Open-set Label Noise Can Improve Robustness Against Inherent Label Noise

Anonymous Author(s)
Affiliation
Address
email

Abstract

Learning with noisy labels is a practically challenging problem in weakly supervised learning. In the existing literature, open-set noises are always considered to be poisonous for generalization, similar to closed-set noises. In this paper, we empirically show that open-set noisy labels can be non-toxic and even benefit the robustness against inherent noisy labels. Inspired by the observations, we propose a simple yet effective regularization by introducing Open-set samples with Dynamic Noisy Labels (ODNL) into training. With ODNL, the extra capacity of the neural network can be largely consumed in a way that does not interfere with learning patterns from clean data. Through the lens of SGD noise, we show that the noises induced by our method are random-direction, conflict-free and biased, which may help the model converge to a flat minimum with superior stability and enforce the model to produce conservative predictions on Out-of-Distribution instances. Extensive experimental results on benchmark datasets with various types of noisy labels demonstrate that the proposed method not only enhances the performance of many existing robust algorithms but also achieves significant improvement on Out-of-Distribution detection tasks even in the label noise setting.

1 Introduction

The success of deep neural networks (DNNs) heavily relies on a large number of training instances with fully accurate labels [8]. In real-world applications, it is expensive and time-consuming to collect such large-scale datasets. To alleviate this problem, some surrogate methods are commonly used to improve labelling efficiency, such as online queries [2] and crowdsourcing [42]. These cheap but imperfect methods usually suffer from unavoidable noisy labels that can be easily memorized by Deep Neural Networks (DNNs), leading to poor generalization performance [1, 45]. Therefore, designing robust algorithms against noisy labels is of great practical importance.

Label noise can be separated into two categories: 1) closed-set noise, where instances with noisy labels have true class labels within the noisy label set [7, 13, 27, 28, 32, 35]. 2) open-set noise, where instances with noisy labels have some true class labels outside the noisy label set [17, 30, 33, 37]. In the existing literature, open-set noises are always considered to be harmful to the training of DNNs like closed-set noises [17, 33, 30, 37]. For example, ILON [33] reduces the effect of open-set noises by using noisy label detector and contrastive loss to pull away noisy samples from clean samples. Extended T [37] uses an extended transition matrix to mitigate the impacts of open-set noises and EvidentialMix [30] filters out the open-set examples by modelling the loss distribution. Therefore, there arises a question to be answered: Do open-set noises always play a negative role in the training?

In this work, we answer this question by experiments with an open-set auxiliary dataset. We empirically show that additional open-set noises can be harmless to generalization if the number of open-set noises is sufficiently large. More surprisingly, we observe that the open-set noises can

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

even benefit the robustness of neural networks against noisy labels from the original training dataset, while the closed-set counterpart could not. Based on the "insufficient capacity" hypothesis [1], an intuitive explanation for this phenomenon is that increasing the number of open-set auxiliary samples slows down the fitting of inherent noises. In other words, fitting open-set label noises consumes extra capacity of the network, thereby reducing the memorization of inherent noisy labels. The details of experimental results are presented in Subsection 2.2.

Inspired by the above observations, we propose an effective and complementary method to improve robustness against noisy labels. The high-level idea is to introduce Open-set samples with Dynamic Noisy Labels (ODNL) into training. In each epoch, we assign random labels to instances of open-set auxiliary dataset by uniformly sampling from the label set. In other words, the label of each open-set sample is independently random and is constantly changing over training epochs. In this manner, ODNL would consistently produce "benign hard examples" to consume the extra representation capacity of neural networks, thereby preventing the neural network from overfitting inherent noisy labels.

To further understand the effect of ODNL, we provide an analysis from the lens of SGD noise [5, 36], and formalize the differences and connections between our regularization and related methods. From the analysis, we show that the SGD noises induced by our method are 1) random-direction, helping the model converge to a flat minimum; 2) conflict-free, leading to good stability for the training; 3) biased, enforcing the model to produce more conservative predictions on OOD examples. In this manner, our method not only improves the robustness against noisy labels, but also provides benefits for OOD detection performance, which is also essential for the deployment of deep learning in safety-critical applications.

To the best of our knowledge, we are the first to explore the benefits of open-set auxiliary dataset in learning with noisy labels. To verify the efficacy of our regularization, we conduct extensive experiments on both simulated and real-world noisy datasets, including CIFAR-10, CIFAR-100 and Clothing1M datasets. In summary, the proposed method can be easily adapted for a wide range of existing algorithms to prevent overfitting noisy labels. Furthermore, experimental results validate that our regularization could also achieve impressive performance for detecting OOD examples, even if the labels of training dataset are noisy.

2 Problem setting and motivation

In this section, we first briefly introduce the problem setting of learning from training data with noisy labels. Then we start in Subsection 2.2 with an observational study highlighting the effect of open-set noisy examples from the auxiliary dataset on generalization and robustness against noisy labels.

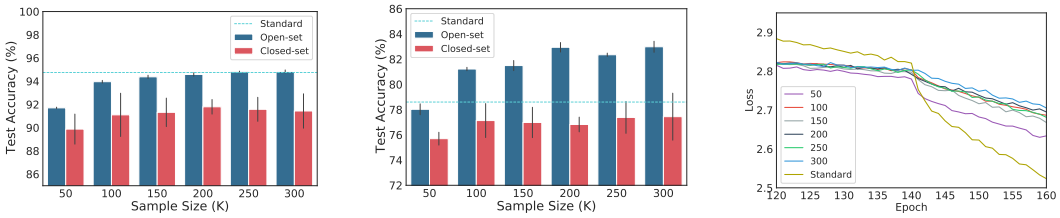
2.1 Preliminaries: learning with inherent noisy labels

In this work, we consider multi-class classification with k classes. Let $\mathcal{X} \subset \mathbb{R}^d$ be the feature space and $\mathcal{Y} = \{1, \ldots, k\}$ be the label space, we suppose the training dataset with N samples is given as $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathcal{X}$ is the i-th instance sampled from a certain data-generating distribution $\mathcal{D}_{\text{train}}$ in an i.i.d. manner and $y_i \in \{1, \ldots, k\}$ represents its observed label that might be different from the ground truth. To prevent confusion, the noisy labels in the training dataset are referred to as *inherent noisy labels*. A classifier is a function that maps the feature space to the label space $f : \mathcal{X} \to \mathbb{R}^k$ with trainable parameter $\boldsymbol{\theta} \in \mathbb{R}^p$. In this paper, we consider the common case where the function f is a DNN with the softmax output layer. The objective function of standard training can be represented as a cross-entropy loss:

$$\mathcal{L}_{CCE} = \mathbb{E}_{\mathcal{D}_{train}} \left[\mathcal{L} \left(f(\boldsymbol{x}; \boldsymbol{\theta}), y_{\boldsymbol{x}} \right) \right] = -\frac{1}{N} \sum_{i=1}^{N} \boldsymbol{e}^{y_i} \log f \left(\boldsymbol{x}_i; \boldsymbol{\theta} \right), \tag{1}$$

where $f\left(\boldsymbol{x}_i; \boldsymbol{\theta}\right)$ denotes the output of the classifier f with $\boldsymbol{\theta}$ for $\boldsymbol{x}_i$ and $\boldsymbol{e}^{y_i} := (0, \ldots, 1, \ldots, 0)^\top \in \{0, 1\}^k$ is a one-hot vector and only the y_i-th entry of $\boldsymbol{e}^{y_i}$ is 1.

In addition to the training data, we consider there is an unlabelled auxiliary dataset consisting of open-set instances $\{\widetilde{\boldsymbol{x}}_i\}_{i=1}^M$. Specifically, open-set instances are sampled from $\mathcal{D}_{out}$, disjoint from $\mathcal{D}_{\text{train}}$. Therefore, they are also called Out-of-Distribution (OOD) instances. In real applications, it is easy to get such an auxiliary dataset, which is commonly used in OOD detection tasks [10, 26].



(a) Learning with Clean data (b) Learning with inherent noises (c) Losses of inherent noises

Figure 1: Average test accuracy (%) on CIFAR-10 with (a) clean labels (b) inherent noisy labels under different auxiliary dataset sizes. We use "open-set" and "closed-set" to denote two types of auxiliary datasets. The results of standard training are also reported for comparison. All experiments are repeated five times and the standard deviations are presented as error bars. (c) Training losses of inherent noises around the 140th epoch under different auxiliary dataset sizes (K).

2.2 Motivation: the role of open-set label noise in generalization and robustness

To analyze the effects of open-set noisy labels, we build an open-set noisy dataset with the unlabelled auxiliary dataset. For each open-set instance, we assign a fixed random label $\widetilde{y}_i$ uniformly sampled from the label set $\{1, \ldots, k\}$. In each iteration, we concatenate the two batches that are sampled from the training dataset and the open-set noisy dataset respectively, then perform standard training. In addition, we conduct controlled experiments by replacing the open-set dataset with a closed-set dataset to verify the effectiveness of open-set noisy dataset during the training. To demonstrate the importance of the sample size of the given auxiliary dataset, we compare the performance of training with different-sized auxiliary datasets, built by randomly sampling from the original auxiliary dataset.

Throughout this subsection, we perform training with WRN-40-2 [44] on CIFAR-10 [16] using standard cross-entropy loss. For experiments under inherent label noise, we use symmetric noise with 40% noisy rate. For auxiliary dataset, we use 80 Million Tiny Images [31] as the open-set auxiliary dataset and CIFAR-5m[1] [11, 23] as the closed-set auxiliary dataset. Specifically, we remove all examples of 80 Million Tiny Images that appear in the CIFAR-10/100 datasets, so that $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{out}}$ are disjoint. More training details can be found in Appendix C.

Learning with clean labels. Fig. 1a presents the results on clean training set of CIFAR-10, training with the open/closed-set noisy auxiliary dataset. The results show that open-set noisy labels from the auxiliary dataset become harmless gradually with the increase of the sample size of the auxiliary dataset, while closed-set noisy labels consistently deteriorate the generalization performance. Furthermore, we observe that closed-set noisy labels result in unstable training shown by the large standard deviations in the figure. It is worthy to note that open-set noisy labels do not encounter this issue.

Learning with inherent noisy labels. Fig. 1b presents the results on CIFAR-10 with inherent noisy labels, training with the open/closed-set noisy auxiliary dataset. Surprisingly, we can observe that as we increase the sample size of the open-set noisy dataset, the open-set noises can even improve the generalization performance beyond standard training without the noisy auxiliary dataset. In this case, closed-set noisy auxiliary dataset still plays a negative role in the training.

An intuitive interpretation. From the results, one can hypothesize that open-set noisy labels in the auxiliary dataset would be harmless and can even benefit the robustness against inherent noisy labels, if the sample size of the auxiliary dataset is large enough. This phenomenon is somewhat counter-intuitive since one would expect that open-set noisy labels should have been "poison" to the training of network, like closed-set noises [17, 30, 33, 37]. Here we provide an intuitive interpretation from "insufficient capacity" [1]: increasing the number of examples while keeping representation capacity fixed would increase the time needed to memorize the data set. Hence, the larger the size of auxiliary dataset is, the more time it needs to memorize the open-set noises in the auxiliary dataset as well as the inherent noises in the training set, relative to clean data. This interpretation is supported by our plot of training losses of inherent noises with different-sized auxiliary datasets in Fig. 1c, where increasing the number of open-set auxiliary samples slows down the fitting of inherent noises.

[1]The dataset is published on `https://github.com/preetum/cifar5m`.

3 Method: Open-set regularization with Dynamic Noisy Labels

Motivated by the previous observations, we propose to utilize open-set auxiliary dataset to further improve the robustness against inherent noisy labels. This approach can be considered as an auxiliary task that continuously consumes the extra capacity of neural networks but does not interfere with learning patterns from clean data. Such auxiliary tasks will prevent deep networks from overfitting noisy data.

Dynamic Noisy Labels. To maintain the effectiveness of the auxiliary task during training, an ideal method is to use an auxiliary dataset with unlimited instances based on "insufficient capacity" [1]. In this way, the network would always try to fit a stream of new instances sequentially in an online learning setting. However, it is unrealistic to provide such an auxiliary dataset in real scenarios.

To relieve the requirement on sample size, we propose an alternative approach called Dynamic Noisy Labels. For each open-set instance $\widetilde{\boldsymbol{x}}_i$, a noisy label $\widetilde{y}_i$ is uniformly sampled from the label set $\{1, \ldots, k\}$ independently, regardless of the instance $\widetilde{\boldsymbol{x}}_i$. More importantly, the generated labels are not fixed and would be changed dynamically over training epochs. In this way, the auxiliary task becomes an "impossible mission", continuously consuming the extra capacity of network. Without Dynamic Noisy Labels, the effectiveness of the auxiliary task would be degraded gradually during training, because deep network can easily memorize the open-set samples, even though the labels of those samples are randomly generated [1, 45].

Training Objective. For the original training dataset, we use the standard cross entropy as the training objective function:

$$\mathcal{L}_1 = \mathbb{E}_{\mathcal{D}_{\text{train}}} \left[\ell \left(f(\boldsymbol{x}; \boldsymbol{\theta}), y \right) \right] = \mathbb{E}_{\mathcal{D}_{\text{train}}} \left[-\boldsymbol{e}^y \log f\left(\boldsymbol{x}; \boldsymbol{\theta}\right) \right], \tag{2}$$

For the auxiliary dataset, we enforce the outputs to be consistent with dynamic noise labels, uniformly sampled from the label set in each epoch. Therefore, the training objective function is as follows:

$$\mathcal{L}_2 = \mathbb{E}_{\mathcal{D}_{\text{out}}} \left[\ell \left(f(\widetilde{\boldsymbol{x}}; \boldsymbol{\theta}), \widetilde{y} \right) \right] = \mathbb{E}_{\mathcal{D}_{\text{out}}} \left[-\boldsymbol{e}^{\widetilde{y}} \log f\left(\widetilde{\boldsymbol{x}}; \boldsymbol{\theta}\right) \right], \text{where } \widetilde{y} \sim \mathcal{U}_k \tag{3}$$

where $\mathcal{U}_k$ represents a discrete uniform distribution on the label set $\{1, \ldots, k\}$.

Combining the original training dataset and the auxiliary dataset in the training, now we can formalize ODNL as minimizing the final objective:

$$\mathcal{L}_{total} = \mathcal{L}_1 + \eta \cdot \mathcal{L}_2 = \mathbb{E}_{\mathcal{D}_{\text{train}}} \left[\ell \left(f(\boldsymbol{x}; \boldsymbol{\theta}), y \right) \right] + \eta \cdot \mathbb{E}_{\mathcal{D}_{\text{out}}} \left[\ell \left(f(\widetilde{\boldsymbol{x}}; \boldsymbol{\theta}), \widetilde{y} \right) \right] \tag{4}$$

where η denotes the trade-off hyperparameter. The details of ODNL are provided in Algorithm 1.

Extensions to Other Robust Training Algorithms. It is worthy to note that our regularization is a general method and can be easily incorporated into existing robust training algorithms, including sample selection[7, 35], loss correction [27], robust loss function [46, 34], etc. Given the original learning objective $\mathcal{L}_{\text{robust}}$ of the robust methods and the hyperparameter η, we can formalize the final objective as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{robust}} + \eta \cdot \mathcal{L}_2 \tag{5}$$

Algorithm 1 Open-set Regularization with Dynamic Noisy Labels

Input: Training dataset $\mathcal{D}_{\text{train}}$. Open-set auxiliary dataset $\mathcal{D}_{\text{out}}$;
1: **for** each iteration **do**
2: Sample a mini-batch of original training data $\{(\boldsymbol{x}_i, y_i)\}_{i=0}^{n}$ from $\mathcal{D}_{\text{train}}$;
3: Sample a mini-batch of open-set data $\{\widetilde{\boldsymbol{x}}_i\}_{i=0}^{m}$ from $\mathcal{D}_{\text{out}}$;
4: Generate random noisy label $\widetilde{y}_i \sim \mathcal{U}_k$ for each open-set data $\widetilde{\boldsymbol{x}}_i$;
5: Perform gradient descent on f with $\mathcal{L}_{\text{total}}$ from Equation (4);
6: **end for**

4 Understanding from SGD noise

To further understand the effect of our regularization, we provide an analysis from the lens of SGD noise. Following SLN [5], here we consider an original training sample $(\boldsymbol{x}, y)$ and an open-set sample

$\widetilde{\boldsymbol{x}}$ at each SGD step for convenience. In parameter updates, our regularization adds noises to the original gradients with the open-set noisy sample $\widetilde{\boldsymbol{x}}$ and dynamic noisy label $\widetilde{y} \sim \mathcal{U}_k$:

$$\widetilde{\nabla}_{\boldsymbol{\theta}} \ell_{\text{total}} = \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}, y) + \eta \cdot \nabla_{\boldsymbol{\theta}} \ell(\widetilde{\boldsymbol{x}}, \widetilde{y}) \tag{6}$$

where ℓ denotes the cross entropy loss for single sample. For convenience, we omit the hyperparameter η and use $\boldsymbol{f}, \widetilde{\boldsymbol{f}}$ to indicate the model output $f(\boldsymbol{x}, \boldsymbol{\theta})$ for the original training sample and the model output $f(\widetilde{\boldsymbol{x}}, \boldsymbol{\theta})$ for the open-set noisy sample. Following the standard notation of the Jacobian matrix, we have $\nabla_{\boldsymbol{\theta}} \ell \in \mathbb{R}^{1 \times \dot{p}}, \nabla_{\boldsymbol{f}} \ell \in \mathbb{R}^{1 \times k}, \nabla_{\boldsymbol{\theta}} \boldsymbol{f} \in \mathbb{R}^{k \times p}, \nabla_{\boldsymbol{\theta}_i} \boldsymbol{f} \in \mathbb{R}^k$ and $\nabla_{\boldsymbol{\theta}} \boldsymbol{f}_i \in \mathbb{R}^{1 \times p}$. The proofs of the following propositions are presented in Appendix B.

Proposition 1 *For the cross-entropy loss, Eq. (6) induces noise* $\boldsymbol{z} = -\frac{\nabla_{\boldsymbol{\theta}} \widetilde{\boldsymbol{f}}_j}{\widetilde{\boldsymbol{f}}_j}$ *on* $\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}, y)$*, s.t.,* $\boldsymbol{z} \in \mathbb{R}^p, j \sim \mathcal{U}_k$*, where* $\frac{\cdot}{\widetilde{\boldsymbol{f}}_j}$ *denotes the element-wise division. Note that the expected value of noise on the* i*-th parameter* $\boldsymbol{\theta}_i$ *is* $-\frac{1}{k} \sum_{j=1}^{k} \frac{\nabla_{\boldsymbol{\theta}_i} \widetilde{\boldsymbol{f}}_j}{\widetilde{\boldsymbol{f}}_j}$.

The effects of SGD noise. From Proposition 1, we find that our regularization induces SGD noise sampled from a uniform distribution over $\left\{-\nabla_{\boldsymbol{\theta}} \widetilde{\boldsymbol{f}}_j / \widetilde{\boldsymbol{f}}_j\right\}_{j=1}^{k}$. The effects of the noises can be analyzed from three aspects:

- Firstly, our regularization yields SGD noises with random direction, uniformly sampled from the gradients of the open-set noisy sample. In this way, the SGD noises make the training difficult to converge and help escape local minima [3, 12, 14, 15, 25]. Without the noises, the model would quickly converge to a local minimum because the optimization of parameters always follows the direction of gradient descent during the training.
- Secondly, the noises z are independently random to the gradients $\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}, y)$ from the original training example as the noises are conducted with the open-set sample $\widetilde{\boldsymbol{x}}$ from $\mathcal{D}_{out}$. If closed-set samples are used here, the generated noises might be conflicted with the original gradients, leading to unstable training and limiting the improvement from noises. This argument is clearly supported by the experimental results in Fig. 1 and Fig. 3a.
- Finally, the expected value of noise is not zero so the noise is biased. In particular, the expected value of noise enforces the model f to produce more conservative predictions on Out-of-Distribution samples. Consequently, this feature would lead to performance improvement on OOD detection tasks as shown in Subsection 5.4.

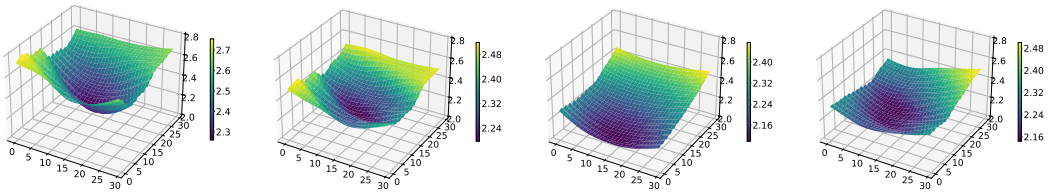
To further demonstrate the importance of open-set samples in the auxiliary dataset, we conduct experiments with synthetic auxiliary datasets consisting of convex combinations $\boldsymbol{x}'$ of open-set samples $\widetilde{\boldsymbol{x}}$ and closed-set samples $\boldsymbol{x}$ with a hyperparameter α: $\boldsymbol{x}' = (1 - \alpha) \cdot \widetilde{\boldsymbol{x}} + \alpha \cdot \boldsymbol{x}$. Here, we use 80 Million Tiny Images [31] and CIFAR-5m [11, 23] as the open-set and closed-set dataset. More training details can be found in Appendix C. The results in Figure. 3a show that using open-set auxiliary dataset achieves better performance on robustness against inherent noises.

Relations to label randomization methods. Label randomization is also applied in some existing work in the context of deep learning[5, 20, 40]. For example, DisturbLabel [40] intentionally generates incorrect labels on a small fraction of training data to improve generalization performance under the settings of clean labels, interpreted as an implicit model ensemble. It can be seen as using a part of training data as a closed-set auxiliary dataset. In such a way, DisturbLabel can be interpreted as a special case of our method with a closed-set auxiliary dataset.

To mitigate the issue of label noise, Stochastic Label Noise (SLN) [5] introduces a variant of SGD noise induced by adding dynamic and mean-zero perturbations on the labels of training dataset, while Label Smoothing [20] uses a fixed and biased perturbation on the label. The noise gradients induced by SLN are given as follows:

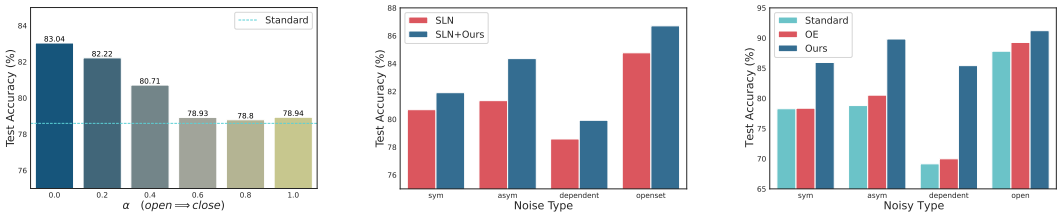
$$\tilde{\nabla}_{\boldsymbol{\theta}} \ell(\boldsymbol{f}, y) = \nabla_{\boldsymbol{\theta}} \ell\left(\boldsymbol{f}, \boldsymbol{e}^y + \sigma_y \boldsymbol{z}_y\right) \tag{7}$$

where $\sigma_y > 0, \boldsymbol{z}_y \in \mathcal{R}^k$, $\boldsymbol{z}_y \sim \mathcal{N}\left(0, \boldsymbol{I}_{k \times k}\right)$.



(a) Standard (b) OE (c) SLN (d) Ours

Figure 2: Loss landscapes around the local minimum of models trained on CIFAR-10 with symmetric-40% noisy labels. We compare their sharpness with the same-scale z-axis and show the loss distribution by drawing color bars separately. We show that our method and SLN lead to a flat minimum, while the models trained by Standard and OE converge to a sharp minimum.



(a) The importance of open set (b) Our method can improve SLN (c) Compare with OE.

Figure 3: (a) Test accuracy (%) of ODNL ($\eta = 1$) on CIFAR-10 with symmetric-40% inherent noisy labels using different synthetic auxiliary datasets: the larger the α is, the closer the auxiliary sample distribution $\mathcal{D}_{\text{out}}$ is to the origin sample distribution $\mathcal{D}_{\text{train}}$. (b) Comparison of performances on CIFAR-10 with different types of inherent noisy labels using SLN and SLN+Ours. Here we follow the settings of SLN and the training details are provided in Appendix C. (c) Comparison of performances on CIFAR10 with different types of inherent noisy labels using OE and Ours.

Proposition 2 (Proposition 3 in Chen et al. [5]) *For the cross-entropy loss, Eq. (7) in SLN induces noise* $\boldsymbol{z} \sim \mathcal{N}(0, \sigma_y^2 M)$ *on* $\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{f}, y)$*, s.t.,* $\boldsymbol{M} \in \mathbb{R}^{p \times p}$ *and* $\boldsymbol{M}_{i,j} = \left(\frac{\nabla_{\boldsymbol{\theta}_i} \boldsymbol{f}}{\boldsymbol{f}}\right)^T \frac{\nabla_{\boldsymbol{\theta}_j} \boldsymbol{f}}{\boldsymbol{f}}, \forall i, j \in \{1, \ldots, p\}$*, where* $\frac{\cdot}{\boldsymbol{f}_j}$ *denotes the element-wise division.*

From Proposition 2, we show that SLN induces SGD noises that obey a Gaussian distribution, where the standard deviation can be adjusted by the hyperparameter σ_y. Therefore, SLN and our ODNL can be considered as different variants of SGD noise, helping the network converge to a flat minimum, which is expected to provide better generalization performance [3, 12, 14, 15, 25]. The effects are explicitly presented in Fig. 2c and Fig. 2d. Moreover, the results in Fig. 3b illustrate that our method could further improve the performance of SLN[2] under various types of label noise, showing the complementarity between SLN and our method.

Relations to Outlier Exposure. Outlier Exposure (OE) [10] is an effective method designed for OOD detection tasks. Specifically, OE also makes use of a large, unlabeled auxiliary dataset sampled from $\mathcal{D}_{\text{out}}$, and regularizes the softmax probabilities to be a uniform distribution for out-of-distribution data in the auxiliary dataset. In such manner, OE train the model to discover signals and learn heuristics to detect whether a query is sampled from $\mathcal{D}_{\text{train}}$ or $\mathcal{D}_{\text{out}}$. The training objective is to minimize: $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{train}}} \left[\ell(f(\boldsymbol{x};\boldsymbol{\theta}), y)\right] + \lambda \cdot \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_{\text{out}}} \left[-\frac{1}{k} \cdot \sum_{i=1}^{k} \log f_i\left(\widetilde{\boldsymbol{x}};\boldsymbol{\theta}\right)\right]$.

Proposition 3 *For the cross-entropy loss, each open-set sample in OE induces bias* $\boldsymbol{z} = -\frac{1}{k} \sum_{j=1}^{k} \frac{\nabla_{\boldsymbol{\theta}} \widetilde{\boldsymbol{f}}_j}{\widetilde{\boldsymbol{f}}_j}$ *on* $\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{x}, y)$*, s.t.,* $\boldsymbol{z} \in \mathbb{R}^p$*, where* $\frac{\cdot}{\widetilde{\boldsymbol{f}}_j}$ *denotes the element-wise division.*

Compared Proposition 3 to Proposition 1, our method should have similar regularization effect to OE on OOD examples, because the expectation of the SGD noises from ODNL is equal to the bias

[2]We use the official implementation: `https://github.com/chenpf1025/SLN`.

Table 1: Average test accuracy (%) with standard deviation on CIFAR-10 under various types of noisy labels (over 5 trials). The bold indicates the improved results by integrating our regularization.

Method	Sym-20%	Sym-50%	Asymmetric	Dependent	Open-set
Standard	86.93±0.49	69.48±0.58	78.34±0.96	68.39±0.65	88.45±0.57
+ ODNL (Ours)	**89.94±0.70**	**81.20±0.68**	**86.91±0.32**	**82.42±0.36**	**91.12±0.39**
Decoupling	88.31±0.39	80.81±0.85	84.85±0.22	82.44±0.54	84.69±0.46
+ ODNL (Ours)	**89.27±0.90**	**81.33±0.77**	**87.03±0.76**	**84.14±0.86**	**86.34±0.32**
F-correction	84.95±0.27	72.11±0.15	84.46±0.96	72.44±0.58	85.54±0.86
+ ODNL (Ours)	**87.95±0.28**	**82.44±0.21**	**87.56±0.19**	**81.83±0.43**	**89.67±0.69**
PHuber-CE	86.69±0.31	70.05±0.55	78.10±0.44	70.36±1.12	88.09±0.14
+ ODNL (Ours)	**91.98±0.38**	**83.19±0.26**	**89.18±0.46**	**80.87±0.48**	**91.62±0.15**
Co-teaching	91.72±0.35	86.65±0.43	87.56±0.36	88.64±0.79	89.91±0.58
+ ODNL (Ours)	**92.80±0.75**	**89.64±0.81**	**88.39±0.39**	**91.36±0.23**	**92.60±0.47**
JoCoR ($\lambda = 0.5$)	92.68±0.33	88.36±0.64	84.06±0.37	90.11±0.19	88.80±0.54
+ ODNL (Ours)	**93.41±0.75**	**90.40±0.43**	**86.46±0.80**	**91.75±0.92**	**91.62±0.88**

induced by OE. However, OE cannot improve the robustness against noisy labels as it does not induce dynamic noises so that the optimization of parameters always follows the direction of gradient descent. The difference is empirically shown in Fig. 3c and we further verify it in Subsection 5.4.

5 Experiments

In this section, we verify the effectiveness of our method on three benchmark datasets: CIFAR-10/100 and Clothing1M. We show that ODNL could not only significantly improve the robustness against various types of inherent noisy labels in standard training, but also enhance the performance of existing robust training techniques. Then we perform a sensitivity analysis to validate the effect of η. Finally, we conduct experiments to evaluate the performance of ODNL on OOD detection task.

5.1 Setups

We comprehensively verify the utility of ODNL on different types of label noise, including symmetric noise, asymmetric noise[46], instance-dependent noise [4] and open-set noise [33] synthesized on CIFAR-10/100 and real-world noise on Clothing1M [39]. Symmetric noise assumes each label has the same probability of flipping to any other classes. We uniformly flip the label to other classes with an overall probability 20% and 50%. Asymmetric noise assumes labels might be only flipped to similar classes [5, 27, 46]. Noisy labels are generated by mapping TRUCK $\rightarrow$ AUTOMOBILE, BIRD $\rightarrow$ AIRPLANE, DEER $\rightarrow$ HORSE, and CAT $\leftrightarrow$ DOG with probability 40% for CIFAR-10. For CIFAR-100, we flip each class into the next circularly with probability 40%. Instance-dependent noise assumes the mislabeling probability is dependent on each instance's input features [4, 5, 38]. We use the instance-dependent noise from PDN [38] with a noisy rate 40%, where the noise is synthesized based on the DNN prediction error. Open-set noise contains samples that do not belong to the label set considered in the classification task [33]. We generate open-set noises on CIFAR-10 by randomly replacing 40% of its training images with images from CIFAR-100.

We perform training with WRN-40-2 [44] on CIFAR-10 and CIFAR-100. The network is trained for 200 epochs using SGD with a momentum of 0.9, a weight decay of 0.0005. In each iteration, both the batch sizes of the original dataset and the open-set auxiliary dataset are set as 128. We set the initial learning rate as 0.1, and reduce it by a factor of 10 after 80 and 140 epochs. We use 5k noisy samples as the validation to tune the hyperparameter η in $\{0.1, 0.5, 1, 2.5, 5\}$, then train the model on the full training set and report the average test accuracy in the last 5 epochs. All experiments are repeated five times with different seeds and more training details can be found in Appendix C. We use 80 Million Tiny Images [31] as the open-set auxiliary dataset, More discussions on the different choices of open-set auxiliary dataset are presented in Appendix D.

Table 2: Average test accuracy (%) with standard deviation on CIFAR-100 under various types of noisy labels (over 5 trials). The bold indicates the improved results by integrating our regularization.

Method	Sym-20%	Sym-50%	Asymmetric	Dependent
Standard	64.87±0.81	48.55±0.65	48.77±0.86	53.94±0.52
+ ODNL (Ours)	**68.98±0.16**	**54.35±0.63**	**55.07±0.38**	**60.47±0.27**
Decoupling	68.98±0.23	49.09±0.81	49.92±0.71	55.17±0.34
+ ODNL (Ours)	**67.06±0.25**	**53.65±0.56**	**57.74±0.54**	**59.32±0.39**
F-correction	62.05±0.35	52.27±0.15	54.45±0.94	57.47±0.26
+ ODNL (Ours)	**65.60±0.21**	**57.34±0.37**	**57.60±0.52**	**61.83±0.93**
PHuber-CE	65.17±0.87	48.46±0.24	47.64±0.45	53.56±0.29
+ ODNL (Ours)	**72.18±0.41**	**63.02±0.46**	**49.27±0.38**	**67.16±0.25**
Co-teaching	72.35±0.60	65.21±0.88	54.16±0.17	67.85±0.82
+ ODNL (Ours)	**73.34±0.13**	**65.43±0.61**	**55.02±0.29**	**68.41±0.14**
JoCoR ($\lambda = 0.5$)	73.44±0.19	67.53±0.79	57.02±0.32	70.01±0.26
+ ODNL (Ours)	**74.74±0.21**	**68.17±0.34**	**58.57±0.46**	**71.68±0.40**

Table 3: Test accuracy (%) on Clothing1M with pretrained ResNet-50. All experiments are implemented based on DivideMix's official implementation. The bold indicates the improved results.

Method	Standard	Forward	Co-teaching	**ODNL (Ours)**	SLN	**+ ODNL (Ours)**	DivideMix	**+ ODNL (Ours)**
best	70.17	70.46	71.32	**72.47**	72.31	**73.12**	74.32	**74.89**
last	68.23	69.34	71.02	**72.38**	71.94	**72.98**	73.83	**74.35**

5.2 CIFAR-10 and CIFAR-100

On CIFAR-10 and CIFAR-100, we verify that ODNL can boost existing robust training methods by integrating ODNL with the following methods: 1) Decoupling [21], which updates the parameters only using instances that have different predictions from two classifiers. 2) F-correction [27], which corrects the prediction by the label transition matrix. As the setting in F-correction, we first train a standard network to estimate the transition matrix Q. 3) PHuber-CE [22], which uses a composite loss-based gradient clipping, a variation of standard gradient clipping for label noise robustness. 4) Co-teaching [7], which trains two networks simultaneously and cross-updates parameters of peer networks. 5) JoCoR [35], which trains two networks simultaneously and reduces the divergence between the two classifiers by explicit regularization.

The average test accuracy with WRN-40-2 [44] are reported in Table 1 and Table 2. The results show that incorporating our regularization into existing robust training methods consistently improves their performance against various types of noisy labels. In particular, we can observe that our method can boost almost all types of existing methods, including robust loss function, sample selection, loss correction, etc., showing that our regularization based upon an open-set auxiliary dataset is a complementary method to existing robust algorithms. More intriguingly, while most existing methods do not bring benefits to the robustness against open-set noisy labels in the training set of CIFAR-10, our regularization could still enhance the performance of all existing methods in this case.

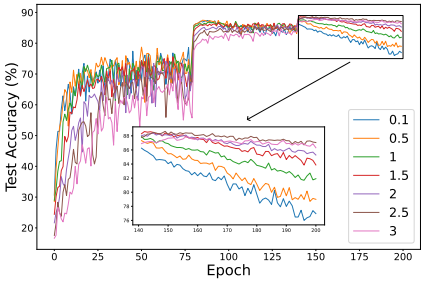


Figure 4: Results of sensitivity analysis on CIFAR-10 with different η.

To further illustrate the influence of η, we present a sensitivity analysis on CIFAR-10 with symmetric-40% noisy labels in Fig. 4. Specifically, we highlight the differences in the trend of test accuracy after the second decay of learning rate at the 140th epoch. We can observe that with a large η, the decreasing trend of test accuracy caused by inherent noisy labels is nearly eliminated. The result verifies that our regularization is an effective method to reduce the overfitting of inherent noisy labels.

Table 4: OOD detection performance comparison on CIFAR-10 with symmetric-40% noisy labels. All values are percentages and are averaged over the ten test datasets described in Appendix C. ↑ indicates larger values are better and ↓ indicates smaller values are better. Bold numbers are superior results. Detailed results for each OOD test dataset can be found in Appendix D.

Method	Test Accuracy ↑	FPR95 ↓	AUROC ↑	AUPR ↑
MSP	77.87	92.01	54.64	19.41
MSP+OE	78.12	26.76	94.02	78.34
MSP+Ours	**85.31**	**14.08**	**96.53**	**84.10**

5.3 Clothing1M

Clothing1M [39] is a large-scale dataset with real-world noisy labels of clothing images. There are 1 million noisy samples for training, 14k and 10k clean examples for validation and test respectively. Following DivideMix [18], we conduct experiments by sampling a class-balanced training subset in each epoch and use ResNet-50 with ImageNet pretrained weights. We set the η of the vanilla ODNL as 5. For the auxiliary dataset in our method, we use ImageNet 2012 dataset [29] and remove all classes related to clothes. Other training details strictly follow the settings of DivideMix [18], presented in Appendix C. As shown in Table 3, *best* denotes the score of the epoch where the validation accuracy is optimal, and *last* denotes the scores at the last epoch. The vanilla ODNL outperforms many simple baselines, and it can further improve the performance of SLN and DivideMix. The results show that our method is applicable for large-scale real-world scenarios.

5.4 OOD detection with noisy labels

Out-of-distribution (OOD) detection is an essential problem for the deployment of deep learning especially in safety-critical applications [10, 19]. From the analysis in Section 4, we know that our regularization should have a similar regularization effect to OE [10], which enforces the model to give conservative outputs on OOD examples. Here, we conduct experiments on CIFAR-10 with symmetric-40% noisy labels to verify the advantage of our method on OOD detection tasks even in a weakly supervised setting. We expect the trained models could perform well in the OOD detection task, even if the labels of training dataset are noisy. Experiments on CIFAR-10 with clean labels are also conducted and the results can be found in Appendix D. The training settings are the same as those in subsection 5.2. Following OE [10], we consider the maximum softmax probability (MSP) baseline [9], which gives the maximum of softmax output as the OOD score. For the OOD test datasets, we use three types of noises and seven common benchmark datasets: Gaussian, Rademacher [10], Blobs [10], Textures [6], SVHN[24], CIFAR100, Places365[47], LSUN-Crop [43], LSUN-Resize [43], and iSUN [41]. We measure the following metrics: (1) the false positive rate (FPR95) of OOD examples when the true positive rate of in-distribution examples is at 95%; (2) the area under the receiver operating characteristic curve (AUROC); and (3) the area under the precision-recall curve (AUPR). Table 4 presents the test accuracy and the average performance over the three types of noises and seven OOD test datasets. We can observe that our method achieves impressive improvement on both the test accuracy and the detection performance, while OE does not bring benefit on the generalization performance. The results can further serve as evidence for the analysis in Section 4.

6 Conclusion

In this paper, we proposed Open-set regularization with Dynamic Noisy Labels (ODNL), a simple yet effective technique that enhances many existing robust training methods to mitigate the issues of inherent noisy labels across various settings. To the best of our knowledge, our method is the first to utilize open-set auxiliary dataset in the problem of noisy labels. We show that ODNL can be considered as a variant of SGD noises that usually leads to a flat minimum. Extensive experiments show that ODNL can help improve the robustness against various types of noisy labels and it is applicable for large-scale real-world scenarios. Moreover, ODNL also brings a "side effect": it can improve the OOD detection performance, which is also essential for the deployment of deep learning in safety-critical applications. On the other hand, ODNL is computationally inexpensive and can be easily integrated into existing algorithms. Overall, our method is an effective and complementary approach for boosting robustness against inherent noisy labels.

References

[1] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 233–242, 2017.

[2] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, 2003.

[3] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[4] Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng. Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise. *arXiv preprint arXiv:2012.05458*, 2020.

[5] Pengfei Chen, Guangyong Chen, Junjie Ye, Jingwei Zhao, and Pheng-Ann Heng. Noise against noise: stochastic label noise helps combat inherent label noise. In *International Conference on Learning Representations*, 2021.

[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[7] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, pages 8527–8537, 2018.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[9] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[10] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *Proceedings of the International Conference on Learning Representations*, 2019.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

[13] 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. *arXiv preprint arXiv:1712.05055*, 2017.

[14] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[15] Bobby Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does sgd escape local minima? In *Proceedings of International Conference on Machine Learning*, pages 2698–2707. PMLR, 2018.

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[17] Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. In *Proceedings of International Conference on Machine Learning*, pages 3763–3772. PMLR, 2019.

[18] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

[19] Weitang Liu, Xiaoyun Wang, John D Owens, and Yixuan Li. Energy-based out-of-distribution detection. *arXiv preprint arXiv:2010.03759*, 2020.

[20] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *Proceedings of International Conference on Machine Learning*, pages 6448–6458. PMLR, 2020.

[21] Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from" how to update". *arXiv preprint arXiv:1706.02613*, 2017.

[22] Aditya Krishna Menon, Ankit Singh Rawat, Sanjiv Kumar, and Sashank Reddi. Can gradient clipping mitigate label noise? In *International Conference on Learning Representations (ICLR)*, 2020.

[23] Preetum Nakkiran, Behnam Neyshabur, and Hanie Sedghi. The deep bootstrap: Good online learners are good offline generalizers. *arXiv preprint arXiv:2010.08127*, 2020.

[24] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[25] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017.

[26] Aristotelis-Angelos Papadopoulos, Mohammad Reza Rajati, Nazim Shaikh, and Jiamian Wang. Outlier exposure with confidence control for out-of-distribution detection. *Neurocomputing*, 441:138–150, 2021.

[27] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1944–1952, 2017.

[28] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[30] Ragav Sachdeva, Filipe R Cordeiro, Vasileios Belagiannis, Ian Reid, and Gustavo Carneiro. Evidentialmix: Learning with combined open-set and closed-set noisy labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3607–3615, 2021.

[31] Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[32] Brendan Van Rooyen, Aditya Menon, and Robert C Williamson. Learning with symmetric label noise: The importance of being unhinged. In *Advances in Neural Information Processing Systems*, pages 10–18, 2015.

[33] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 8688–8696, 2018.

[34] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019.

[35] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13726–13735, 2020.

[36] Jingfeng Wu, Wenqing Hu, Haoyi Xiong, Jun Huan, Vladimir Braverman, and Zhanxing Zhu. On the noisy gradient descent that generalizes as sgd. In *International Conference on Machine Learning*, pages 10367–10376. PMLR, 2020.

[37] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Jiankang Deng, Jiatong Li, and Yinian Mao. Extended t: Learning with mixed closed-set and open-set noisy labels. *arXiv preprint arXiv:2012.00932*, 2020.

[38] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Parts-dependent label noise: Towards instance-dependent label noise. *arXiv preprint arXiv:2006.07836*, 2020.

[39] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2691–2699, 2015.

[40] Lingxi Xie, Jingdong Wang, Zhen Wei, Meng Wang, and Qi Tian. Disturblabel: Regularizing cnn on the loss layer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4753–4762, 2016.

[41] Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.

[42] Yan Yan, Rómer Rosales, Glenn Fung, Ramanathan Subramanian, and Jennifer Dy. Learning from multiple annotators with varying expertise. *Machine Learning*, 95(3):291–327, 2014.

[43] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[44] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[45] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *Proceedings of International Conference on Learning Representations*, 2016.

[46] Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *arXiv preprint arXiv:1805.07836*, 2018.

[47] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Appendix E.
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Subsection 5.1 and Appendix C.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Fig. 1.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix C.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]