Learning from Training Dynamics: Identifying Mislabeled Data beyond Manually Designed Features

Qingrui Jia[1,2*], Xuhong Li[2*], Lei Yu[1,3], Jiang Bian[2], Penghao Zhao[2], Shupeng Li[2], Haoyi Xiong[2†], Dejing Dou[4]

[1] Sino-French Engineer School, Beihang University, Beijing, China
[2] Baidu Inc. Beijing, China
[3] Beihang Hangzhou Innovation Institute Yuhang, Hangzhou, China
[4] BCG Greater China, Beijing, China
{jiaqingrui, lixuhong, xionghaoyi}@baidu.com, {jiaqr, yulei}@buaa.edu.cn

Abstract

While mislabeled or ambiguously-labeled samples in the training set could negatively affect the performance of deep models, diagnosing the dataset and identifying mislabeled samples helps to improve the generalization power. Training dynamics, i.e., the traces left by iterations of optimization algorithms, have recently been proved to be effective to localize mislabeled samples with hand-crafted features. In this paper, beyond manually designed features, we introduce a novel learning-based solution, leveraging a noise detector, instanced by an LSTM network, which learns to predict whether a sample was mislabeled using the raw training dynamics as input. Specifically, the proposed method trains the noise detector in a supervised manner using the dataset with synthesized label noises and can adapt to various datasets (either naturally or synthesized label-noised) without retraining. We conduct extensive experiments to evaluate the proposed method. We train the noise detector based on the synthesized label-noised CIFAR dataset and test such noise detector on Tiny ImageNet, CUB-200, Caltech-256, WebVision and Clothing1M. Results show that the proposed method precisely detects mislabeled samples on various datasets without further adaptation, and outperforms state-of-the-art methods. Besides, more experiments demonstrate that the mislabel identification can guide a label correction, namely data debugging, providing orthogonal improvements of algorithm-centric state-of-the-art techniques from the data aspect.

Introduction

Deep learning models have achieved remarkable performances across various tasks (Salakhutdinov 2014) due to the rapid development of paralleled devices, advanced network architectures and learning algorithms, as well as the large dataset collection and high-quality annotations (Deng et al. 2009a; Li et al. 2017a; Zhou et al. 2018). However, the difficulty and expensiveness of acquiring high-quality labels inevitably leave an amount of label noises on the datasets. In many real-world settings, training samples and their labels are collected through proxy variables or web scraping (Chen and Gupta 2015; Xiao et al. 2015; Joulin et al. 2016; Li et al. 2017a), which certainly leads to a larger amount of noises. According to recent studies (Zhang et al. 2021; Arpit et al. 2017; Arora et al. 2019), datasets with poor-quality labels, including mislabeled and ambiguously labeled training samples, would negatively affect the generalization performance of deep models.

To address the label noises, two lines of researches have been proposed. One is from the *algorithm* aspect (Song et al. 2020), including robust objective functions (Pereyra et al. 2017; Ghosh, Kumar, and Sastry 2017; Wang et al. 2019), sample reweight (Patrini et al. 2017; Jiang et al. 2018) and hybrid approaches (Li, Socher, and Hoi 2020; Nguyen et al. 2020; Nishi et al. 2021; Xia et al. 2021). Another line is from the *data* aspect, focusing on identifying mislabeled samples and improving the quality of datasets. A series of methods following this line are based on the observation that in the context of Stochastic Gradient Descent (SGD), hand-crafted features on **training dynamics** are able to separate the mislabeled samples from clean ones (Toneva et al. 2019; Pleiss et al. 2020; Swayamdipta et al. 2020; Paul, Ganguli, and Dziugaite 2021). Subtle differences on the training dynamics can be discerned, as in Figure 1, where the averaged probability of being classified into the labeled category (given-label probability) across mislabeled samples is lower and their variance is higher compared to the clean ones. More distinguishable features of training dynamics can even characterize the learning difficulty of each training sample (Hacohen, Choshen, and Weinshall 2020; Baldock, Maennel, and Neyshabur 2021).

However, the aforementioned distinguishable features are usually extracted manually from the training dynamics with prior knowledge and domain expertise, and it is not always fitting well and being efficient. Tracing back the history of machine learning, the deep learning methods are much more effective than traditional learning with handcraft features, by learning features from raw data and solving the learning tasks in end-to-end fashions. In light of the evolution of machine learning, we propose a novel solution through *learning from training dynamics* to identify mislabeled samples. Beyond manually designed features, our

[*] The first two authors contributed equally to this work.
[†] Corresponding author.

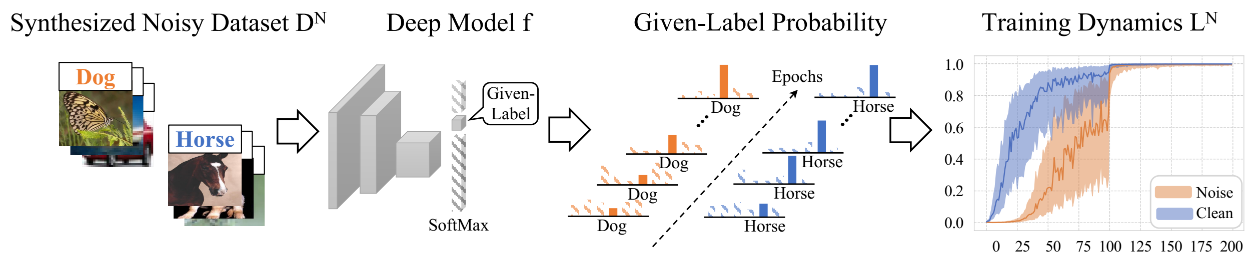

Figure 1: Acquisition of training dynamics. The plots on the right are averaged across 1000 random samples from the synthesized-noise Caltech-256 dataset, with the training epoch index as X-axis and the given-label probability as Y-axis.

proposed method[1] trains a *noise detector*, instanced by an LSTM network (Hochreiter and Schmidhuber 1997), on the raw training dynamics, with the objective of learning distinguishable features to solve the binary classification task, i.e., separating mislabeled samples from clean ones.

The most efficient and effective way to train the noise detector is the supervised learning, where the binary annotations whether samples being mislabeled or not are required. However, this kind of information is usually not provided by public datasets. To cope with this issue, we purposely wrong-set the labels of certain samples from an (almost) clean dataset, and conduct the standard training process on it. In this way, the raw training dynamics can be trained with the supervision of the synthesized label noises.

We carry out three types of evaluation experiments to validate the effectiveness of our method in identifying mislabeled samples. The results show that (1) The noise detector trained on the synthesized noisy CIFAR-100 can precisely identify the mislabels on other synthesized and real-world noisy datasets without further adaptation or retraining, showing a good transfer ability; (2) Compared to the state-of-the-art approaches from the data aspect, our method outperforms with a clear margin; and (3) Combining with methods from the algorithm aspect, our method provides an orthogonal improvement over the current methods.

In summary, our main contributions are the following:

- To identify the mislabeled samples, we propose a novel supervised-learning-based solution to train an LSTM-instanced label *noise detector* based on the synthesized noisy dataset and its training dynamics.
- The noise detector trained on a noised CIFAR-100 is an off-the-shelf detector for identifying the mislabeled samples, i.e., it can accurately find label noises without any further adaptation or retraining on other noised datasets, including Tiny ImageNet, CUB-200, Caltech-256, WebVision and Clothing 100K, where the latter two are real-world noisy datasets.
- On three types of evaluation experiments, our proposed method outperforms all the existing methods of identifying the mislabeled samples. Analyses and discussions are also provided to validate the effectiveness of our method.
- Combing with the learning-with-noisy-labels algorithms, our proposed method further boosts test accuracies of the models trained on noisy datasets. This also proves that the improvements from the data aspect can be orthogonal to those from the algorithm aspect.

[1]Code available at https://github.com/Christophe-Jia/mislabel-detection and at InterpretDL (Li et al. 2022) as well.

Related Work

Learning with label noise has achieved remarkable success. We review a good amount of prior works most relevant to ours and categorize them into two aspects, *algorithms* or *data*. For a complete review, we refer to Song et al. (2020); Algan and Ulusoy (2021); Mazumder et al. (2022).

From Algorithms Aspect. Tremendous approaches provide algorithms-centric perspectives and stay in the conventional mainstream for robust learning in presence of mislabeled samples. Some approaches proposed to estimate the noise transition matrix via pretrained model (Patrini et al. 2017), clean validation set (Hendrycks et al. 2018) or expectation-maximization (EM) algorithm (Goldberger and Ben-Reuven 2017). A number of methods propose to improve the noise immunity by designing robust loss functions (Ghosh, Kumar, and Sastry 2017; Wang et al. 2019), or modify loss or probabilities (Arazo et al. 2019; Yao et al. 2019; Reed et al. 2014) to compensate for negative impact owing to noisy labels. Some alternative methodologies aim to refurbish the corrupted labels, by characterizing the distribution of noisy labels (Xiao et al. 2015; Vahdat 2017; Li et al. 2017b; Lee et al. 2018) or in a meta-learning manner (Li et al. 2017c; Zheng, Awadallah, and Dumais 2021).

Remarkably, the state-of-the-art methods are usually hybrid, integrated of above components. For examples, DivideMix (Li, Socher, and Hoi 2020) leverages Gaussian mixture model to distinguish mislabel ones, then regards them as unlabeled samples, and trains the model in a semi-supervised learning manner. Nishi et al. (2021) improves the DivideMix framework by applying different data augmentation strategies for finding noisy labels and learning models. Through curriculum learning (Bengio et al. 2009), RoCL (Zhou, Wang, and Bilmes 2021) first learns from clean data and then gradually involves noisy-labeled data corrected by the predictions from an ensemble based model.

In brief, algorithm-centric training with label noises requires a modification of the optimization algorithm. The proposed approach in this paper, from a data-centric aspect, does not modify the optimization procedure but filters the label noises in the first step. By all means, algorithm-centric methods are used to improve the optimization procedures, orthogonal to data-centric methods introduced as follows.

From Data Aspect. Fundamentally different from the above methods in learning with label noise, a lot of approaches focus on data-centric amelioration and construct higher quality datasets, decoupled from training procedures. A straightforward thought of harnessing mislabeled samples is excluding them from the training data and then retraining on the remaining samples (Brodley and Friedl 1999). Therefore, various studies endeavor to detect mislabeled samples or sort clean samples. One of the prevalent methods is unsupervised outliers removal (Liu, Hua, and Smith 2014; Xia et al. 2015), which is based on the assumption that outliers are incorrectly labeled.

Recently, instead of simply checking the final predictions, more works analyze the *training dynamics*, i.e., the traces left by the steps during SGD. For instance, Toneva et al. (2019) defines a forgetting event using training dynamics, and samples prone to be forgotten with high frequency probably correspond to be mislabeled. Swayamdipta et al. (2020) builds a data map with training dynamics, then marks off a region composed of instances that the model finds hard to learn in dataset mapping, which might be labeling errors. Pleiss et al. (2020) proffers a statistic called area under margin (AUM) and introduces threshold samples to separate clean and mislabeled samples. In addition, other solutions filter noisy data from perspective of influence functions (Koh and Liang 2017) for identifying training points that exert the most influence on the model. Ghorbani and Zou (2019) introduced data Shapley to quantify the value of each training data to the predictor performance and considered the low Shapley value data as corruptions. All of them utilize statistics or manually designed features to diagnose the dataset. In this paper, based on training dynamics, we propose a novel learning-based solution, that trains a noise detector on the synthesized noisy dataset and detects the mislabeled samples in the original dataset.

The Proposed Approach

Training dynamics are distinguishable features to identify the mislabeled samples. As shown in Figure 1, the mean of the given-label probability over noisy samples is lower than the one over clean samples; but the variance is higher, especially at the latter half of the training, indicating that the mislabeled samples are more complicated to fit than the clean samples. However, with these visually visible differences, it is still challenging to identify the mislabeled samples precisely. Designing more expert features will surely be helpful, but instead, we propose to benefit from the advantage of deep learning models to learn good features for classifying the mislabeled samples from clean ones.

Definition of Training Dynamics

We give notations and definitions within the context of classification learning problems without loss of generality.

In the context of SGD or its variant, we define the **training dynamics** as the probability values of all training samples obtained by the model at each training epoch, denoted as $\boldsymbol{L}$. Specifically, $\boldsymbol{L}(i,t)$ is the probabilities of the i-th sample at the t-th training epoch. At each epoch, the model would pass through forward and backward propagation for

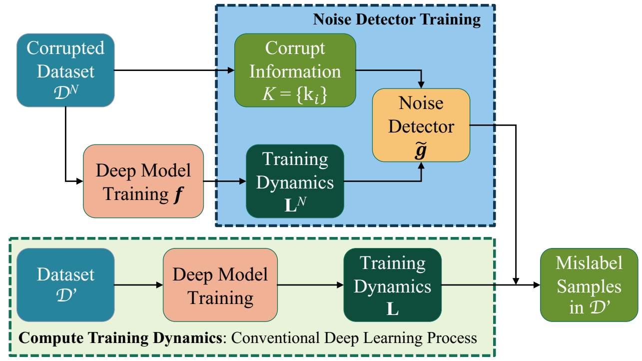

Figure 2: Supervised Learning from Training Dynamics.

each training sample, with the probability values as intermediate results that can be recorded during the training process without additional computations[2]. Note that $\boldsymbol{L}(i,t)$ is a vector of C elements, where C is the number of classes, and we note $\boldsymbol{L}_c(i,t)$ as the probability of the c-th class. Noting the deep model as $\boldsymbol{f}$, then we have

$$\boldsymbol{L}(i,t) = \boldsymbol{f}^{(t)}(x_i), \tag{1}$$

where x_i is the input data. For computing gradients and updating the model's parameters, the label y_i is needed.

Supervised Learning from Training Dynamics

We present the supervised learning from training dynamics in three steps as following.

Step 1. Given an (almost) clean dataset $\mathcal{D}$, we first generate the label noises and obtain the noised dataset $\mathcal{D}^N$. Specifically, $\mathcal{D}$ contains the data instances (x_i, y_i) while $\mathcal{D}^N$ contains (x_i, y_i^N, k_i). The additional term k_i indicates that the sample is intentionally mislabeled ($k_i = 1$) or not ($k_i = 0$). The design of the noise generator follows the previous works (Pleiss et al. 2020; Northcutt, Jiang, and Chuang 2021), where the symmetric and asymmetric noises are used in our experiments. Following the noise generator, the new values of y_i^N will overwrite the original information in $\mathcal{D}$, and k_i can be simply obtained by

$$k_i = \begin{cases} 1, & \text{if } y_i^N \neq y_i; \\ 0, & \text{if } y_i^N = y_i. \end{cases} \tag{2}$$

The noised datasets $\mathcal{D}^N$ are only used to train the LSTM-instanced noise detector, denoted as $\boldsymbol{g}$. The trained noise detector $\tilde{\boldsymbol{g}}$ (as introduced in Eq (3)) is directly applicable to other real-world datasets.

As for the (almost) clean dataset $\mathcal{D}$, though some famous datasets, such as CIFAR-10 and ImageNet, contain a small ratio ($< 3\%$) of mislabeled samples (Northcutt, Athalye, and Mueller 2021), our method is well-adapted to a mild quantify of underlying noises in the dataset. This is shown in Subsection .

Step 2. We train the deep model $\boldsymbol{f}$ on $\mathcal{D}^N$ using SGD or one of its variants, and record the training dynamics $\boldsymbol{L}^N$. We

[2] For cases when the value of a certain sample at some epoch is missing, interpolations across training epochs can be used.

Algorithm 1: Supervised Learning from Training Dynamics.

Input: $\mathcal{D}$ an (almost) clean dataset for noise detector training, $\mathcal{D}'$ datasets that requires mislabel identification, $\boldsymbol{f}$ deep model architecture.
Step 1:
  Generate label noises based on $\mathcal{D}$, and obtain $\mathcal{D}^N$ for training the noise detector $\boldsymbol{g}$ in Step 2.
Step 2:
  Train $\boldsymbol{f}$ on $\mathcal{D}^N$, and get $\boldsymbol{L}^N$.
  Train the noise detector $\boldsymbol{g}$ using $\boldsymbol{L}^N$ and $K = \{k_i\}$ according to Eq (3), and obtain $\tilde{\boldsymbol{g}}$.
Step 3:
  Train $\boldsymbol{f}$ on $\mathcal{D}'$ and get $\boldsymbol{L}$.
  Compute $\tilde{\boldsymbol{g}}(\boldsymbol{L}_{y_i}(i))$ for each sample (x_i, y_i) in $\mathcal{D}'$, and predict whether (x_i, y_i) is mislabeled.
Output: Predicted mislabeled samples in $\mathcal{D}'$.

supervise the noise detector $\boldsymbol{g}$ taking the training dynamics $\boldsymbol{L}^N$ as feature inputs and the label noise information $K = \{k_i\}$ as supervision labels. Specifically, the noise detector is optimized following

$$\tilde{\boldsymbol{g}} = \arg\min_{\boldsymbol{g}} \sum_i \mathcal{L}(\boldsymbol{g}(\boldsymbol{L}^N(i)), k_i) \ , \tag{3}$$

where $\mathcal{L}(\cdot, \cdot)$ is the binary cross entropy loss function, and the optimization process is guided by an AdamW optimizer (Kingma and Ba 2015; Loshchilov and Hutter 2019). Moreover, to deal with the epoch-wise training dynamics, $\boldsymbol{g}$ is instanced by an LSTM model which is a popular choice for time-series sequences. In practice, for efficiency and generality, we only take the given-label probability as input, i.e., $\boldsymbol{L}^N_{y_i}(i)$. Involving other classes would increase the parameters and computations in the LSTM model and the efficiency would be encumbered for datasets with over hundreds of classes.

Step 3. Finally, we conduct the standard training process on the clean dataset $\mathcal{D}'$, and record the training dynamics $\boldsymbol{L}$. The previously trained $\tilde{\boldsymbol{g}}$ is then used to estimate the probability of the label y_i being mislabeled according to the training dynamics $\boldsymbol{L}_{y_i}(i)$, i.e., computing $\tilde{\boldsymbol{g}}(\boldsymbol{L}_{y_i}(i))$.

The procedure of the proposed approach is summarized in Algorithm 1. We introduce three extensions of this proposed approach in the following subsection to show the applicability and robustness of our approach for practical usages.

Characteristic Analysis

The proposed approach effectively identifies the mislabeled samples, demonstrated by the experiments in Section . Here we provide the characteristic analysis to show the transferability, robustness and applicability of our method.

Off-The-Shelf Noise Detector and Transferability. A trained $\tilde{\boldsymbol{g}}$ on one dataset is capable of identifying the mislabeled samples on other datasets. Once the noise detector $\tilde{\boldsymbol{g}}$ is trained, **Step 3** can process standalone for any dataset $\mathcal{D}'$, which proves that $\tilde{\boldsymbol{g}}$ is generalizable and has learned some common features for identifying mislabeled samples. In Section , we show that $\tilde{\boldsymbol{g}}$ trained from the noised CIFAR-100 works quite well on other (supposedly-) clean datasets and noisy datasets. Note that further fine-tuning of $\tilde{\boldsymbol{g}}$ on $\mathcal{D}'$, following the exact procedure described in Section , can always provide further improvements.

Robustness against Underlying Noises. In real-world scenarios, the dataset $\mathcal{D}$ may contain a small or mild ratio of noises. This would introduce noises in the training process of $\boldsymbol{g}$. We simulate this scenario by manually changing the labels of the (supposedly-)clean dataset and setting it to $\mathcal{D}$. Then we process the three steps described previously and use $\tilde{\boldsymbol{g}}$ to detect the mislabeled samples. According to our analytical experiments, the noise detector $\tilde{\boldsymbol{g}}$ is still effective in this setting, up to a large ratio of noises. This relaxes the constraint on the label quality of $\mathcal{D}$.

Usability of Super-Classes Noises. During the dataset annotation, human experts usually mislabel the samples into similar categories but rarely to those with very different categories. This pattern of mislabeling is easy to mimic, especially in CIFAR-100, which has 20 super-classes, followed by 5 finer categories. We follow this setting in Section to generate the noisy labels within the super-class, instead of across all 100 classes. Experiments show that our method is still capable of accurately identifying the mislabeled samples under this super-classes noise setting.

Experiments

On synthesized and real-world noisy datasets, three types of evaluations are conducted and presented in Section , and , respectively: (1) On synthesized datasets, where the mislabeled samples are known, the precision and recall can be directly calculated. (2) Following the procedure of previous works (Brodley and Friedl 1999; Pleiss et al. 2020), the evaluations are performed by checking the performance of the supervised learning with/without (by excluding) mislabeled samples in the datasets. A higher test accuracy trained on the dataset without mislabeled samples indicates a better identification of mislabeled samples. (3) Similar to the data debugging evaluation scheme (Mazumder et al. 2022), the evaluation procedure first identifies the label errors and then corrects them with the true labels or pseudo labels generated by the model in (2), i.e., trained excluding label errors. Afterward, the corrected dataset is incorporated with the state-of-the-art learning algorithms, e.g., DivideMix (Li, Socher, and Hoi 2020) and DM-AugDesc (Nishi et al. 2021), to enjoy the orthogonal improvements from the data aspect.

Identify Mislabeled Samples

This section validates our approach on the datasets with synthesized label noises. We follow the commonly used noise distributions (Northcutt, Jiang, and Chuang 2021; Pleiss et al. 2020) to generate label noises at different ratios and change the labels of the training set with symmetry (or asymmetry) noises but leave the testing set unchanged.

Here we mainly compare our method with **AUM** (Pleiss et al. 2020), one of the most effective approaches that also use the training dynamics to identify mislabeled samples.

Experiment Setups. Evaluations are conducted on

Dataset	CIFAR-10				CIFAR-100			
Noise	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
Standard	75.0±0.3	56.7±0.4	36.7±0.4	16.6±0.2	49.6±0.2	37.5±0.2	23.8±0.4	8.2±0.3
DY-Bootstrap	79.4±0.1	68.8±1.4	56.4±1.7	Diverged	53.0±0.4	43.0±0.3	36.6±0.5	12.8±0.5
Data Param	82.1±0.2	70.8±0.9	49.3±0.7	18.9±0.3	56.3±1.4	46.1±0.4	32.8±2.3	11.9±1.2
INCV	89.5±0.1	86.8±0.1	81.1±0.3	53.3±1.9	58.6±0.5	55.4±0.2	43.7±0.3	23.7±0.6
AUM	90.2±0.0	87.5±0.1	82.1±0.0	54.4±1.6	**65.5±0.2**	**61.3±0.1**	53.0±0.5	**31.7±0.7**
Ours	**91.1±0.0**	**88.9±0.0**	**83.5±0.2**	**55.5±1.4**	**65.6±0.3**	60.8±0.2	**54.3±0.4**	30.1±0.6
Oracle	91.0±0.1	90.3±0.4	89.2±0.9	87.4±1.7	64.5±0.1	61.0±0.2	55.2±0.5	44.9±0.4

Table 1: Testing accuracy on the CIFAR-10/100 with symmetric label noises (ResNet-32).

four datasets with synthesized noisy labels: CIFAR-10/100 (Krizhevsky and Hinton 2009), CUB-200-2011 (Wah et al. 2011) and Caltech256 (Griffin, Holub, and Perona 2007). Evaluation metrics include **mAP**, **ROC AUC** scores and **Precision@95**, where **Precision@95** represents the precision when the recall reach 95%, which can be directly measured because of the available ground-truth. Training dynamics are insensitive to the deep model architectures (Toneva et al. 2019; Pleiss et al. 2020). We therefore choose ResNets as the deep model $\boldsymbol{f}$ for the higher testing accuracy in original datasets, while we have also tested VGGs (Simonyan and Zisserman 2015), which show similar results in preliminary experiments. Concerning the noise detector, for efficiency, an LSTM of two hidden layers with 64 dimensions for each is used. All the noise detectors share the same architecture, allowing the possible transfer between datasets.

Results. Following the steps in Section , we supervise the noise detector instanced by an LSTM model on the noised CIFAR-10/100 datasets with the corrupted labels. Our method is competitive with AUM without threshold strategy. Figure 3 depicts the mAP, AUC and Precision@95 scores for identifying mislabeled samples from the noised CIFAR-10/100 datasets. Our procedure excels AUM in both symmetric and asymmetric noise settings, the later is confirmed in Figure 4.

Additional experiments have been conducted on the noised CUB-200-2011 and Caltech-256. As shown in Figure 5, our approach benefits from the well-trained noise detector $\tilde{g}$ on CIFAR-100 with 30% noise ratio. Then we fine-tune $\tilde{g}$ on the noised CUB-200-2011 and Caltech-256, respectively, to further improve the performance of the noise detection with a clear margin.

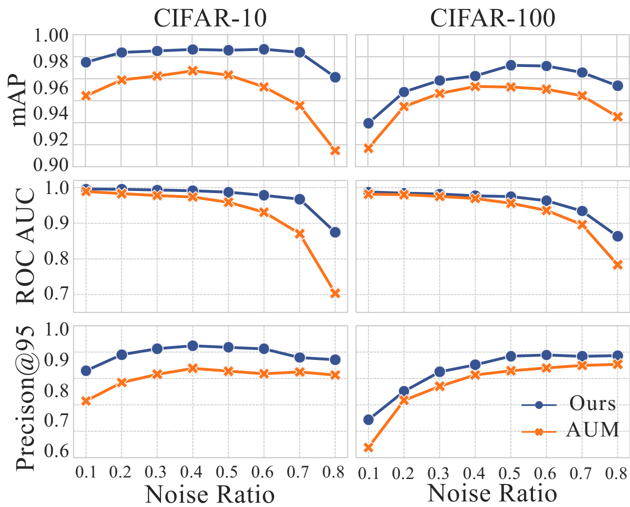

Figure 3: Scores of mAP, ROC AUC and Precision@95 for identifying mislabeled samples on CIFAR-10/100 with symmetric label noises.

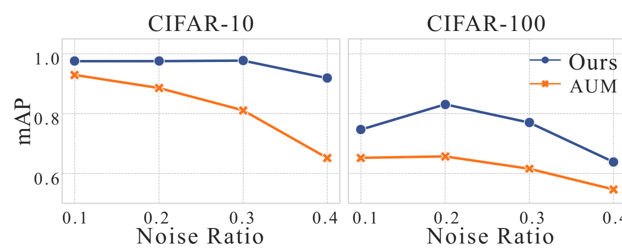

Figure 4: Scores of mAP for identifying mislabeled samples on CIFAR-10/100 with asymmetric label noises.

Retrain after Excluding Mislabeled Samples

Mislabeled samples are harmful to the model (Arora et al. 2019; Zhang et al. 2021), and removing them should be beneficial for its accuracy and generalizability. To validate the effectiveness of identification, we remove samples predicted to be mislabeled using the proposed approach and then retrain the deep model. Based on the testing accuracy after excluding samples, we are able to measure the effectiveness of mislabeled sample identification. Within this setting, we compare our method against several existing ones, introduced in the following paragraph.

Baselines. We compare our approach with several methods from the existing literature. **DY-Bootstrap** (Arazo et al. 2019) proposes to fit a beta mixture model to estimate the probability that a sample is mislabeled. **Data Param** (Saxena, Tuzel, and DeCoste 2019) allows a learnable parameter to each sample or class, which controls their importance during the training stage. **INCV** (Chen et al. 2019) presents a strategy to select a clean subset successively harnessing iterative cross-validation. **AUM** (Pleiss et al. 2020) has been described above. **Standard** represents training with the entire dataset, without removing any samples. **Oracle** means training without any synthesized noisy samples.

Experiment Setups. We test our approach on 7 datasets,

	CIFAR-10	CIFAR-100	Tiny ImageNet	WebVision50	Clothing 100K
Standard	91.9±0.1	67.0±0.3	50.7±0.1	78.6	64.2
Data Param	91.9±0.0	63.6±1.4	51.6±0.2	78.5	64.5
DY-Bootstrap	90.0±0.0	65.1±0.1	48.4±0.0	74.2	61.6
INCV	90.9±0.0	61.8±0.1	43.9±0.1	77.9	66.7
AUM	92.1±0.0	68.2±0.1	51.4±0.1	80.2	66.5
Ours	**92.5±0.1**	**68.7±0.3**	**52.2±0.0**	**81.4**	**68.2**

Table 2: Testing accuracy on real-world datasets (ResNet-32 for CIFAR/Tiny I.N., ResNet-50 for others).

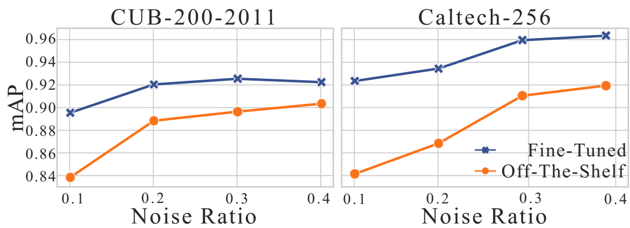

Figure 5: mAP on fine-tuned and off-the-shelf detectors on two synthesized noisy CUB-200-2011 and Caltech-256.

Algorithm	CUB-200-2011 20% Sym. Best	CUB-200-2011 20% Sym. Last	mini WebVision Top1	mini WebVision Top5
DM-AugDesc	79.65	79.01	78.64	93.20
+ 5% D.D.[Ours]	79.55	**79.15**	**79.32**	92.56
+ 10% D.D.[Ours]	**80.14**	**79.65**	**78.84**	92.84
DivideMix	75.96	73.89	77.32	91.64
+ 5% D.D.[Ours]	**76.06**	**74.44**	**77.84**	**92.16**
+ 10% D.D.[Ours]	**76.77**	**74.99**	**78.24**	**92.72**

Table 3: Testing accuracy on noisy CUB-200-2011 and original mini WebVision. D.D. denotes data debugging for short. Competitive results (better than baselines) are bolded.

i.e., two synthesized noisy CIFAR datasets yet with four different noise ratios, two original CIFAR10/100 datasets, Tiny ImageNet (Deng et al. 2009b), WebVision (Li et al. 2017a) and Clothing1M (Xiao et al. 2015), where the latter two are real-world noisy datasets. Note that for a fair comparison with AUM and other baseline methods, we use WebVision50 of both Google and Flickr images and a subset of Clothing1M, named Clothing100K. Both of them contain around 100,000 images. For the same reason of fair comparisons with baselines, the same deep model is used for all experiments, where the model architecture is indicated for respective experiments.

Results on Synthesized Noisy CIFAR-10/100. Table 1 displays the test accuracies on noised CIFAR-10/100, each noised by four different ratios. Our identification procedure outperforms other methods in almost every setting and surpasses oracle performance on both 20% noisy CIFAR-10/100, indicating that some label noises in the original version of CIFAR-10/100 have been found.

Results on Public Datasets. In this evaluation, the noise detector used in our approach is trained in advance on a noised CIFAR-100 and applied in an off-the-shelf manner.

Original CIFAR-10/100 and (Tiny)ImageNet datasets also contain mild label noises (Northcutt, Athalye, and Mueller 2021). We apply our method to diagnose the CIFAR-10/100 and Tiny ImageNet datasets and find a small part of images that are considered as mislabeled samples by our method. Then we remove them from the training set and retrain the deep models. Table 2 (three columns on the left) shows that our identification method works well on low-noise datasets and surpasses the performance of existing approaches.

WebVision50 and Clothing100K are known for their label noises and are often used to evaluate the algorithms of identifying mislabeled samples. In Table 2 (two columns on the right), removing suspected mislabeled samples in the training set leads to a 2.8% and 4% enlargement of test accuracies on WebVision50 and Clothing100K, respectively. By use of the novel learning solution based on training dynamics, our method shows versatility in identifying mislabeled samples across model architectures, data distributions, and generations of label noises.

Combine with Algorithm-Centric Approaches

From an orthogonal aspect, we evaluate the enhancement to algorithm-centric state-of-the-art solution attributed to decontamination of training data. Following the debugging algorithm described in Mazumder et al. (2022), we first select a number of samples with the most suspicion as label noise. Labels of these samples are then replaced by error-free ones, namely data debugging. We conduct experiments on both synthesized and real-world noisy datasets, i.e., a noised CUB-200 with 20% symmetric noises and mini WebVision[3]. Note that the noise detector used here is trained in advance on a noised CIFAR-100. We follow the same experiment setups as DivideMix and DM-AugDesc, where ResNet-50 (He et al. 2016) and Inception-ResNet v2 (Szegedy et al. 2017) are used respectively.

Data Debugging Boosts. Table 3 shows the performance after debugging some training instances detected by our approach. By simply applying our data-centric solution to algorithm-centric states of the art, i.e., DivideMix and DM-AugDesc, testing accuracies on both synthesized and real-world noisy datasets are further improved without any changes in model architectures or optimization processes.

Analyses

We provide three analyses on the proposed noise detector.

Tolerance to Underlying Noise. Underlying noises may exist in real-world datasets, even in CIFAR-10. The super-

[3]mini WebVision denotes the Google-resourcing mini WebVision as previous works(Li, Socher, and Hoi 2020; Nishi et al. 2021).

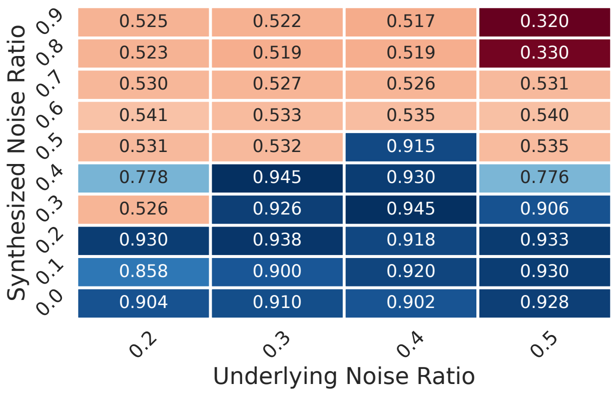

Figure 6: Robustness against underlying noise in mAP. The X-axis and Y-axis represent synthesized and underlying noise ratio of a twice-contaminated CIFAR-10.

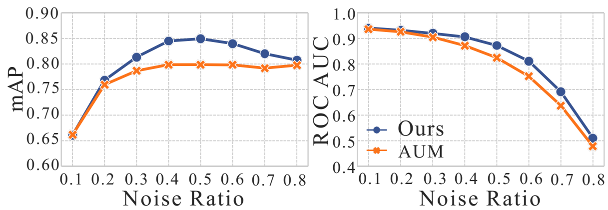

Figure 7: Scores of mAP for identifying noises on CIFAR-100 where the noises are generated within super-classes.

vision information used in our approach would be affected by the unknown underlying noises. To mimic this situation, we define a **twice-contaminated** operation. That is to say, we contaminate the dataset twice, where the first contamination is representative of underlying noises and the second one for synthesized noises. Empirically, we demonstrate that the noise detector can tolerate large underlying noises ratios.

Figure 6 displays the mAP scores against the twice-contaminated datasets with underlying noises (y-axis) and synthesized noises (x-axis). The mAP scores are measured on an independent synthesized noised Caltech256 with 40% symmetric noises, using an LSTM network as the noise detector trained on the twice-contaminated CIFAR-10 datasets. Two facts are observed: 1) Under the same synthesized noise ratios, more underlying noises would harm the detector. 2) The performance of the noise detector is satisfied if underlying noise ratio is no larger than 20%, showing good robustness against underlying noise.

Noises within Super-Classes of CIFAR-100. The 100 classes in the CIFAR-100 are grouped into 20 super-classes. To simulate the mislabeling in the annotation process, we conduct experiments generating noises within the super-class. This is more challenging but closer to practical scenarios. We also report the ROC AUC and mAP scores and compare them with AUM. Figure 7 demonstrates the advantage of our approach in identifying label noises within super-class in all settings of noise ratio.

Feature Importance Analysis through LIME. Though the LSTM network does not necessarily have a set of interpretable rules to identify the mislabeled samples, we could analyze the importance of features through local explanation methods, such as LIME (Ribeiro, Singh, and Guestrin 2016). Figure 8 shows the results of LIME on three mislabeled samples from noised CIFAR-10 that are correctly found by the noise detector. From the LIME results, the first half of training dynamics (before the learning rate drops) are more important than the second half, which is mainly in accordance with AUM. Moreover, the trained LSTM network also takes another half into consideration, where the given-label probabilities of mislabeled samples have a more significant variance than clean samples, as shown in Figure 1. While the exact formula of identifying the noise is hard to deduce, the local explanations indicate that such LSTM has learned relevant features to find mislabeled samples.

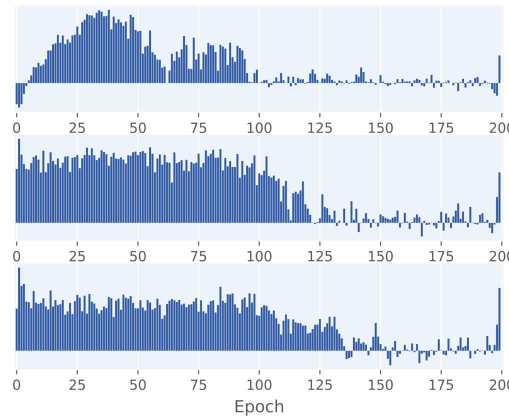

Figure 8: Instance-wise feature importance given by LIME. The X-axis and Y-axis are respectively the feature index, i.e., epoch, and the importance score.

Conclusion

In this paper, we introduced a learning-based approach to identify mislabeled samples from training dynamics, which trains label *noise detectors* instanced by an LSTM network using synthesized noises. Beyond handcraft features, detectors can automatically learn relevant features from the time-series data to identify the mislabeled samples in the training set. This trained noise detector is then applicable to other noisy datasets without further adaptation or retraining. To validate the effectiveness of the proposed approach in detecting label noises, we conducted experiments on synthesized and real-world noisy datasets, where our approach excels the existing data-centric methods in both settings. We further combined our approach with the mainstream algorithms of learning with noisy labels, and provided orthogonal improvements from the data aspect. More analyses on the robustness and applicability of our approach were also provided in details.

For both researchers and practitioners, identifying mislabeled data can prelude the model development pipeline, at the sacrifice of only one additional step to obtain the corresponding training dynamics. Samples predicted to be mislabeled can be further re-used in other manners, such as advanced data debugging and semi-supervised learning.

Acknowledgements

This work is supported in part by National Key R&D Programs of China under the grant No. 2021ZD0110303.

References

Algan, G.; and Ulusoy, I. 2021. Image classification with deep learning in the presence of noisy labels: A survey. *Knowl. Based Syst.*, 106771.

Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N. E.; and McGuinness, K. 2019. Unsupervised Label Noise Modeling and Loss Correction. In *Proc. of ICML*, 312–321.

Arora, S.; Du, S. S.; Hu, W.; Li, Z.; and Wang, R. 2019. Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks. In *Proc. of ICML*, 322–332.

Arpit, D.; Jastrzebski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M. S.; Maharaj, T.; Fischer, A.; Courville, A. C.; Bengio, Y.; and Lacoste-Julien, S. 2017. A Closer Look at Memorization in Deep Networks. In *Proc. of ICML*, 233–242.

Baldock, R. J. N.; Maennel, H.; and Neyshabur, B. 2021. Deep Learning Through the Lens of Example Difficulty. In *Proc. of NeurIPS*, 10876–10889.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proc. of ICML*, 41–48.

Brodley, C. E.; and Friedl, M. A. 1999. Identifying Mislabeled Training Data. *J. Artif. Intell. Res.*, 131–167.

Chen, P.; Liao, B.; Chen, G.; and Zhang, S. 2019. Understanding and Utilizing Deep Neural Networks Trained with Noisy Labels. In *Proc. of ICML*, 1062–1070.

Chen, X.; and Gupta, A. 2015. Webly Supervised Learning of Convolutional Networks. In *Proc. of ICCV*, 1431–1439.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Li, F. 2009a. ImageNet: A large-scale hierarchical image database. In *Proc. of CVPR*, 248–255.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Li, F. 2009b. ImageNet: A large-scale hierarchical image database. In *Proc. of CVPR*, 248–255.

Ghorbani, A.; and Zou, J. Y. 2019. Data Shapley: Equitable Valuation of Data for Machine Learning. In *Proc. of ICML*, 2242–2251.

Ghosh, A.; Kumar, H.; and Sastry, P. S. 2017. Robust Loss Functions under Label Noise for Deep Neural Networks. In *Proc. of AAAI*, 1919–1925.

Goldberger, J.; and Ben-Reuven, E. 2017. Training deep neural-networks using a noise adaptation layer. In *Proc. of ICLR*, 1–9.

Griffin, G.; Holub, A.; and Perona, P. 2007. Caltech-256 Object Category Dataset. *California Institute of Technology*.

Hacohen, G.; Choshen, L.; and Weinshall, D. 2020. Let's Agree to Agree: Neural Networks Share Classification Order on Real Datasets. In *Proc. of ICML*, 3950–3960.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proc. of CVPR*, 770–778.

Hendrycks, D.; Mazeika, M.; Wilson, D.; and Gimpel, K. 2018. Using Trusted Data to Train Deep Networks on Labels Corrupted by Severe Noise. In *Proc. of NeurIPS*, 10477–10486.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Comput.*, 1735–1780.

Jiang, L.; Zhou, Z.; Leung, T.; Li, L.; and Fei-Fei, L. 2018. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. In *Proc. of ICML*, 2309–2318.

Joulin, A.; van der Maaten, L.; Jabri, A.; and Vasilache, N. 2016. Learning Visual Features from Large Weakly Supervised Data. In *Proc. of ECCV*, 67–84.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proc. of ICLR*, 134–167.

Koh, P. W.; and Liang, P. 2017. Understanding Black-box Predictions via Influence Functions. In *Proc. of ICML*, 1885–1894.

Krizhevsky, A.; and Hinton, G. 2009. Learning Multiple Layers of Features from Tiny Images. *Master's thesis, Department of Computer Science, University of Toronto*, 32–33.

Lee, K.; He, X.; Zhang, L.; and Yang, L. 2018. CleanNet: Transfer Learning for Scalable Image Classifier Training With Label Noise. In *Proc. of CVPR*, 5447–5456.

Li, J.; Socher, R.; and Hoi, S. C. H. 2020. DivideMix: Learning with Noisy Labels as Semi-supervised Learning. In *Proc. of ICLR*, 1–14.

Li, W.; Wang, L.; Li, W.; Agustsson, E.; and Gool, L. V. 2017a. WebVision Database: Visual Learning and Understanding from Web Data. *ArXiv preprint*.

Li, X.; Xiong, H.; Li, X.; Wu, X.; Chen, Z.; and Dou, D. 2022. InterpretDL: Explaining Deep Models in PaddlePaddle. *Journal of Machine Learning Research*, 1–6.

Li, Y.; Yang, J.; Song, Y.; Cao, L.; Luo, J.; and Li, L. 2017b. Learning from Noisy Labels with Distillation. In *Proc. of ICCV*, 1928–1936.

Li, Y.; Yang, J.; Song, Y.; Cao, L.; Luo, J.; and Li, L. 2017c. Learning from Noisy Labels with Distillation. In *Proc. of ICCV*, 1928–1936.

Liu, W.; Hua, G.; and Smith, J. R. 2014. Unsupervised One-Class Learning for Automatic Outlier Removal. In *Proc. of CVPR*, 3826–3833.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *Proc. of ICLR*, 1–11.

Mazumder, M.; Banbury, C. R.; Yao, X.; Karlas, B.; Rojas, W. G.; Diamos, S. F.; Diamos, G.; He, L.; Kiela, D.; Jurado, D.; Kanter, D.; Mosquera, R.; Ciro, J.; Aroyo, L.; Acun, B.; Eyuboglu, S.; Ghorbani, A.; Goodman, E. D.; Kane, T.; Kirkpatrick, C. R.; Kuo, T.; Mueller, J.; Thrush, T.; Vanschoren, J.; Warren, M.; Williams, A.; Yeung, S.; Ardalani, N.; Paritosh, P. K.; Zhang, C.; Zou, J.; Wu, C.; Coleman, C.; Ng, A. Y.; Mattson, P.; and Reddi, V. J. 2022. DataPerf: Benchmarks for Data-Centric AI Development. *ArXiv preprint*.

Nguyen, D. T.; Mummadi, C. K.; Ngo, T.; Nguyen, T. H. P.; Beggel, L.; and Brox, T. 2020. SELF: Learning to Filter Noisy Labels with Self-Ensembling. In *Proc. of ICLR*, 21–29.

Nishi, K.; Ding, Y.; Rich, A.; and Höllerer, T. 2021. Augmentation Strategies for Learning With Noisy Labels. In *Proc. of CVPR*, 8022–8031.

Northcutt, C. G.; Athalye, A.; and Mueller, J. 2021. Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks. In *Proc. of NeurIPS*, 206–217.

Northcutt, C. G.; Jiang, L.; and Chuang, I. L. 2021. Confident Learning: Estimating Uncertainty in Dataset Labels. *J. Artif. Intell. Res.*, 1373–1411.

Patrini, G.; Rozza, A.; Menon, A. K.; Nock, R.; and Qu, L. 2017. Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach. In *Proc. of CVPR*, 2233–2241.

Paul, M.; Ganguli, S.; and Dziugaite, G. K. 2021. Deep Learning on a Data Diet: Finding Important Examples Early in Training. In *Proc. of NeurIPS*, 20596–20607.

Pereyra, G.; Tucker, G.; Chorowski, J.; Kaiser, L.; and Hinton, G. E. 2017. Regularizing Neural Networks by Penalizing Confident Output Distributions. In *Proc. of ICLR*, 11–19.

Pleiss, G.; Zhang, T.; Elenberg, E. R.; and Weinberger, K. Q. 2020. Identifying Mislabeled Data using the Area Under the Margin Ranking. In *Proc. of NeurIPS*, 17044–17056.

Reed, S.; Lee, H.; Anguelov, D.; Szegedy, C.; Erhan, D.; and Rabinovich, A. 2014. Training deep neural networks on noisy labels with bootstrapping. *ArXiv preprint*.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proc. of KDD*, 1135–1144.

Salakhutdinov, R. 2014. Deep learning. In *Proc. of KDD*, 1973.

Saxena, S.; Tuzel, O.; and DeCoste, D. 2019. Data Parameters: A New Family of Parameters for Learning a Differentiable Curriculum. In *Proc. of NeurIPS*, 11093–11103.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proc. of ICLR*, 1–15.

Song, H.; Kim, M.; Park, D.; and Lee, J. 2020. Learning from Noisy Labels with Deep Neural Networks: A Survey. *ArXiv preprint*.

Swayamdipta, S.; Schwartz, R.; Lourie, N.; Wang, Y.; Hajishirzi, H.; Smith, N. A.; and Choi, Y. 2020. Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics. In *Proc. of EMNLP*, 9275–9293.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A. 2017. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *Proc. of AAAI*, 4278–4284.

Toneva, M.; Sordoni, A.; des Combes, R. T.; Trischler, A.; Bengio, Y.; and Gordon, G. J. 2019. An Empirical Study of Example Forgetting during Deep Neural Network Learning. In *Proc. of ICLR*, 1–13.

Vahdat, A. 2017. Toward Robustness against Label Noise in Training Deep Discriminative Neural Networks. In *Proc. of NeurIPS*, 5596–5605.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The caltech-ucsd birds-200-2011 dataset. *California Institute of Technology*.

Wang, Y.; Ma, X.; Chen, Z.; Luo, Y.; Yi, J.; and Bailey, J. 2019. Symmetric Cross Entropy for Robust Learning With Noisy Labels. In *Proc. of ICCV*, 322–330.

Xia, X.; Liu, T.; Han, B.; Gong, C.; Wang, N.; Ge, Z.; and Chang, Y. 2021. Robust early-learning: Hindering the memorization of noisy labels. In *Proc. of ICLR*, 31–39.

Xia, Y.; Cao, X.; Wen, F.; Hua, G.; and Sun, J. 2015. Learning Discriminative Reconstructions for Unsupervised Outlier Removal. In *Proc. of ICCV*, 1511–1519.

Xiao, T.; Xia, T.; Yang, Y.; Huang, C.; and Wang, X. 2015. Learning from massive noisy labeled data for image classification. In *Proc. of CVPR*, 2691–2699.

Yao, J.; Wu, H.; Zhang, Y.; Tsang, I. W.; and Sun, J. 2019. Safeguarded Dynamic Label Regression for Noisy Supervision. In *Proc. of AAAI*, 9103–9110.

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

Zheng, G.; Awadallah, A. H.; and Dumais, S. T. 2021. Meta Label Correction for Noisy Label Learning. In *Proc. of AAAI*, 11053–11061.

Zhou, B.; Lapedriza, À.; Khosla, A.; Oliva, A.; and Torralba, A. 2018. Places: A 10 Million Image Database for Scene Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1452–1464.

Zhou, T.; Wang, S.; and Bilmes, J. A. 2021. Robust Curriculum Learning: from clean label detection to noisy label self-correction. In *Proc. of ICLR*, 41–49.