# Correct Twice at Once: Learning to Correct Noisy Labels for Robust Deep Learning

Jingzheng Li
State Key Lab of Software Development Environment, School of Computer Science and Engineering, Beihang University
Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University
Beijing, China

Hailong Sun†
State Key Lab of Software Development Environment, School of Software, Beihang University
Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University
Beijing, China

## ABSTRACT

Deep Neural Networks (DNNs) have shown impressive performance on large-scale training data with high-quality annotations. However, the collected annotations inevitably contain inaccurate labels in consideration of time and money budget, which causes DNNs to generalize poorly on the test set. To combat noisy labels in deep learning, the label correction methods are dedicated to simultaneously updating model parameters and correcting noisy labels, in which the noisy labels are usually corrected based on model predictions, the topological structures of data, or the aggregation of multiple models. However, such self-training manner cannot guarantee that the direction of label correction is always reliable. In view of this, we propose a novel label correction method to supervise and guide the process of label correction. In particular, the proposed label correction is an online two-fold process at each iteration only through back-propagation. The first label correction minimizes the empirical risk on noisy training data using noise-tolerant loss function, and the second label correction adopts a meta-learning paradigm to rectify the direction of first label correction so that the model can perform optimally in the evaluation procedure. Extensive experiments demonstrate the effectiveness of the proposed method on real-world and synthetic datasets.

## CCS CONCEPTS

• **Computing methodologies** → **Computer vision problems**.

## KEYWORDS

Deep learning, Learning with noisy labels, Meta learning

**ACM Reference Format:**
Jingzheng Li and Hailong Sun†. 2022. Correct Twice at Once: Learning to Correct Noisy Labels for Robust Deep Learning. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3503161.3547861

†Corresponding author.


*MM '22, October 10–14, 2022, Lisboa, Portugal*

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3547861

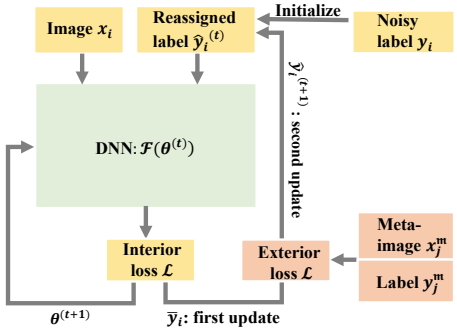


**Figure 1: The concept of our proposal. The label correction objective is formalized as a bilevel optimization problem. At the interior level, we update the model parameters while updating the label distribution for the first time given noisy training data. At the exterior level, we rectify the updated label distribution of the interior level given meta-data.**

## 1 INTRODUCTION

Deep Neural Networks (DNNs) have shown tremendous success in many applications of artificial intelligence, such as image understanding [14, 24, 77], audio/video processing [12, 30], and multimodal deep learning [39, 86]. It is well known that the performance of DNNs highly relies on a large amount of cleanly-annotated training data. However, large-scale and high-quality data is often hard to obtain in consideration of annotation cost and time. Alternatively, crowdsourcing [41] or search engines (automatically collect images related to a query keyword from internet) provide inexpensive approaches to obtain low-quality annotations, but which inevitably contain label noise. Previous works [76, 85] have shown that DNNs with powerful memorization will completely fit the noisy training data resulting in poor generalization over test data. Hence, it is crucial to train DNNs that are robust to noisy labels.

The methods of dealing with noisy labels mainly include designing robust loss functions [15, 47, 48, 75] to learn noise-tolerant models, or removing or down-weighting the mislabeled samples [29, 33, 79] that would impair model training. Although these methods empirically work well for learning robust models, many hard/difficult samples that are important for classification tend to be ignored

during training, which would make some practical applications inferior such as keypoint detection [18, 35]. Thus, in this paper we aim at label correction [38, 66, 84]. In detail, the incorrectly labeled samples are identified and then corrected during training, while model parameters are updated. Identifying mislabeled samples is generally an uncertainty-based measure [49, 62], which can be measured by the confidence of the model output [29, 87] or the discrepancy between multiple model outputs [40, 91]. Then the incorrect labels can be replaced with the reassigned training labels (i.e., pseudo-labels) based on model predictions [37, 40, 70, 78], the topological structures of data [23, 23, 23, 38], or the aggregation of multiple models [50, 59, 67, 92], etc. The representative label correction method is the joint optimization framework [37, 66, 84], which iteratively corrects noisy labels for approximating latent true labels and updates model parameters. Specifically, the update of noisy labels is pseudo-labels based on model predictions under the assumption that the samples with large loss are more likely incorrect labels, and vice versa.

It can be seen that most label correction methods belong to self-training paradigm [74], which cannot guarantee the direction of label correction is always reliable. If mistakes occur in the process of label correction, then these mistakes will be amplified in the subsequent training. Therefore, we consider using a small but clean dataset called meta-data to guide the process of label correction. Although annotating all the data is costly, it is often easy to obtain a small amount of clean labels [3, 25, 57, 61]. Inspired by the meta learning [16, 60] also known as learning to learn, the key idea of our method is: the optimal direction of label correction should be able to minimize the empirical risk not only on corrected training data, but also on meta-data that is consistent with the evaluation procedure. Fig. 1 shows the concept of our proposal. The optimization objective is formulated as a bilevel optimization problem. At the interior level, we *first* correct the labels while updating model parameters given noisy training data. At the exterior level, we *second* rectify the labels derived by the interior level using meta-data. We adopt an online strategy to update label distribution which can be implemented by second-order automatic differentiation. Therefore, our method can be integrated into any neural network structure. Some prior works [29, 38] also use meta-data to rectify the incorrect output of model by training an additional label-cleaning network [68] or evaluating the true noise transition matrix [25]. Nevertheless, the model is prone to overfit on meta-data especially when the amount of meta-data is small, leading to unstable prediction results since the model does not learn the noise patterns sufficiently. In our method, the meta-data is not directly used to update model parameters but to rectify the label correction process to prevent the model from overfitting noisy labels and error accumulation, acting as a kind of regularization on model predictions.

In a nutshell, our contributions can be summarized as follows.

- We propose a novel label correction method, which adopts meta-learning paradigm to guide label correction process so that the model could perform optimally in the evaluation procedure.
- The proposed method updates both the noisy labels and model parameters end to end. Specifically, the label correction is an online two-fold process at each iteration only through back-propagation. Therefore, our method can be integrated into any neural network structure.
- Experimental results show the proposed method achieves higher recovery accuracy than other label correction methods, which verifies the efficacy of meta-data rectification. Further, our method achieves advanced test accuracy on synthetic and real-world datasets with noisy labels.

## 2 RELATED WORK

### 2.1 Learning with Noisy Labels

To our knowledge, robust learning from noisy labels [6, 10, 31, 51, 52, 65] with DNNs can be roughly categorized into four groups. The first is explicit or implicit regularization techniques [4, 5] such as dropout, weight decay or confident penalty [54] sometimes help, but do not directly address nor completely prevent noise label memorization. The second is sample reweighting [7, 29] through up-weighting on clean labels but down-weighting on corrupted labels, which degenerates into sample selection [22] when assigning weights of 0 or 1 to samples. The third [53, 72, 89] aims to modify the loss function for getting rid of the effect of corrupted labels in empirical risk minimization scheme. More recently, Wang et al., [69] reveal the connection between data reweighting and loss function from the perspective of gradient-based optimization. The fourth [9, 64] designs robust network architecture to adapt to the noise behavior in training process of DNNs, including noise adaptation layer [19] to model noise transition matrix [45, 47], or other dedicated architectures [71, 82]. The interested reader is referred to these surveys [1, 21, 32, 58, 63] and the references therein.

### 2.2 Label Correction

In this paper, we focus on correcting noisy labels for robust deep learning. Tanaka et al., [66] propose a joint optimization framework that iteratively updates labels and model parameters. Further, Wang et al., [70] and Wu et al., [78] adopt a convex combination of noisy labels and model predictions for updating label distribution, and different extensions [40, 55] of this strategy have been considered in the manner of semi-supervised learning. Closer to our work, Yi and Wu [84] adopt noise-tolerant KL-divergence between model predictions and noise label distribution to correct noisy labels and update model parameters in a principled end to end manner, using only back-propagation process. Of particular interest, Lee et al., [38] and Han et al., [23] apply the "class prototype" (i.e., representative samples) to represent each class and decide whether the label for a sample is correct or not by comparing it with its associated prototype. We argue that most label correction methods have an inherent deficiency to some extent that the direction of label correction is not always reliable and even the model is over-confident in errors. In view of this, we use a small amount of unbiased meta-data to supervise and guide the label correction process.

### 2.3 Meta Learning for Learning with Noisy Labels

Meta-learning [16], also known as "learning to learn", intends to design models that can learn new skills or adapt to new environments rapidly from one task and generalize well to unseen tasks

proficiently. Meta learning is applied to robust deep learning in the presence of label noise mainly based on two schemes: fast adaption [42, 81] and learning to update [73, 90]. For example, Li et al., [42] aim to obtain model parameters from multiple mini-batches with synthetically corrupted labels that the model can be robust to various noisy labels. Wang et al., [90] combine the advantages of the sample re-weighting and label correction through training a meta learner that tries to correct the noisy labels and a main model that tries to build the best predictive model. Inspired by MAML [16], using validation loss as the meta-objective to learn how to reweight samples has been explored [46, 57, 61, 80, 90]. Our method is similar to these works that take one gradient descent step on the meta-objective at each iteration. However our method is designed to update parameters for achieving the optimal label distribution rather than optimal weights of samples. Closer to our work, Algan et al., [3] directly correct noisy label distribution by using the meta-objective sharing a similar objective with our method. However, our method differs from their work in that our method goes through a two-fold correction on label distribution, which takes advantage of both the manner of self-training and meta-learning to ensure the robustness of model training.

# 3 METHOD

## 3.1 Notation and Problem Formulation

Let $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\} = \{(\mathbf{x}_i, y_i), 1 \le i \le N\}$ be the training data, in which $(\mathbf{x}_i, y_i)$ is the $i$-th input-label pair, and $y_i \in \{0,1\}^c$ is a one-hot vector representing the corresponding noisy label over $c$ classes. We assume that there is a small amount of unbiased meta-data (i.e., with clean and balanced label distribution) $\left\{\left(\mathbf{x}_i^{\mathfrak{m}}, y_i^{\mathfrak{m}}\right), 1 \le i \le M\right\}$, and $M \ll N$, in which the superscript $\mathfrak{m}$ denotes the data is drawn from meta-data distinguished from the training data. A DNN model $\mathcal{F}(\theta)$ with parameter $\theta$ is defined to transfer the input data to label probability distribution $\mathcal{F}(\theta, \mathbf{x}_i)$ with softmax function processed. In standard training, an optimization objective over training data is defined as

$$\theta^* = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathcal{F}(\theta, \mathbf{x}_i), y_i) \tag{1}$$

where the $\mathcal{L}$ is the empirical risk such as Cross Entropy. Then the model parameter is updated to optimize the Eq. 1 by using SGD or its variants. However, since the provided labels contain noise, the DNN would overfit to noisy labels and perform poorly on unseen test data.

To fight against DNNs' memorization on noisy labels, most previous works focus on adjusting the term in Eq. 1 for achieving optimal generalization ability. Regarding the label correction methods with DNN, they not only update model parameters as in standard training, but also update labels for approximating latent true labels. Hence, the objective to simultaneously optimize model parameters and labels is formulated as

$$\mathbf{Y}^*, \theta^* = \arg\min_{\hat{\mathbf{Y}}, \theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathcal{F}(\theta, \mathbf{x}_i), \hat{y}_i), \tag{2}$$

in which the $\hat{\mathbf{Y}}$ represents updated labels, initialized from original noisy labels. Concretely, Tanaka et al., [66] directly use model predictions $\mathcal{F}(\theta, \mathbf{x}_i)$ to replace label distribution, and iteratively update model parameters under the framework of joint optimization. Yi et al., [84] adopt label probability distributions to supervise model training and to update labels distribution and model parameters through back-propagation end-to-end.

As a manner of updating labels, two ways can be considered: the one-hot label and the soft label distribution. Knowledge distillation [26] and label distribution learning [17] have shown that soft label distribution could extract the *dark knowledge* contained in data that the one-hot label cannot do. Unless otherwise specified, we use soft label distribution $\hat{\mathbf{Y}} = \left\{\hat{y}_i : \hat{y}_i \in [0,1]^c, \mathbf{1}^\top \hat{y}_i = 1, 1 \le i \le N\right\}$, from which $\mathbf{1}$ is a vector of all-ones, representing the estimation of true labels.

## 3.2 Proposed Label Correction Objective

In this study, we use the meta-data to guide the label correction process. The key idea of our learning objective is: the optimal direction of label correction based on model predictions should be able to minimize the empirical risk on meta-data that is consistent with evaluation procedure. With meta-data, the goal of interest is to rectify label distribution derived by model predictions and update model parameters so that the model performs well on meta-data. Formally, we propose to optimize

$$\begin{aligned} \bar{\mathbf{Y}}, \hat{\theta} &= \arg\min_{\hat{\mathbf{Y}}, \theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathcal{F}(\theta, \mathbf{x}_i), \hat{y}_i) \\ \text{s.t.} \quad \hat{\mathbf{Y}} &= \arg\min_{\bar{\mathbf{Y}}} \frac{1}{M} \sum_{j=1}^{M} \mathcal{L}\left(\mathcal{F}\left(\bar{\theta}, \mathbf{x}_j^{\mathfrak{m}}\right), y_j^{\mathfrak{m}}\right) \end{aligned} \tag{3}$$

in which $\bar{\mathbf{Y}}$ represents the first update of label distribution derived by model predictions, through which we obtain the temporary model parameter $\bar{\theta}$ that will be discussed in the next subsection, and $\hat{\theta}$ and $\hat{\mathbf{Y}}$ denote updated model parameters and second updated labels distribution respectively.

We can see that the objective is bilevel optimization [27, 46]. Specifically, the interior level is set to update model parameters for minimizing the empirical risk on noisy training data while first updating label distribution $\bar{\mathbf{Y}}$ based on model predictions in the manner of self-training. In the exterior level, we minimize the empirical risk on the meta-data through access to temporary model parameters $\bar{\theta}$ for updating label distribution $\hat{\mathbf{Y}}$. Note that the exterior level does not have access to the label distribution of noisy training data. A classical method [13] to solve the above bilevel optimization problem is to solve a linear system in the second-order derivative of the exterior level objective, which can be very expensive in the nested loops. In the next subsection, we instead adopt an online strategy to update labels, to guarantee the efficiency of our method.

## 3.3 Optimization

For most training of DNNs, SGD or its variants are used to optimize empirical risk. At every iteration of training, a mini-batch of training data $\{(\mathbf{x}_i, y_i), 1 \le i \le n\}$ is sampled, where $n$ is the mini-batch size. For notation convenience, we use $\mathcal{L}(\theta, y_i)$ to denote $\mathcal{L}(\mathcal{F}(\theta, \mathbf{x}_i), y_i)$.

At the $t$-th iteration, regarding the interior level, given the currently updated label distribution $\hat{\mathbf{Y}}^{(t)}$, let's consider vanilla SGD with learning rate $\lambda$ by using mini-batches from training data to update model parameters as follows.

$$\theta^{(t+1)} = \theta^{(t)} - \lambda \frac{1}{n} \sum_{i=1}^{n} \nabla_{\theta} \mathcal{L}\left(\theta^{(t)}, \hat{y}_i^{(t)}\right). \tag{4}$$

Meanwhile, following Yi et al., [84] that utilizes back-propagation to update label distribution, we also take an SGD step with label update step $\gamma$ to *first* update the label distribution $\bar{\mathbf{Y}}$ by using certain noise-tolerant loss function between model predictions and current label distribution, yielding

$$\bar{\mathbf{Y}} = \hat{\mathbf{Y}}^{(t)} - \gamma \frac{1}{n} \sum_{i=1}^{n} \nabla_{\hat{\mathbf{Y}}} \mathcal{L}\left(\theta^{(t)}, \hat{y}_i^{(t)}\right). \tag{5}$$

We here slightly abuse notations $\bar{\mathbf{Y}}$ and $\hat{\mathbf{Y}}$ in mini-batch training, which contain label distribution in current mini-batch.

Next, for exterior level, we use meta-data to rectify first label distribution $\bar{\mathbf{Y}}$ derived by model predictions so that the empirical risk of model over meta-data can be minimized. However, the objective of exterior level in Eq. 3 does not have access to the label distribution of noisy training data. Instead, we would like to understand what would be the impact of first label distribution $\bar{\mathbf{Y}}$ across mini-batches of training data towards the performance of the meta-data at training step. Hence, we calculate, rather than update, the temporary model parameters over first label distribution $\bar{\mathbf{Y}}$ for evaluating the performance of meta-data, which is obtained by

$$\bar{\theta} = \theta^{(t)} - \lambda \frac{1}{n} \sum_{i=1}^{n} \nabla_{\theta} \mathcal{L}\left(\theta^{(t)}, \bar{y}_i\right). \tag{6}$$

Unfortunately, this can still be quite time-consuming, i.e., updating the label distribution of the current training mini-batch requires assessing the empirical risk of all meta-data in each training mini-batch. To get an efficient estimate of meta-data at current training mini-batch, we take a single mini-batch from meta-data such as $\left\{\left(\mathbf{x}_j^{\mathrm{m}}, y_j^{\mathrm{m}}\right), 1 \leq j \leq m\right\}$. Finally, we rectify label distribution derived by model predictions to minimize the empirical risk of mini-batch of meta-data. To this end, we adopt an online approximate strategy such that it can be approximated through a first-order Tayler expansion based on current model parameters, yielding

$$\mathcal{L}\left(\bar{\theta}, y_j^{\mathrm{m}}\right) \simeq \mathcal{L}\left(\theta^{(t)}, y_j^{\mathrm{m}}\right) + \nabla_{\bar{\theta}} \mathcal{L}\left(\theta^{(t)}, y_j^{\mathrm{m}}\right)^{\top} \left(\bar{\theta} - \theta^{(t)}\right). \tag{7}$$

Since the above Taylor expansion holds only in the proximity of the model parameters $\theta^{(t)}$, we thus set a relatively low learning rate in Eq. 6. By plugging Eq. 6 into Eq. 7 and discarding constant term which is not relevant to the optimization procedure, the exterior level objective becomes

$$\hat{\mathbf{Y}} = \arg\min_{\bar{\mathbf{Y}}} \sum_{j=1}^{m} \sum_{i=1}^{n} -\nabla_{\bar{\theta}} \mathcal{L}\left(\theta^{(t)}, y_j^{\mathrm{m}}\right)^{\top} \nabla_{\theta} \mathcal{L}\left(\theta^{(t)}, \bar{y}_i\right). \tag{8}$$

The resulting update scheme of labels is quite intuitive: the direction of the label update is to maximize its similarity with meta-data. We can then take an SGD step of Eq. 8 to *second* update label distribution

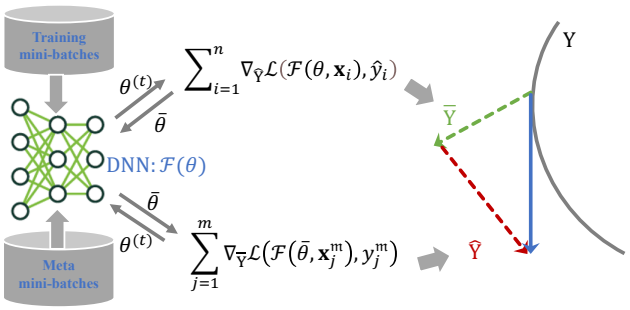


**Figure 2: The label correction process of our bilevel optimization. At each iteration we draw mini-batches from the noisy training data and meta-data, respectively. As we can see, the first label correction is based on model predictions on noisy training data, and the second label correction is a rectification of the first label correction using meta-data.**

w.r.t $\bar{\mathbf{Y}}$, yielding:

$$\hat{\mathbf{Y}}^{(t+1)} = \bar{\mathbf{Y}} + \gamma \sum_{j=1}^{n} \nabla_{\bar{y}_j} \left( \sum_{i=1}^{m} G_j(\bar{y}_i) \right) \tag{9}$$

in which $G_j(\bar{y}_i) = \nabla_{\bar{\theta}} \mathcal{L}\left(\theta^{(t)}, y_j^{\mathrm{m}}\right)^{\top} \nabla_{\theta} \mathcal{L}\left(\theta^{(t)}, \bar{y}_i\right)$. The above update scheme can be efficiently implemented using second-order automatic differentiation. Thus, this implementation can be generalized to any deep learning architectures.

An overview of the proposed label correction process is presented in Figure 2. By observing the label update scheme of Eq. 9, we can draw an intuitive conclusion: for each training iteration, if the gradient of current training mini-batch is agree with the gradient of corresponding meta mini-batch, i.e., $G_j(\bar{y}_i) > 0$, then we encourage the update of label distribution in its gradient direction, and conversely, if those two gradients are orthogonal or in the opposite directions, i.e., $G_j(\bar{y}_i) \leqslant 0$, then we suppress the update of label distribution in its gradient direction.

## 3.4 Loss Function

In the previous subsection, we present the label correction objective and the corresponding online optimization strategy. Next, we propose to use suitable loss functions for optimization.

For interior level of the optimization objective, following Yi and Wu [84], since KL-divergence is an asymmetric function, we exchange the two operands of Kl-divergence between the updated label distribution and model output $\mathcal{F}(\theta, \mathbf{x}_i)$ over $C$ classes. Thus, we obtain a new loss function

$$\mathcal{L}_{\mathrm{KL}}\left(\mathcal{F}\left(\theta; \mathbf{x}_i\right), \hat{y}_i\right) = \sum_{c=1}^{C} \mathcal{F}_c\left(\theta; \mathbf{x}_i\right) \log \left( \frac{\mathcal{F}_c\left(\theta; \mathbf{x}_i\right)}{\hat{y}_{ic}} \right). \tag{10}$$

The KL-divergence loss function in Eq. 10 can be used for combating label noise, which is also confirmed by Wang et al., [72]. Besides, as for updating model parameters in Eq. 4, we define two additional regularization terms, i.e., entropy minimization [20, 54] and label smoothing [40]. Specifically, when we update the soft label distribution, the model may be stuck in local optima such that

the learning process does not proceed. The entropy minimization enforces DNN outputs to peak only at one class and zero out on others. The entropy minimization for model predictions is defined as

$$\mathcal{L}_{\mathrm{E}}(\mathcal{F}(\theta;\mathbf{x}_i)) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{c=1}^{C}\mathcal{F}_c(\theta;\mathbf{x}_i)\log\mathcal{F}_c(\theta;\mathbf{x}_i). \tag{11}$$

The label smoothing is proposed to prevent the assignment of all labels to a single class for obtaining a trivial global optimal solution, which applies a uniform prior distribution $\pi = \{\pi_c : \pi_c = 1/C, 1 \le i \le C\}$ to regularize the model's average output, formulated as

$$\mathcal{L}_{\mathrm{S}}(\mathcal{F}(\theta;\mathbf{x}_i)) = \sum_{c=1}^{C}\pi_c \log\frac{\pi_c}{\frac{1}{n}\sum_{i=1}^{n}\mathcal{F}_c(\theta;\mathbf{x}_i)}. \tag{12}$$

Therefore, the loss function for updating model parameters in Eq. 4 is

$$\mathcal{L}(\mathcal{F}(\theta;\mathbf{x}_i),\hat{y}_i) = \mathcal{L}_{\mathrm{KL}} + \alpha\mathcal{L}_{\mathrm{E}} + \beta\mathcal{L}_{\mathrm{S}} \tag{13}$$

in which $\alpha$ and $\beta$ are hyper-parameters that control the strength of regularization terms.

For exterior level of the optimization objective, since the labels of meta-data are clean, we use the classic Cross Entropy as loss function to second update the label distribution in Eq. 9. In doing so, the updated label distribution $\hat{\mathbf{Y}}$ gradually approaches latent true labels from original noisy labels as the training proceeds. Alg. 1 delineates the steps of the label correction in our method. Since original labels from training data are one-hot label, it could cause computational problem in KL-divergence: zero values inside the logarithm. To address this issue, we initialize original noisy labels $\mathbf{Y}$ with the following formula to ensure that initialized label distribution is normalized to a probability distribution.

$$\hat{\mathbf{Y}} = \mathrm{softmax}(K\mathbf{Y}), \tag{14}$$

where $K$ is a relatively large constant ($K = 10$ in experiments [84]).

**Algorithm 1:** The label correction stage

**Input:** Training data $\{(\mathbf{x}_i, y_i), 1 \le i \le N\}$, meta-data $\left\{\left(\mathbf{x}_i^{\mathrm{m}}, y_i^{\mathrm{m}}\right), 1 \le i \le M\right\}$, and a DNN model $\mathcal{F}(\theta)$

**Output:** The trained network model $\mathcal{F}(\hat{\theta})$ and the corrected labels $\hat{\mathbf{Y}}$

initialize $\hat{\mathbf{Y}}$ by Eq. 14;
**for** *t=1 to T* **do**
  draw a mini-batch $\{(\mathbf{x}_i, y_i), 1 \le i \le B\}$ from training data;
  draw a mini-batch $\left\{\left(\mathbf{x}_i^{\mathrm{m}}, y_i^{\mathrm{m}}\right), 1 \le i \le B\right\}$ from meta-data;
  *first* update label distribution $\{\bar{y}_i, 1 \le i \le B\}$ as Eq. 5 by using loss function $\mathcal{L}_{\mathrm{KL}}$;
  *second* update label distribution $\{\hat{y}_i, 1 \le i \le B\}$ as Eq. 9 by using Cross Entropy loss;
  update model parameter $\hat{\theta}$ as Eq. 4 by using loss function $\mathcal{L}_{\mathrm{KL}} + \alpha\mathcal{L}_{\mathrm{E}} + \beta\mathcal{L}_{\mathrm{S}}$;
**end**

## 3.5 Training Scheme

To implement our method, the training scheme consists of three stages.

1) Warm-up training: at the beginning, we conduct standard training on training data so that DNN can learn discriminative ability to some extent. If there is no warm-up training, then the random output from DNN would lead to the correction of label distribution in a completely wrong direction.
2) Noisy labels correction training: secondly, on the basis of the warmed-up DNN model, we use the proposed label correction method to update both model parameters and label distribution.
3) Fine-tuning training: finally, when the noisy label correction is complete in the second stage, we use the fixed label distribution to fine-tune the DNN model using only noise-tolerant Kl-divergence loss function without regularization terms.

# 4 EVALUATION

## 4.1 Datasets and Baselines

*4.1.1 Synthetic Datasets.* The two CIFAR datasets consist of 32 × 32 colored natural images with 10 classes (CIFAR-10) and 100 classes (CIFAR-100), respectively. Both datasets contain 60k images with clean labels. We separate 5k images for testing and another 5k as meta-data (validation data). The remaining 50k images are corrupted with synthetic label noise.

We generate two types of label noise: *uniform noise* and *feature-dependent noise*. For uniform noise, we randomly flip a correct label to one of the other incorrect labels uniformly. The feature-dependent noise has recently gained considerable interests [8, 11, 83, 88]. It is more realistic and challenging since the noisy labels depend on both true labels and data features. In terms of implementation, a pre-trained network is used as feature extractor; then those labels of images are flipped to labels of their neighbor images with similar features in other classes. For detailed descriptions, we refer the readers to Algan and Ulusoy [2].

*4.1.2 Real-World Dataset.* Clothing1M is a large-scale dataset which consists of 1 million clothing images obtained from online shopping websites with 14 classes. The labels in this dataset are extremely noisy (with an estimate accuracy of 61.54%) and their structure is unknown. This dataset is seriously imbalanced and the label mistakes mostly happen in those images whose content is similar to other classes. And it also provides 50k, 14k, 10k clean data for training, validation, and testing respectively.

*4.1.3 Baselines.* Our method is compared with diverse baselines, including robust loss function, sample selection, sample reweighting, and three representative label correction methods as follows.

1) Cross Entropy: standard training with cross entropy.
2) Bootstrap [56]: a bootstrapping method whose loss function is a combination of noisy labels and model predictions. The soft version is used and the hyper-parameter is tuned in $\{0.05, 0.2, 0.5\}$.
3) Co-Teaching [22]: a method that trains two deep neural networks, and back-propagates the sample selected by its peer network and updates itself.

**Table 1: Experimental comparison on test accuracy (%). The test accuracy of the best during training (left) and the last epoch (right) are listed on CIFAR-10 and CIFAR-100 injected with two kinds of noise. The best results are in boldface.**

**(a) Results on CIFAR-10 dataset**

| Methods | Clean data | Uniform noise | | | Feature-dependent noise | | |
|---|---|---|---|---|---|---|---|
| | | $\delta = 0.2$ | $\delta = 0.4$ | $\delta = 0.6$ | $\delta = 0.2$ | $\delta = 0.4$ | $\delta = 0.6$ |
| Cross Entropy | **91.12**/90.92 | 85.08/79.32 | 79.76/67.34 | 69.14/44.09 | 81.45/80.36 | 71.74/67.02 | 69.40/63.22 |
| Bootstrap [56] | 90.26/90.24 | 84.66/77.60 | 79.90/63.86 | 70.66/43.52 | 82.20/80.82 | 72.40/66.06 | 70.08/61.82 |
| Meta-Net [61] | 90.64/90.64 | 86.50/84.24 | 82.12/80.18 | 74.98/70.78 | 81.22/80.74 | 71.84/68.17 | 68.40/62.26 |
| Co-Teaching [22] | 87.44/87.26 | 87.78/87.77 | 81.85/77.48 | 71.53/62.72 | 82.56/82.25 | 72.69/70.87 | 62.44/43.40 |
| Joint-Opt [66] | 90.96/90.46 | 85.76/79.90 | 81.81/67.14 | 71.86/47.68 | 82.36/80.42 | 75.12/67.30 | 72.22/63.94 |
| MSLG [3] | 88.25/87.44 | 83.99/82.62 | 80.09/76.93 | 71.94/68.80 | 82.92/82.40 | 82.78/82.19 | 78.94/77.64 |
| PENCIL [84] | 90.34/90.22 | 87.70/87.62 | 84.26/84.22 | 77.86/77.72 | 83.14/83.10 | 75.46/75.36 | 67.12/60.70 |
| Ours | 90.40/90.40 | 87.94/**87.94** | 84.90/**84.90** | **80.10**/80.08 | 85.12/**85.12** | 82.08/**82.08** | 80.11/**80.11** |

**(b) Results on CIFAR-100 dataset**

| Methods | Clean data | Uniform noise | | | Feature-dependent noise | | |
|---|---|---|---|---|---|---|---|
| | | $\delta = 0.2$ | $\delta = 0.4$ | $\delta = 0.6$ | $\delta = 0.2$ | $\delta = 0.4$ | $\delta = 0.6$ |
| Cross Entropy | 62.52/62.46 | 54.89/52.18 | 46.26/43.05 | 36.48/34.52 | 49.58/49.51 | 36.92/36.77 | 24.87/24.26 |
| Bootstrap [56] | 63.86/63.78 | 53.26/48.30 | 45.58/35.36 | 35.18/22.50 | 48.94/48.70 | 36.16/35.98 | 25.28/25.22 |
| Meta-Net [61] | 63.16/62.96 | 53.84/53.66 | 48.68/46.04 | 37.78/33.98 | 49.28/49.14 | 35.06/35.06 | 24.66/22.16 |
| Co-Teaching [22] | 61.82/61.68 | **60.51**/60.19 | 52.28/50.65 | 41.79/38.66 | 48.55/48.43 | 33.13/32.98 | 20.12/20.12 |
| Joint-Opt [66] | 62.08/62.06 | 56.46/53.60 | 49.48/45.32 | 39.60/35.36 | 50.12/49.84 | 37.84/37.78 | 26.44/26.04 |
| MSLG [3] | 59.08/59.08 | 52.34/52.02 | 45.12/45.06 | 34.28/34.28 | 49.02/49.02 | 34.78/34.78 | 21.30/20.70 |
| PENCIL [84] | **63.86**/63.74 | 56.14/56.10 | 48.92/48.90 | 36.52/36.40 | 48.72/48.60 | 32.76/30.56 | 22.80/19.14 |
| Ours | 63.08/62.96 | 58.92/58.84 | **52.78**/52.54 | **42.70**/42.62 | **53.96**/53.64 | **47.56**/47.18 | **35.44**/35.44 |

4) Meta-Net [61]: a meta-learning method that extracts sample weights adaptively by a weight net. Similar to our method, the weight net parameters are guided by meta-data.
5) MSLG [3]: a meta-learning method that directly corrects label distribution by using validation loss as the meta-objective.
6) Joint-Opt [66]: a joint optimization framework that can correct labels by alternately updating model parameters and labels.
7) PENCIL [84]: an end-to-end framework that can update both model parameters and labels distribution through back-propagation process.

The implementations of competing methods are based on our own implementations according to their open-sourced codes. To have a fair comparison, we use the same training data, backbone network architecture, and we perform the same training epochs.

## 4.2 Evaluation on CIFAR Datasets

*4.2.1 Setup Details.* Following previous works [22, 36], we here use an 8-layer CNN architecture with 6 convolutional layers and 2 fully connected layers for CIFAR datasets. We use SGD optimizer with a momentum of 0.9 and weight decay of 1e-4. The batch sizes of training data and meta-data are both 128. The numbers of epoch for three steps are 40, 60, and 20. We begin training with an initial learning rate of 0.1, which is divided by 10 at 50% and 90% of the total number of epochs, and we set $\alpha = 1$ and $\beta = 0.1$ via a grid search within the range of $[0, 1]$. Regarding the label update step $\gamma$, we set $\gamma = 400$ for CIFAR-10 and $\gamma = 4000$ for CIFAR-100 (see Section 4.4 for the analysis of hyper-parameter sensitivity). Note that although we choose the same hyper-parameter settings in most cases, we can achieve better results by further tuning the hyper-parameters.

*4.2.2 Performance Comparison.* We first evaluate test accuracy of our method on noisy versions of CIFAR-10 and CIFAR-100 compared with other baselines. Jiang et al., [28] point out that early stopping is not always effective in deep learning with noisy labels, although some previous studies only report the best results. Therefore, we report not only the optimal test score but also the test score at the end of training, to see the robustness of methods under label noise, which are listed in Table 1.

We draw some conclusions from Table 1. First, our method outperforms other baselines in terms of uniform noise and feature-dependent noise, except for Co-Teaching on CIFAR-100 with 20% uniform noise where our method also obtains competitive results. In particular, when the feature-dependent noise is severe, the performance of other baselines yields a big drop, while our method still performs relatively well. Our proposal is also applicable to datasets with clean labels. Second, our method is also very robust, and there is no big drop between the best score and the last score. For example, on the CIFAR-10 dataset, although both our method and MSLG use meta-data to correct the label distribution, our method outperforms MSLG in terms of robustness and test accuracy. Last but not least, for most comparative baselines, a larger drop between the last score and the best score on uniform noise than that on feature-dependent

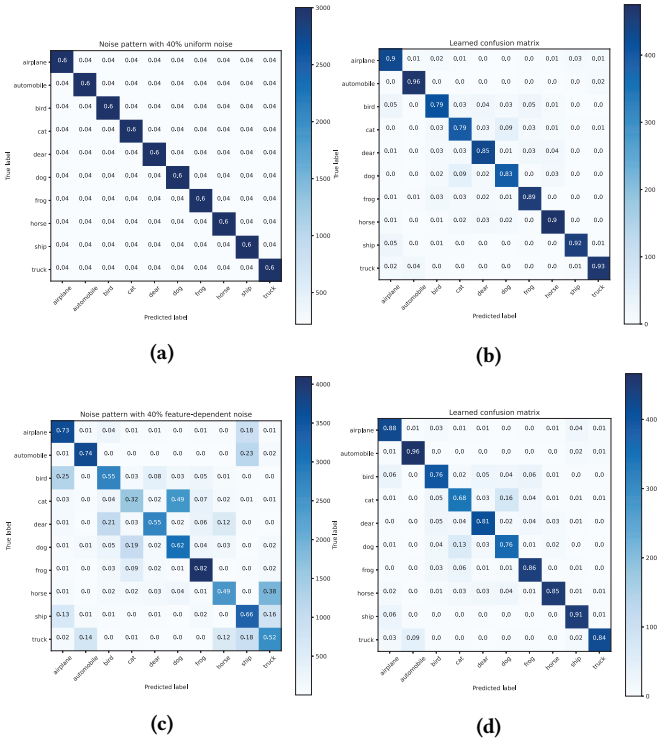


**Figure 3: A visualization of the confusion matrices on CIFAR-10. (a) and (c) denote noise patterns on training data with 40% uniform noise and feature-dependent noise, respectively. (b) and (d) denote the recovery confusion matrices on test data learned by our method.**

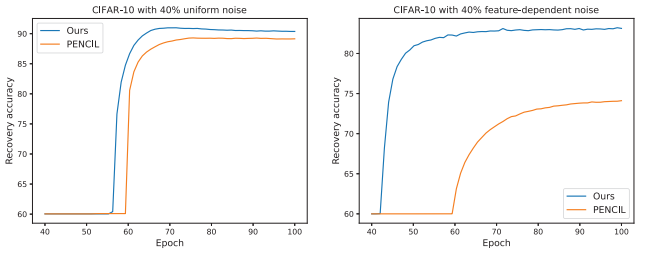


**Figure 4: Comparison between our method and the PENCIL, showing the recovery accuracy.**

noise suggests that 1) DNNs can learn a better generalizable "pattern" under uniform noise during the early training stage; 2) the feature-dependent noise is more complicated than uniform noise because they are more relevant (visually or semantically) to the associated clean images. Therefore, it is more difficult for DNNs to capture meaningful patterns automatically. This conclusion is consistent with the finding of Jiang et al., [28].

Figure 3 shows the pre-defined noise patterns and the confusion matrices learned by our method. It shows that our method could prevent model from overfitting noise pattern during training and achieve good generalization performance on test data.

**Table 2: Comparison with state-of-the-art methods in test accuracy (%) on Clothing1M using ResNet-50. The star$^\star$ indicates the results are taken from original papers.**

| Methods | Accuracy | Methods | Accuracy |
|---|---|---|---|
| Cross Entropy | 70.86 | CleanNet$^\star$ [38] | 74.69 |
| Bootstrap [56] | 71.15 | DivideMix$^\star$ [40] | 74.76 |
| Meta-Net$^\star$ [61] | 73.72 | TopoFiler$^\star$ [79] | 74.10 |
| Co-Teaching [22] | 72.15 | CORES$^{2\star}$ [11] | 73.24 |
| Joint-Opt$^\star$ [66] | 72.23 | MLC$^\star$ [90] | 75.78 |
| MSLG$^\star$ [3] | 76.02 | PLC$^\star$ [88] | 74.02 |
| PENCIL$^\star$ [84] | 73.49 | Ours | **77.38** |

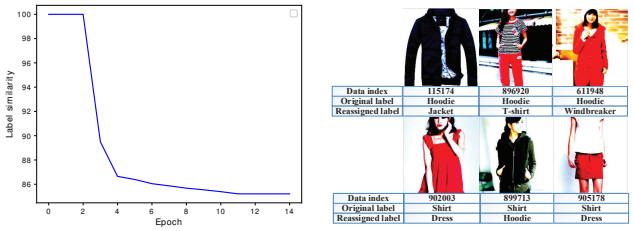


**Figure 5: Left: the variation of label similarity on noisy training data during training. Right: the images with incorrect labels such as two categories shirt and hoodie are assigned the correct labels by our method.**

To examine the effectiveness of the label correction during training, we show the *recovery accuracy* of noisy labels correction stage in training process. The recovery accuracy represents the accuracy of the reassigned training labels. Figure 4 depicts the behaviors between our method and PENCIL on the CIFAR10 in recovery accuracy. Actually, our method uses meta-data to rectify the label distribution derived by PENCIL. Therefore, the comparison between our method with PENCIL can be seen as an ablation study to test the efficacy of mete-data rectification. Figure 4 shows that our method achieves higher recovery accuracy than PENCIL does, which accounts for the reason why the better test accuracy our method achieves.

## 4.3 Evaluation on Clothing1M Dataset

*4.3.1 Setup Details.* For the Clothing1M, We use ResNet-50 with ImageNet pre-training as the backbone network. Following previous work [79, 84], we use a randomly sampled pseudo-balanced subset as a training set including about 270k images. For preprocessing, we resize the images to $256 \times 256$, perform mean subtraction, and crop the middle $224 \times 224$. We use SGD with a momentum of 0.9 and a weight decay of 1e-3. The batch sizes of training data and meta-data are both 32. The numbers of epoch for three steps are 2, 10, and 3. We begin training with a learning rate of 1e-3, which is divided by 10 at second stage and third stage respectively. We set $\alpha = 0.1$, $\beta = 0.1$ and $\gamma = 2000$. In addition to the compared baselines on CIFAR datasets, we also compare other advanced methods on Clothing1M. Note that the MLC and Meta-Net use 50K and 7K additionally clean training data as meta-data, respectively. The MSLG and our method use 14K clean validation data as meta-data.

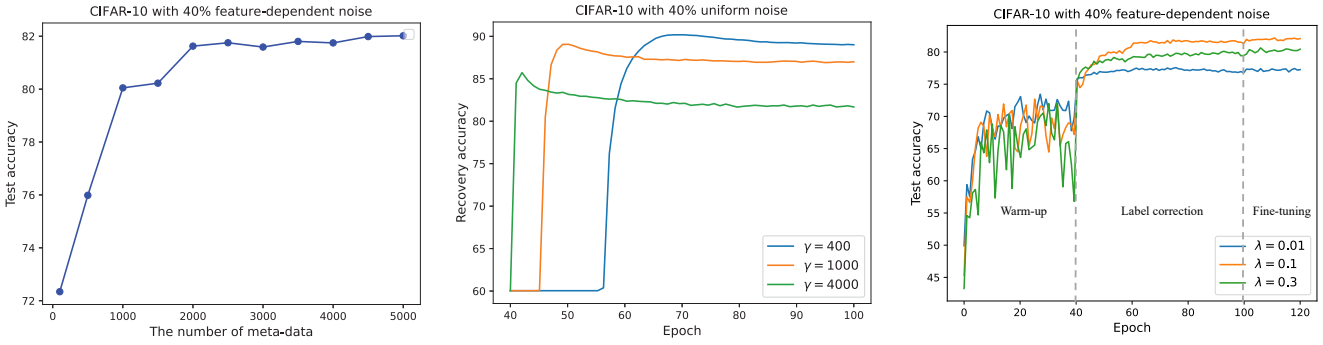


**Figure 6: Plot of various hyper-parameters analysis on our method. (a) Variation in test accuracy for different numbers of meta-data. (b) Effect of varying label update step $\gamma$ as training proceeds. (c) Effect of varying learning rate $\lambda$ as training proceeds.**

*4.3.2 Performance Comparison.* As shown in Table 2, our method maintains its dominance over other baselines on realistic noise and exceed the other baselines of 1-6% in test accuracy. Specifically, our method is intended to rectify the label distribution derived by PENCIL, and reports a improvement of 3.5% compared to PENCIL, which demonstrates the efficacy of using meta-data to guide the process of label correction.

In addition, in Figure 5 (left), we show the variation of *label similarity* on noisy training data of Clothing1M as training proceeds. Label similarity indicates the overall change ratio of all the original labels to the labels being reassigned by our method. It can be seen that around 14% of the labels are modified by our method (the true labels of the noisy training data is unknown). More specifically, Figure 5 (right) shows several examples where mislabeled labels are reassigned by our method to the correct labels. Note that here the reassigned labels are *hard* labels obtained from the largest score of model predictions.

## 4.4 Effect of Hyper-parameters

We first examine our method by varying the number of meta-data, to evaluate how much meta-data can make our method effective. As can be seen in Figure 6a, the test accuracy improves as the number of meta-data increases on CIFAR-10 with 40% feature-dependent noise. Meanwhile, we observe that 1K meta-data (2% of 50k noisy training data) allows our method to achieve approximately satisfactory result. Although our method requires a small amount of unbiased data as meta-data, the meta-data required is very small usually less than 2% of noisy training data and is easily available in many cases.

We further evaluate the hyper-parameter sensitivity of our method in terms of label update step $\gamma$ and learning rate $\lambda$ (i.e., parameter update step). Figure 6b shows that the variation of recovery accuracy on CIFAR-10 with 40% uniform label noise by varying the label update step $\gamma$ in $\{400, 1000, 4000\}$. Analogously, we vary the initial learning rate $\lambda$ from a lower value of 0.01 to a higher value of 0.3 in $\{0.01, 0.1, 0.3\}$ to observe the behavior of test accuracy during training on CIFAR-10 with 40% feature-dependent label noise, which is shown in Figure 6c.

We take a closer look at how $\gamma$ and $\lambda$ affect the performance of our method, respectively. We can conclude that: 1) Regarding label update step $\gamma$, the size of $\gamma$ determines the speed and accuracy of label correction. The larger the value of $\gamma$, the faster the labels update but the lower the recovery accuracy. Thus, on the trade-off of recovery accuracy and update speed, we can take an appropriate value of $\gamma$ within a limited number of training iterations. 2) For the learning rate $\lambda$, when the learning rate $\lambda$ is relatively small such as 0.01, the test accuracy is relatively high in warm-up training stage but performs not good in the label correction stage. Then the learning rate $\lambda$ is increased to 0.1, compared to $\lambda = 0.01$, despite the performance drops in the warm-up training stage, our method eventually achieves superior performance in the label correction stage. Continuing to increase learning rate $\lambda = 0.3$ would make the learning ability of model insufficient in warm-up training stage, resulting in limited performance improvement in subsequent label correction stage. This means that a relatively high $\lambda$, but not too high, could alleviate the overfitting of the model to noisy labels in the warm-up training [66].

## 5 CONCLUSIONS, LIMITATIONS AND FUTURE RESEARCH

In this paper, we aim at correcting noisy labels for robust deep learning. Specifically, a small amount of unbiased meta-data is utilized to supervise and rectify the label correction process end-to-end using only back propagation so that the model performs optimally in the evaluation procedure. Experimental results illustrate that our method outperforms other baselines on noisy version of CIFAR datasets in terms of test accuracy and robustness. Further, on real-world Clothing1M dataset, our method performs the best.

In some fields such as medical images, clean data annotations are often difficult to obtain, even a very small amount of meta-data. On the other hand, a small amount of specific meta-data may not comprehensively represent the "class prototype", which potentially reflects a distribution shift [34] between meta-data and evaluation procedure. All these factors may limit the applicability of our method. In the future, we will explore how to extract "meta-representation" [43, 44] from large-scale noisy training data itself by self-supervised learning to guide label correction process.

## ACKNOWLEDGMENTS

This work was support by National Natural Science Foundation of China under Grant Nos.(61932007, 61972013, 62141209).

## REFERENCES

[1] Görkem Algan and Ilkay Ulusoy. 2019. Image classification with deep learning in the presence of noisy labels: A survey. *arXiv preprint arXiv:1912.05170* (2019).
[2] Görkem Algan and İlkay Ulusoy. 2020. Label Noise Types and Their Effects on Deep Learning. *arXiv preprint arXiv:2003.10471* (2020).
[3] Görkem Algan and Ilkay Ulusoy. 2020. Meta Soft Label Generation for Noisy Labels. *arXiv preprint arXiv:2007.05836* (2020).
[4] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. *arXiv preprint arXiv:1706.05394* (2017).
[5] Samaneh Azadi, Jiashi Feng, Stefanie Jegelka, and Trevor Darrell. 2015. Auxiliary image regularization for deep cnns with noisy labels. *arXiv preprint arXiv:1511.07069* (2015).
[6] Oshrat Bar, Amnon Drory, and Raja Giryes. 2020. A Spectral Perspective of Neural Networks Robustness to Label Noise. (2020).
[7] Antonin Berthon, Bo Han, Gang Niu, Tongliang Liu, and Masashi Sugiyama. 2021. Confidence scores make instance-dependent label-noise learning possible. In *International Conference on Machine Learning*. PMLR, 825–836.
[8] Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng. 2020. Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise. *arXiv preprint arXiv:2012.05458* (2020).
[9] Xinlei Chen and Abhinav Gupta. 2015. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 1431–1439.
[10] Yifang Chen, Simon S Du, and Kevin G Jamieson. 2021. Corruption Robust Active Learning. *Advances in Neural Information Processing Systems* 34 (2021).
[11] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. 2020. Learning with Instance-Dependent Label Noise: A Sample Sieve Approach. In *International Conference on Learning Representations*.
[12] Shangzhe Di, Zeren Jiang, Si Liu, Zhaokai Wang, Leyan Zhu, Zexin He, Hongming Liu, and Shuicheng Yan. 2021. Video Background Music Generation with Controllable Music Transformer. In *Proceedings of the 29th ACM International Conference on Multimedia*. 2037–2045.
[13] Justin Domke. 2012. Generic methods for optimization-based modeling. In *Artificial Intelligence and Statistics*. 318–326.
[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).
[15] Erik Englesson and Hossein Azizpour. 2021. Generalized Jensen-Shannon Divergence Loss for Learning with Noisy Labels. *Advances in Neural Information Processing Systems* 34 (2021).
[16] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*. PMLR, 1126–1135.
[17] Xin Geng. 2016. Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering* 28, 7 (2016), 1734–1748.
[18] Shane Gilroy, Martin Glavin, Edward Jones, and Darragh Mullins. 2021. Pedestrian Occlusion Level Classification using Keypoint Detection and 2D Body Surface Area Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3833–3839.
[19] Jacob Goldberger and Ehud Ben-Reuven. 2016. Training deep neural-networks using a noise adaptation layer. (2016).
[20] Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised learning by entropy minimization. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*. 529–536.
[21] Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. 2020. A Survey of Label-noise Representation Learning: Past, Present and Future. *arXiv preprint arXiv:2011.04406* (2020).
[22] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*. 8527–8537.
[23] Jiangfan Han, Ping Luo, and Xiaogang Wang. 2019. Deep self-learning from noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*. 5138–5147. https://doi.org/DOI:10.1109/ICCV.2019.00524
[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778. https://doi.org/DOI:10.1109/cvpr.2016.90
[25] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. *arXiv preprint arXiv:1802.05300* (2018).
[26] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).
[27] Simon Jenni and Paolo Favaro. 2018. Deep bilevel learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 618–633.
[28] Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. 2020. Beyond synthetic noise: Deep learning on controlled noisy labels. ICML.
[29] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*. 2304–2313.
[30] Wentao Jiang, Si Liu, Chen Gao, Ran He, Bo Li, and Shuicheng Yan. 2020. Beautify as you like. In *Proceedings of the 28th ACM International Conference on Multimedia*. 4542–4544.
[31] Zhimeng Jiang, Kaixiong Zhou, Zirui Liu, Li Li, Rui Chen, Soo-Hyun Choi, and Xia Hu. 2021. An Information Fusion Approach to Learning with Instance-Dependent Label Noise. In *International Conference on Learning Representations*.
[32] Davood Karimi, Haoran Dou, Simon K Warfield, and Ali Gholipour. 2020. Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. *Medical Image Analysis* 65 (2020), 101759.
[33] Taehyeon Kim, Jongwoo Ko, JinHwan Choi, Se-Young Yun, et al. 2021. FINE Samples for Learning with Noisy Labels. *Advances in Neural Information Processing Systems* 34 (2021).
[34] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, et al. 2020. WILDS: A Benchmark of in-the-Wild Distribution Shifts. *arXiv preprint arXiv:2012.07421* (2020).
[35] Axel Barroso Laguna and Krystian Mikolajczyk. 2022. Key. Net: Keypoint Detection by Handcrafted and Learned CNN Filters Revisited. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).
[36] Samuli Laine and Timo Aila. 2016. Temporal Ensembling for Semi-Supervised Learning. (2016).
[37] Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, Vol. 3. 896.
[38] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. 2018. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5447–5456. https://doi.org/DOI:10.1109/CVPR.2018.00571
[39] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in Neural Information Processing Systems* 34 (2021).
[40] Junnan Li, Richard Socher, and Steven CH Hoi. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394* (2020).
[41] Jingzheng Li, Hailong Sun, and Jiyi Li. 2022. Beyond Confusion Matrix: Learning from Multiple Annotators with Awareness of Instance Features. *Machine Learning* (2022). https://doi.org/10.1007/s10994-022-06211-x
[42] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. 2019. Learning to learn from noisy labeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5051–5059.
[43] Junnan Li, Caiming Xiong, and Steven CH Hoi. 2020. Mopro: Webly supervised learning with momentum prototypes. *arXiv preprint arXiv:2009.07995* (2020).
[44] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. 2020. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966* (2020).
[45] Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. 2021. Provably end-to-end label-noise learning without anchor points. In *International Conference on Machine Learning*. PMLR, 6403–6413.
[46] Or Litany and Daniel Freedman. 2018. Soseleto: A unified approach to transfer learning and training with noisy labels. *arXiv preprint arXiv:1805.09622* (2018).
[47] Yang Liu and Hongyi Guo. 2020. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International Conference on Machine Learning*. PMLR, 6226–6236.
[48] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. 2020. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*. PMLR, 6543–6553.
[49] Andrey Malinin and Mark Gales. 2018. Predictive uncertainty estimation via prior networks. *arXiv preprint arXiv:1802.10501* (2018).
[50] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. 2019. Self: Learning to filter noisy labels with self-ensembling. *arXiv preprint arXiv:1910.01842* (2019).
[51] Amanda Olmin and Fredrik Lindsten. 2021. Robustness and reliability when training with noisy labels. *arXiv preprint arXiv:2110.03321* (2021).
[52] Sung Woo Park and Junseok Kwon. 2021. Wasserstein Distributional Normalization For Robust Distributional Certification of Noisy Labeled Data. In *International Conference on Machine Learning*. PMLR, 8381–8390.
[53] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1944–1952.

[54] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548* (2017).

[55] Hieu Pham, Qizhe Xie, Zihang Dai, and Quoc V Le. 2020. Meta pseudo labels. *arXiv preprint arXiv:2003.10580* (2020).

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

[57] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050* (2018).

[58] Maryam Sabzevari, Gonzalo Martinez-Munoz, and Alberto Suárez. 2015. Small margin ensembles can be robust to class-label noise. *Neurocomputing* 160 (2015), 18–33. https://doi.org/10.1016/j.neucom.2014.12.086

[59] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. 2017. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*. PMLR, 2988–2997.

[60] Jürgen Schmidhuber. 1995. On learning how to learn learning strategies. (1995).

[61] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*. 1919–1930.

[62] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. 2019. SELFIE: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning*. 5907–5915.

[63] Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. 2020. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199* (2020).

[64] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2014. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080* (2014).

[65] Cheng Tan, Jun Xia, Lirong Wu, and Stan Z Li. 2021. Co-learning: Learning from noisy labels with self-supervision. In *Proceedings of the 29th ACM International Conference on Multimedia*. 1405–1413.

[66] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2018. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5552–5560. https://doi.org/DOI:10.1109/CVPR.2018.00582

[67] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*. 1195–1204.

[68] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. 2017. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 839–847.

[69] Xinshao Wang, Elyor Kodirov, Yang Hua, and Neil M Robertson. 2019. Derivative manipulation for general example weighting. *arXiv preprint arXiv:1905.11233* (2019).

[70] Yulin Wang, Rui Huang, Gao Huang, Shiji Song, and Cheng Wu. 2020. Collaborative learning with corrupted labels. *Neural Networks* (2020).

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

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

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

[74] Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. 2020. Theoretical analysis of self-training with deep networks on unlabeled data. *arXiv preprint arXiv:2010.03622* (2020).

[75] Jiaheng Wei and Yang Liu. 2020. When optimizing $f$-divergence is robust with label noise. *arXiv preprint arXiv:2011.03687* (2020).

[76] Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. 2021. Learning with noisy labels revisited: A study using real-world human annotations. *arXiv preprint arXiv:2110.12088* (2021).

[77] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. *arXiv preprint arXiv:2203.05482* (2022).

[78] Dongxian Wu, Yisen Wang, Zhuobin Zheng, and Shu-Tao Xia. 2020. Temporal Calibrated Regularization for Robust Noisy Label Learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[79] Pengxiang Wu, Songzhu Zheng, Mayank Goswami, Dimitris N Metaxas, and Chao Chen. 2020. A Topological Filter for Learning with Label Noise. *Advances in neural information processing systems* 33 (2020).

[80] Yichen Wu, Jun Shu, Qi Xie, Qian Zhao, and Deyu Meng. 2020. Learning to Purify Noisy Labels via Meta Soft Label Corrector. *arXiv preprint arXiv:2008.00627* (2020).

[81] Youjiang Xu, Linchao Zhu, Lu Jiang, and Yi Yang. 2021. Faster meta update strategy for noise-robust deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 144–153.

[82] Jiangchao Yao, Jiajie Wang, Ivor W Tsang, Ya Zhang, Jun Sun, Chengqi Zhang, and Rui Zhang. 2018. Deep learning from noisy image labels with quality embedding. *IEEE Transactions on Image Processing* 28, 4 (2018), 1909–1922.

[83] Yu Yao, Tongliang Liu, Mingming Gong, Bo Han, Gang Niu, and Kun Zhang. 2021. Instance-dependent Label-noise Learning under a Structural Causal Model. *Advances in Neural Information Processing Systems* 34 (2021).

[84] Kun Yi and Jianxin Wu. 2019. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7017–7025. https://doi.org/DOI:10.1109/CVPR.2019.00718

[85] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM* 64, 3 (2021), 107–115. https://doi.org/10.1145/3446776

[86] Huaizheng Zhang, Yong Luo, Qiming Ai, Yonggang Wen, and Han Hu. 2020. Look, read and feel: Benchmarking ads understanding with multimodal multitask learning. In *Proceedings of the 28th ACM International Conference on Multimedia*. 430–438.

[87] Yivan Zhang and Masashi Sugiyama. 2021. Approximating Instance-Dependent Noise via Instance-Confidence Embedding. *arXiv preprint arXiv:2103.13569* (2021).

[88] Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. 2020. Learning with Feature-Dependent Label Noise: A Progressive Approach. In *International Conference on Learning Representations*.

[89] Zhilu Zhang and Mert Sabuncu. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in neural information processing systems*. 8778–8788.

[90] Guoqing Zheng, Ahmed Hassan Awadallah, and Susan Dumais. 2021. Meta label correction for noisy label learning. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*.

[91] Zhedong Zheng and Yi Yang. 2021. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *International Journal of Computer Vision* (2021), 1–15.

[92] Zhi-Hua Zhou and Ming Li. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering* 17, 11 (2005), 1529–1541.