CIFAR-100) and 20k OOD noise (Places-365).

Table 7: Ablation study. GNC denotes graph-based noise correction. CS denotes confidence-based selection and GS denotes graph-based selection. For experiments whose noise type is OOD, Places-365 is used as OOD dataset and 50% sym. IND noise is injected into training set.

Dataset	CIFAR-10			CIFAR-100		
Noise type	OOD	Sym.	Asym.	OOD	Sym.	
Noise level	20k	50%	40%	20k	50%	80%
w/o GNC	92.13	94.32	85.85	72.85	74.20	55.56
w/o CS	87.20	92.44	89.68	63.78	73.22	37.82
w/o GS	86.55	85.59	81.17	65.34	67.18	35.16
w/o $\mathcal{L}^{inst}$	92.45	94.02	82.67	71.38	73.30	51.59
w/o $\mathcal{L}^{subgraph}$	70.39	85.12	79.17	55.12	58.06	41.42
w/o mixup	89.51	90.73	84.24	66.93	68.06	42.59
w/o AugMix	93.62	94.53	89.39	71.49	75.18	61.75
Ours	**93.67**	**94.54**	**90.55**	**73.44**	**75.91**	**62.70**

subgraph-level contrastive learning serve as the most important parts in our framework, without which the performance deteriorates severely. The observations validate that confidence-based (CS) and geometry-based selection (GS) can exploit neighborhood information from graph structure effectively. As a result, the test accuracy and OOD detection performance also improve as shown in Figure 5a, demonstrating the good generalization ability of our method. In supplementary material, we also demonstrate the robustness of our method to hyperparameters, i.e., η in Eq. (3) and k which is used to construct the k-NN graph.

Discussion on subgraph selection. To further examine the effect of the two subgraph selection strategies, we investigate the impact of each one for selecting clean samples. In Figure 4a and Figure 4b, we can see that the noise rate in the selected data by performing each strategy alone is significantly larger than the combined strategy. Figure 4c and Figure 4d further show that both IND and OOD noise can be drastically removed by the combined strategy, while merely using one of them has little effect. This is because the confidence-based selection can degrade the connectivity

between clean samples and noisy samples such that samples in the largest connected component are clean. Moreover, we divide the nodes into three parts: clean data, IND noise and OOD noise, and analyze the average degrees of nodes in each part after performing confidence-based selection. As demonstrated in Figure 5b, we find that as the training process progresses, the average node degree of OOD noisy samples is decreasing, while the average degree of clean samples is increasing. This further validates that confidence-based strategy facilitates the selection of the largest connected component in geometry-based strategy.
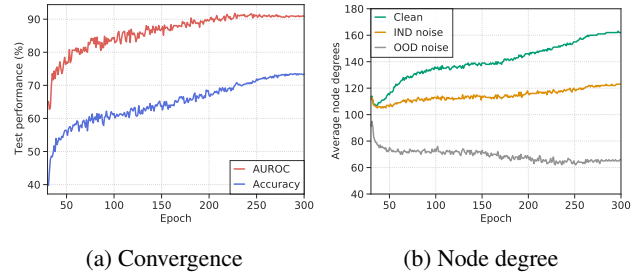

| (a) Convergence | (b) Node degree |

Figure 5: Visualization for convergence of our method and node degrees under 50% IND noise (CIFAR-100) and 20k OOD (Places-365).

5. Conclusion

In this paper, we study a realistic problem where the training dataset contains both IND and OOD noise, and the presence of OOD samples at test time. To address this problem, we introduce a noisy graph cleaning framework that simultaneously performs noise correction and clean data selection based on prediction confidence and geometric structure of data in latent feature space. NGC outperforms many existing methods on different datasets with varying degrees of noise. Our work may motivate researchers in two directions: learning from IND and OOD noisy data is worth further exploration due to its broad range of applications and OOD detection from weakly-labeled datasets is promising.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin Mcguinness. Unsupervised label noise modeling and loss correction. In *ICML*, pages 312–321, 2019.

[2] Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *CVPR*, pages 1563–1572, 2016.

[3] Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *ICML*, pages 1062–1070, 2019.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020.

[5] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *NeurIPS*, pages 4878–4887, 2017.

[6] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *NeurIPS*, pages 9781–9791, 2018.

[7] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R. Scott, and Dinglong Huang. Curriculumnet: Weakly supervised learning from large-scale web images. In *ECCV*, pages 139–154, 2018.

[8] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W. Tsang, and Masashi Sugiyama. Coteaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pages 8536–8546, 2018.

[9] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep selflearning from noisy labels. In *ICCV*, pages 5137–5146, 2019.

[10] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.

[11] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*, pages 15637–15648, 2019.

[12] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *ICLR*, 2020.

[13] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *CVPR*, pages 5070–5079, 2019.

[14] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pages 2304–2313, 2018.

[15] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, pages 18661–18673, 2020.

[16] Youngdong Kim, June Yim, Juseung Yun, and Junmo Kim. NLNL: negative learning for noisy labels. In *ICCV*, pages 101–110, 2019.

[17] Samuli Laine and Timo Aila. Temporal ensembling for semisupervised learning. In *ICLR*, 2017.

[18] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*, 2018.

[19] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, pages 7167–7177, 2018.

[20] Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. In *ICML*, pages 3763–3772, 2019.

[21] Junnan Li, Richard Socher, and Steven CH Hi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, 2020.

[22] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S. Kankanhalli. Learning to learn from noisy labeled data. In *CVPR*, pages 5051–5059, 2019.

[23] Junnan Li, Caiming Xiong, and Steven Hoi. Learning from noisy data with robust representation learning, 2021.

[24] Junnan Li, Caiming Xiong, and Steven CH Hoi. Mopro: Webly supervised learning with momentum prototypes. In *ICLR*, 2021.

[25] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *CoRR*, abs/1708.02862, 2017.

[26] Yu-Feng Li and De-Ming Liang. Lightweight label propagation for large-scale network data. *IEEE Transactions on Knowledge and Data Engineering*, 33(5):2071–2082, 2021.

[27] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.

[28] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *NeurIPS*, pages 20331–20342, 2020.

[29] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE TPAMI*, 38(3):447–461, 2016.

[30] Xingjun Ma, Yisen Wang, Michael E. Houle, Shuo Zhou, Sarah M. Erfani, Shu-Tao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *ICML*, pages 3355–3364, 2018.

[31] Eran Malach and Shai Shalev-Shwartz. Decoupling "when to update" from "how to update". In *NeurIPS*, pages 960–970, 2017.

[32] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. SELF: learning to filter noisy labels with self-ensembling. In *ICLR*, 2020.

[33] Kento Nishi, Yi Ding, Alex Rich, and Tobias Höllerer. Augmentation strategies for learning with noisy labels. In *CVPR*, pages 8022–8031, 2021.

[34] Giorgio Patrini, Alessandro Rozza, Aditya K. Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pages 2233–2241, 2017.

[35] Geoff Pleiss, Tianyi Zhang, Ethan R. Elenberg, and Kilian Q. Weinberger. Identifying mislabeled data using the area under the margin ranking. In *NeurIPS*, pages 17044–17056, 2020.

[36] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, pages 4334–4343, 2018.

[37] Ragav Sachdeva, Filipe R Cordeiro, Vasileios Belagiannis, Ian Reid, and Gustavo Carneiro. Evidentialmix: Learning with combined open-set and closed-set noisy labels. In *WACV*, pages 3607–3615, 2021.

[38] Vikash Sehwag, Mung Chiang, and Prateek Mittal. SSD: A unified framework for self-supervised outlier detection. In *ICLR*, 2021.

[39] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, pages 5552–5560, 2018.

[40] Xiaobo Wang, Shuo Wang, Hailin Shi, Jun Wang, and Tao Mei. Co-mining: Deep face recognition with noisy labels. In *ICCV*, pages 9357–9366, 2019.

[41] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. In *CVPR*, pages 8688–8696, 2018.

[42] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *ICCV*, pages 322–330, 2019.

[43] Tong Wei, Lan-Zhe Guo, Yu-Feng Li, and Wei Gao. Learning safe multi-label prediction for weakly labeled data. *Machine Learning*, 107(4):703–725, 2018.

[44] Tong Wei and Yu-Feng Li. Does tail label help for large-scale multi-label learning? *IEEE Transaction Neural Networks Learning Systems*, 31(7):2315–2324, 2020.

[45] Pengxiang Wu, Songzhu Zheng, Mayank Goswami, Dimitris N. Metaxas, and Chao Chen. A topological filter for learning with label noise. In *NeurIPS*, pages 21382–21393, 2020.

[46] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018.

[47] Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *ICLR*, 2021.

[48] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. In *NeurIPS*, pages 7597–7610, 2020.

[49] Jingkang Yang, Litong Feng, Weirong Chen, Xiaopeng Yan, Huabin Zheng, Ping Luo, and Wayne Zhang. Webly supervised image classification with self-contained confidence. In *ECCV*, pages 779–795, 2020.

[50] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual T: reducing estimation error for transition matrix in label-noise learning. In *NeurIPS*, pages 7260–7271, 2020.

[51] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *CVPR*, pages 7017–7025, 2019.

[52] Qing Yu and Kiyoharu Aizawa. Unsupervised out-of-distribution detection by maximum classifier discrepancy. In *ICCV*, pages 9517–9525, 2019.

[53] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W. Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *ICML*, pages 7164–7173, 2019.

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

[55] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *ICLR*, 2017.

[56] Zizhao Zhang, Han Zhang, Sercan O Arik, Honglak Lee, and Tomas Pfister. Distilling effective supervision from severe label noise. In *CVPR*, pages 9294–9303, 2020.

[57] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NeurIPS*, pages 321–328, 2003.

[58] Xiaojin Zhu, John Lafferty, and Ronald Rosenfeld. *Semi-supervised learning with graphs*. PhD thesis, Carnegie Mellon University, 2005.