How to Prevent the Continuous Damage of Noises to Model Training?

Xiaotian Yu[1], Yang Jiang[5], Tianqi Shi[5], Zunlei Feng[12], Yuexuan Wang[4], Mingli Song[12], Li Sun[3*]
[1]Zhejiang University, [2]Shanghai Institute for Advanced Study of Zhejiang University,
[3]Ningbo Innovation Center Zhejiang University, [4]University of Hong Kong, [5]Alibaba Group

Abstract

Deep learning with noisy labels is challenging and inevitable in many circumstances. Existing methods reduce the impact of mislabeled samples by reducing loss weights or screening, which highly rely on the model's superior discriminative power for identifying mislabeled samples. However, in the training stage, the trainee model is imperfect and will wrongly predict some mislabeled samples, which cause continuous damage to the model training. Consequently, there is a large performance gap between existing anti-noise models trained with noisy samples and models trained with clean samples. In this paper, we put forward a Gradient Switching Strategy (GSS) to prevent the continuous damage of mislabeled samples to the classifier. Theoretical analysis shows that the damage comes from the misleading gradient direction computed from the mislabeled samples. The trainee model will deviate from the correct optimization direction under the influence of the accumulated misleading gradient of mislabeled samples. To address this problem, the proposed GSS alleviates the damage by switching the gradient direction of each sample based on the gradient direction pool, which contains all-class gradient directions with different probabilities. During training, each gradient direction pool is updated iteratively, which assigns higher probabilities to potential principal directions for high-confidence samples. Conversely, uncertain samples are forced to explore in different directions rather than mislead model in a fixed direction. Extensive experiments show that GSS can achieve comparable performance with a model trained with clean data. Moreover, the proposed GSS is pluggable for existing frameworks. This idea of switching gradient directions provides a new perspective for future noisy-label learning.

1. Introduction

Recently, Deep Neural Networks (DNNs) have achieved breakthrough results across various computer vision tasks [9, 14–16, 20, 34, 36, 49, 50]. The high performance of DNNs requires a large amount of labeled data, but it is hard to guarantee label quality in many circumstances. As a matter of fact, many benchmark datasets inevitably contain noisy labels according to investigation results in [35].

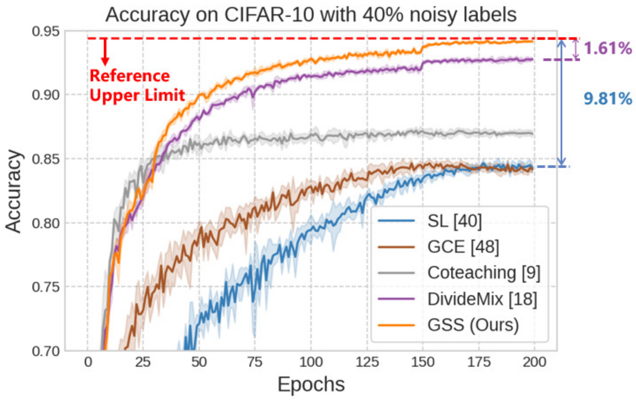


Figure 1. Performance comparison of existing methods and the proposed GSS on CIFAR-10 with 40% noisy labels. The red dashed line denotes the upper limit, which is the accuracy of models trained with completely clean labels.

Various types of researches are proposed to address the noisy-label problem. The mainstream types are robust loss function [18, 27, 44, 54] and sample screening [30, 37, 40]. These methods deal with mislabeled samples in essentially similar ways, that is, by decreasing the weights of low-confidence samples, which highly rely on the trainee model's discriminative power of identifying mislabeled samples. However, during training the trainee model is imperfect and will miss many mislabeled samples, which will continuously damage the model. That is why there is a large performance gap between existing anti-noise models trained with noisy samples and models trained with clean samples. As shown in Fig. 1, existing anti-noise methods (denoted by solid lines) have $1.61\% \sim 9.81\%$ accuracy gaps compared with the reference upper limit (red dash line), which denotes the performance of models trained with clean samples. It raises an important question: *how to prevent the continuous damage of noises to model training?* The theoretical analysis in Section 4 shows that the noise damage comes from the misleading gradient direc-

[*] Li Sun is the corresponding author.

tions caused by noises. Therefore, it is a viable solution to handle the continuous damage of noises to model training by eliminating the impact of misleading gradient directions.

In this paper, we put forward a Gradient Switching Strategy (GSS) to prevent the continuous damage of mislabeled samples to the model training. The core idea is assigning a random gradient direction to cancel out the negative impact of mislabeled samples, especially for uncertain samples which could continuously generate a misleading gradient in a single direction. For high-confidence samples, the model will be optimized using their potential principal directions with a larger probability. As the model's discriminative power grows over training time, parts of uncertain samples will become high-confidence samples, which in turn optimizes the model with their potential principal directions. Finally, the model will be well-trained with almost all samples in the dataset step by step.

Specifically, we devise a gradient direction pool for each sample, which contains all-class gradient directions with different probabilities. The probabilities of different gradient directions are determined based on the original noisy label, predictions, and partial randomness. In the training stage, for uncertain samples, the probabilities of different gradient directions are dominated by randomness. The multiple random gradient directions prevent a fixed misdirection from continuously damaging the training.

The high-confidence samples consist of two groups: the predictions are consistent with original labels (consistent sample), and the predictions are not consistent with original labels (non-consistent sample). For consistent samples, the gradient direction of the original label (potential principal direction) has a higher probability than those for the remaining gradient directions. For non-consistent samples, two highest probabilities correspond to the gradient directions of the original label and model prediction. The model explores two gradient directions and determines the potential principal direction during training. In summary, the potential principal directions of high-confidence samples guide the optimization of the model.

Experiment results demonstrate that the proposed GSS can effectively prevent the damage of mislabeled samples to the model training. The proposed GSS is pluggable for existing frameworks for noisy-label learning, which can achieve $1.23\% \sim 9.22\%$ accuracy improvement than SOTA for high noise rates. Additionally, the model with GSS trained on noisy samples can achieve comparable performance with models trained with clean samples.

Overall, our contributions are summarized as follows:

- This paper is the first to clarify the continuous damage of the mislabeled samples to model training. Theoretical analysis shows the continuous damage comes from the misleading gradient direction derived from mislabeled samples, which provides a new perspective for future noisy-label learning research.
- We propose the Gradient Switching Strategy (GSS) to prevent the continuous gradient damage of mislabeled samples to the model training. A gradient direction pool containing gradient directions of all classes with dynamic probabilities for each sample is devised to alleviate the impact of uncertain samples and optimize the model with the potential principal direction.
- Detailed theoretical analysis and extensive experimental results show that the proposed GSS can effectively prevent damage of mislabeled samples. Through combining GSS with existing anti-noise learning methods, the final classification performance can achieve up to $1.23\% \sim 9.22\%$ accuracy improvement over SOTA on datasets with severe noise, some of which are even comparable to the model trained with clean samples.

2. Related Work

Previous researches have proposed various methods for noisy-label learning, such as label correction [39, 41, 51], noisy adaptation [5, 32, 42], and meta learning [48, 52, 55]. Among existing methods, the mainstream in this field includes robust loss function and sample cleaning, which have been proven effective in various tasks with noisy labels [38].

Robust loss function based methods theoretically prove that the classifier trained with noisy data can achieve the same misclassification probability as that trained with clean data [3, 8, 31, 43]. Symmetric Cross Entropy (SCE) [44] was proposed to address the under-fitting and over-fitting problems with noisy datasets. Some researchers [1, 2] proposed replacing the Softmax layer with the exponential functions, which allows transitioning between non-convex and convex losses by different temperature parameters. Based on the discovery that DNNs are robust to noisy labels in the early learning stage, Early-learning Regularization (ELR) [25] was proposed to take past model outputs as targets for improving the robustness of models. Active Passive Loss (APL) [28] proposed a normalization function to make any loss robust to noisy labels. Compared to Cross Entropy (CE), these loss functions have an anti-noise effect on model training. However, even though these losses have been theoretically proven to be robust, there is still a big gap between the experimental performance on clean data and that on noisy data. As we analyzed in Section 4.2, the effects of most robust losses essentially reduce the gradient weight of uncertain samples. Although the model is less corrupted by the noise, these methods also reduce the weights of hard samples and result in low generalization.

Sample cleaning is another common technique devised for handling noisy labels, which is a special case of sample weighting (with binary weights) [4, 21, 29]. MentorNet [13] attempted to learn data-driven curriculums by re-weighting samples for the student model. Co-teaching [10] maintained two networks with random initialization and cross-updated the parameters. In every iteration, small-loss samples were delivered to the peer network. Some researchers [17,46,53] proposed new sample selection strategies to obtain clean datasets. Filtering the uncertain samples is a more direct approach than robust loss, but similarly, it has the side effect of reducing the sample amount. To address this problem, semi-supervised method [22,33,45,56] is applied to utilize mislabeled samples. DivideMix [22] divided the datasets and treated the low-confidence samples as unlabeled ones. The semi-supervised method was adopted to train unlabeled data based on model predictions. However, the model predictions are unreliable during the training, which can easily introduce extra noise. Moreover, since this type of noise is derived from the model predictions, it can hardly be rectified and keep corrupting the training.

3. Preliminaries

Given a training dataset $\{(x^{(n)}, \widetilde{y}^{(n)})|1 \leq n \leq N\}$, where N denotes the sample amount, $x^{(n)}$ denotes the n-th sample, and $\widetilde{y}^{(n)} \in \{1, 2, ..., K\}$ denotes the annotated labels. Assuming the true label of the n-th sample is $y^{(n)}$, the dataset is noisy if $y^{(n)} \neq \widetilde{y}^{(n)}, \exists n$. And the noise ratio of the noisy dataset is $\eta = \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}\left[y^{(n)} \neq \widetilde{y}^{(n)}\right]$. $\mathbb{1}\left[\cdot\right]$ is the indicator function. The classifier f takes samples x as the input to compute the logits z_k and the probabilities p_k of each category $k \in \{1, 2, ..., K\}$. For the l-th convolutional layer of classifier f, the kernel weights are denoted as w^l, the output feature map is denoted as m^l, and the corresponding activated feature map is denoted as $a^l = BatchNorm(\sigma(m^l))$, where $\sigma()$ and $BatchNorm()$ denote the activation function and Batch Normalization. To be noted, a^l is also the input of the $(l+1)$-th layer.

4. Theoretical Analysis of Noise Damage

The experimental results given in Fig. 1 show that the accuracy of a model trained by clean samples is $1.61\% \sim 9.81\%$ higher than that of existing methods trained by noisy labels. Although these methods reduce the influence of noise, there is still a large gap. This section discusses the question about *How Noisy Labels Affect the Training* and *Why Do Existing Methods Have Limited Effects*.

4.1. How Noisy Labels Affect the Training

For deep convolution networks, the training process updates the parameters through backward gradients, minimizing the loss function eventually. For the convolution operation $m^l = a^{l-1} \otimes w^l$ ('$\otimes$' denotes the convolution operation), the calculation of the output can be derived as:

$$m^l_{i,j} = \sum_{i'}\sum_{j'} w^l_{i',j'} a^{l-1}_{is+i',js+j'}, \tag{1}$$

where i and j are indexes of the output feature map m^l, i', and j' are indexes of the convolution kernel w^l, and s denotes the stride. Based on Eqn. (1), the gradient of loss w.r.t. the convolution kernel weights can be derived as:

$$\begin{aligned}\frac{\partial \mathcal{L}(\widetilde{y})}{\partial w^l_{i',j'}} &= \sum_{i}\sum_{j} \frac{\partial \mathcal{L}(\widetilde{y})}{\partial m^l_{i,j}} \frac{\partial m^l_{i,j}}{\partial w^l_{i',j'}} \\ &= \sum_{i}\sum_{j} dil_s\left(\frac{\partial \mathcal{L}(\widetilde{y})}{\partial m^l}\right)_{is,js} a^{l-1}_{i'+is,j'+js},\end{aligned} \tag{2}$$

where $\mathcal{L}(\widetilde{y})$ denotes the loss function with the noisy label $\widetilde{y}$, $dil_s(\cdot)$ denotes the matrix dilation, and its dilation rate equals the slide s. By comparing the subscript relationship with Eqn. (1), the Eqn. (2) can be simplified as:

$$\frac{\partial \mathcal{L}(\widetilde{y})}{\partial w^l} = a^{l-1} \otimes dil_s\left(\frac{\partial \mathcal{L}(\widetilde{y})}{\partial m^l}\right), \tag{3}$$

where the convolution stride in Eqn. (3) equals 1. Since $\frac{\partial \mathcal{L}(\widetilde{y})}{\partial m^l} = \sum_k \left(\frac{\partial \mathcal{L}(\widetilde{y})}{\partial z_k}\frac{\partial z_k}{\partial m^l}\right)$, the weight update of model layer l depends on three terms: the activated feature map of the last layer a^{l-1}, the gradient of loss w.r.t. the logits of each category $\frac{\partial \mathcal{L}}{\partial z_k}$, and the gradient of each category's logits w.r.t. the output feature map $\frac{\partial z_k}{\partial m^l}$. The second term can be regarded as the weight of the k-th category, and the third term can be regarded as the gradient direction of the k-th category. In this paper, we refer to these two terms as the **'gradient weight'** and the **'gradient direction'**. So the gradient can be regarded as the weighted sum of each category's gradient directions. For Cross-Entropy (CE) loss, the gradient weight $\frac{\partial \mathcal{L}(\widetilde{y})}{\partial z_k} = p_k - q_k$, where $q_k = \mathbb{1}\left[\widetilde{y} = k\right]$. So that the gradient weight is negative for the target category and positive for the others. For samples with wrong labels, the gradient bias can be derived based on Eqn. (3):

$$\begin{aligned}\frac{\partial \mathcal{L}(y)}{\partial w^l} - \frac{\partial \mathcal{L}(\widetilde{y})}{\partial w^l} &= a^{l-1} \otimes \Bigg(\left(\frac{\partial \mathcal{L}(y)}{\partial z_y} - \frac{\partial \mathcal{L}(\widetilde{y})}{\partial z_y}\right)\frac{\partial z_y}{\partial m^l} \\ &\quad + \left(\frac{\partial \mathcal{L}(y)}{\partial z_{\widetilde{y}}} - \frac{\partial \mathcal{L}(\widetilde{y})}{\partial z_{\widetilde{y}}}\right)\frac{\partial z_{\widetilde{y}}}{\partial m^l}\Bigg) \\ &= a^{l-1} \otimes \left(\frac{\partial z_{\widetilde{y}}}{\partial m^l} - \frac{\partial z_y}{\partial m^l}\right).\end{aligned} \tag{4}$$

It can be seen the gradient bias is related with the difference of two categories' gradient directions. ***Since the labels are fixed, this bias will accumulate during training and continuously affect the model.***

4.2. Why Do Existing Methods Have Limited Effects

Based on the above analysis, this section demonstrates the effects of existing methods for noisy-label learning. After analyzing from the perspective of gradients, we find many methods are essentially similar, including sample cleaning, reweighting, and robust loss. Sample cleaning with various strategies removes the uncertain samples, equivalent to reweighting with a binary weight. For robust loss functions, we calculate the gradients and find that they essentially reduce the gradient weight of uncertain samples.

According to the methodologies of existing methods, we derive their formulas of feature weight in Table 1, where GCE [54], SL [44], ELR [25], and Peer Loss [26] belong to robust loss functions. EG Reweighting [29] and CIW [21] belong to sample reweighting. Co-teaching [10] and DivideMix [22] belong to sample cleaning. The theoretical deductions of above methods and other recent works are given in the *supplementary materials*. Among these methods, α, β, and γ are positive hyper-parameters, A is set to a negative constant to replace $-\log 0$, and τ, τ', τ'' are cleaning thresholds. In dual branch models Co-teaching and DivideMix, p and p^* denote the prediction of the current branch and the other branch, respectively. GMM denotes the Gaussian mixture model to predict clean samples.

Based on the above analyses, it can be seen that these methods have the same characteristics. They are essentially in enhancing or inhibiting the gradient weight term $\frac{\partial \mathcal{L}}{\partial z_k}$. The gradient weight of these methods can be summarized as $\mathcal{W}(p_k - q_k)$, where $\mathcal{W}$ is positively correlated with the prediction on the annotated label p_y. The gradient weight of samples with low confidence would be reduced to avoid the influence of noise. That is how existing methods work for noisy-label learning. However, though these methods avoid the negative effect of the noise, the positive effect of hard samples on the model is also suppressed. That is why there is a big gap between these methods and the model trained by clean data. For methods that reuse uncertain samples through SSL, new noise will be introduced since unreliable predictions are applied to replace the original noisy labels. And the newly added noise is consistent with the model predictions and can be hard to rectify. In a word, for samples that are wrongly distinguished, gradient bias will accumulate. The gap between the model trained by clean and noisy data becomes larger during the training.

5. Methodology

As we discussed above, the problem that existing methods have not solved is that the misidentified samples have continuous damage to the model training, which is caused by the bias of wrong gradient direction. In this paper, we propose the Gradient Switching Strategy to address the problem. We introduce the process of the proposed GSS in Section 5.1, the combination details with various frameworks in Section 5.2, and the effect of GSS for noisy-label learning is analyzed in Section 5.3. The pseudo-code given in the *supplements* illustrates the full process of GSS.

Method	**Formula of gradient weight** $\partial \mathcal{L}(\widetilde{y}) / \partial z_k$
Cross Entropy	$p_k - q_k$
GCE [54]	$\left(-(p_y)^{\gamma-1}\right)(p_k - q_k)$
SL [44]	$(\alpha + \beta\|A\|p_y)(p_k - q_k)$
ELR [25]	$(p_k - q_k) + \frac{\sum_i p_i \hat{p}_i - \hat{p}_k}{1 - \sum_i p_i \hat{p}_i} \theta p_k$
Peer Loss [26]	$(p_k^{(n)} - q_k^{(n)}) - (p_k^{(n1)} - q_k^{(n2)})$
EG Reweighting [29]	$w_{EG}\,(p_k - q_k)$
CIW [21]	$w_{CIW}\,(p_k - q_k)$
Co-teaching [10]	$\mathbb{1}\left[\mathcal{L}(p^*)_y < \tau'\right](p_k - q_k)$
DivideMix [22]	$\mathbb{1}\left[GMM\left(\mathcal{L}(p^*)_y\right) > \tau''\right](p_k - q_k)$

Table 1. The summarized gradient weight of existing methods. Here $q_k = \mathbb{1}\left[\widetilde{y} = k\right]$.

5.1. Gradient Switching Strategy

Considering that misidentified samples can cause continuous damage in the same gradient direction, the proposed GSS is to prevent noise damage by gradient switching. Instead of switching the gradient into another fixed direction, the gradient direction pool is conducted for each sample to randomly select directions with different probabilities. The probability of selecting each category depends on the proportion in the direction pool:

$$P(\widehat{y}_e^{(n)} = k) = \mathcal{D}_k^{(n)} / \sum_{k'} \mathcal{D}_{k'}^{(n)}, \tag{5}$$

where $\mathcal{D}^{(n)} \in \mathbb{R}^K$ denotes the direction pool of the sample $x^{(n)}$, and $\widehat{y}_e$ denotes the flipped label in the e-th iteration. For each sample, the direction pool is used to randomly select the direction in each iteration. And the gradient directions are switched through flipping labels. During training, the gradient direction pool of each sample will update synchronously, and thus appropriate gradient switching strategies can be adjusted at different stages of training. In general, the GSS has two additional processes in each iteration, gradient switching and updating the direction pool.

The updating strategy of the gradient direction pool is crucial to the effect of gradient switching. For samples with high predicted confidence, the gradient will be switched to the principal direction with high probability. The uncertain samples are encouraged to be trained in various directions, rather than in a fixed direction to cause continuous damage. As the model performance improves, these uncertain samples will gradually generate their principal directions and participate in the training. Thus, three types of labels are applied for updating the direction pool, including the original labels $\mathcal{Y}^{or}$, predicted labels $\mathcal{Y}^{pr}$, and random labels $\mathcal{Y}^{rd}$. Among them, $\mathcal{Y}^{or}$ and $\mathcal{Y}^{pr}$ are one-hot vectors, and $\mathcal{Y}^{rd}$ is a K-dimensional vector with all values of $1/K$. The corresponding updating weights are denoted as v^{or}, v^{pr}, and v^{rd}, respectively. During training, the updating weights of each sample are set as follows:

$$v^{or} = p_{\widetilde{y}}(1 - e/E), \tag{6}$$

$$v^{pr} = p_{\widetilde{y}}(\lambda_1 e/E), \tag{7}$$

$$v^{rd} = \lambda_2 e/E, \tag{8}$$

where $p_{\widetilde{y}}$ denotes the predicted confidence on noisy labels, E denotes the total amount of epochs, e denotes the current epoch, and λ_1, λ_2 are parameters to determine the importance of two types of labels. The gradient direction pool of each sample is updated by the weighted sum in each epoch:

$$\mathcal{D}^{[e+1]} = \mathcal{D}^{[e]} + \mathcal{Y}^{or}v^{or} + \mathcal{Y}^{pr}v^{pr} + \mathcal{Y}^{rd}v^{rd}, \tag{9}$$

where $\mathcal{D}^{[e]}$ denotes the gradient direction pool of each sample in the e-th epoch of the training.

This group of updating weights adjusts the tendency for gradient switching on different training phases. The beginning phase tends to the original labels for fast convergence. With model predictions more accurate than annotated labels, the later phase tends to the other two types of labels. During the model training, the weights of predicted labels are increased to improve label reliability. Also, the random labels are emphasized to prevent continuous damage caused by overfitting in the fixed direction.

Thus, the gradient direction pool varies on samples with confident and uncertain predictions. Whether the annotated labels are correct or wrong, confident samples have explicit principal directions. So that mislabeled samples with accurate predictions can be trained in correct directions, rather than being removed directly. For uncertain samples, the gradients switch more randomly across all categories, which allows the model to explore in various directions without being affected by the continuous damage. As the model performance improves through training, these uncertain samples can generate their principal directions. In summary, the gradient switching strategy achieves both the utilization of all samples and the prevention of continuous damage.

5.2. Combination with Existing Frameworks

The proposed GSS is a pluggable technique, which is simple but can significantly improve the performance of existing methods. We combine our GSS with three kinds of frameworks, including the single branch model, dual branch model, and dual branch model with semi-supervised learning. For the single branch model, the strategy is given in Section 5.1 without additional settings. This section will introduce the combination strategies with the other two combinations with dual branches.

For the GSS with dual branches (GSS-DB), the difference between the two models can further prevent the gradient direction pool updating from being misled by the noise. In GSS-DB, the models of two branches are denoted as f_1, f_2. For each sample $x^{(n)}$, two gradient direction pools $\mathcal{D}_1^{(n)}$ and $\mathcal{D}_2^{(n)}$ are conducted, which are used to select gradient directions $\widehat{y}_{1,e}^{(n)}$ and $\widehat{y}_{2,e}^{(n)}$ for the training of two models, respectively. Moreover, the updating of direction pools use the predicted labels of the other model. So that even if one of the models wrongly predicts some uncertain samples, the corresponding direction pools will be updated by predictions of the other model. Compared with Co-teaching [10], GSS-DB prevents noise damage without filtering samples, making the model trained by more samples.

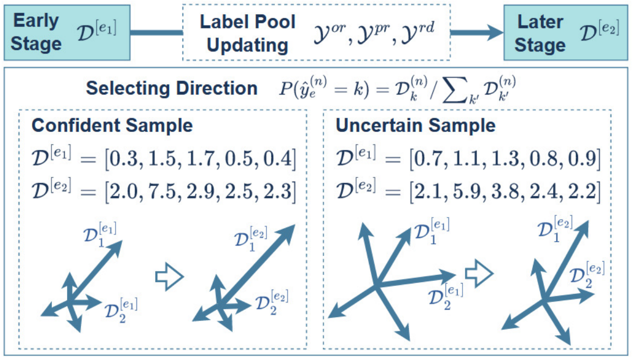

Early Stage
$\mathcal{D}^{[e_1]}$
Label Pool Updating
$\mathcal{Y}^{or}, \mathcal{Y}^{pr}, \mathcal{Y}^{rd}$
Later Stage
$\mathcal{D}^{[e_2]}$
Selecting Direction
$P(\hat{y}_e^{(n)} = k) = \mathcal{D}_k^{(n)} / \sum_{k'} \mathcal{D}_{k'}^{(n)}$
Confident Sample
$\mathcal{D}^{[e_1]} = [0.3, 1.5, 1.7, 0.5, 0.4]$
$\mathcal{D}^{[e_2]} = [2.0, 7.5, 2.9, 2.5, 2.3]$
$\mathcal{D}_1^{[e_1]}$
$\mathcal{D}_2^{[e_1]}$
$\mathcal{D}_1^{[e_2]}$
$\mathcal{D}_2^{[e_2]}$
Uncertain Sample
$\mathcal{D}^{[e_1]} = [0.7, 1.1, 1.3, 0.8, 0.9]$
$\mathcal{D}^{[e_2]} = [2.1, 5.9, 3.8, 2.4, 2.2]$
$\mathcal{D}_1^{[e_1]}$
$\mathcal{D}_2^{[e_1]}$
$\mathcal{D}_1^{[e_2]}$
$\mathcal{D}_2^{[e_2]}$

Figure 2. Illustration of gradient directions on various samples with GSS. During training, the gradient direction pool of each sample $\mathcal{D}^{(n)}$ is updated iteratively. In each epoch, new directions are selected with different probabilities determined by its direction pool. Two circumstances of confident and uncertain samples are illustrated from the early stage e_1 to the later stage e_2. The example samples have the noisy label $\widetilde{y} = 2$ and the true label $y = 1$.

From dual branches with semi-supervised learning, DivideMix [22] removes the labels of uncertain samples and trains these samples based on the predictions of the other model. Still, the method of SSL introduces new noise by using predictions as targets. In this way, the newly added noise is consistent with predictions and can hardly be corrected, causing continuous damage to the model learning. Thus, we apply GSS in this type of framework with semi-supervised learning (GSS-SSL). GSS-SSL also applies the dual branches so that the main process is consistent with GSS-DB. Additionally, two updating strategies of gradient direction pools are used for different purposes. The labeled samples are more credible than the unlabeled samples, so λ_1 is larger than λ_2 for labeled samples. On the contrary, λ_2 is larger for unlabeled samples, encouraging these samples to explore various directions to prevent noise damage. More details of GSS combinations are given in the *supplements*.

5.3. Effectiveness analysis of Gradient Switching

To demonstrate the effect of gradient switching, theoretical and experimental analyses are conducted in this section. As discussed above, Eqn. 4 denotes the gradient bias of each sample with the noisy label $\widetilde{y}$ and the clean label y. So that the total gradient bias caused by each sample within a small number of iterations $\mathcal{E}$ can be derived as:

$$\Delta g = \sum_{e}^{\mathcal{E}} \mu a^{e} \otimes \left| d_{\widetilde{y}}^{e} - d_{y}^{e} \right| , \tag{10}$$

where Δg denotes the gradient bias, d_y denotes the shorthand for the gradient direction $\partial z_y / \partial m^l$, and μ denotes the learning rate. The layer l is omitted in the interest of brevity. Assuming the activation feature map a and gradient direction d of the same sample is basically constant in a small number of iterations (experimental demonstrated in *supplementary materials*), the total bias can be simplified as:

$$\Delta g_{ori} = \mu a \otimes \left| \mathcal{E} \left(d_{\widetilde{y}} - d_y \right) \right|, \tag{11}$$

where $d_{\widetilde{y}} - d_y$ is the fixed bias that misled the model iteratively. Conversely, with unfixed directions of GSS, the total gradient bias can be derived as follows:

$$\Delta g_{gss} = \mu a \otimes \left| \sum_{e}^{\mathcal{E}} \left(d_{\widehat{y}_e} - d_y \right) \right|. \tag{12}$$

The difference between fixed and unfixed directions is that the latter replaces the accumulated bias $\mathcal{E} \left(d_{\widetilde{y}} - d_y \right)$ with the summary bias of various directions $\sum_{e}^{\mathcal{E}} \left(d_{\widehat{y}_e} - d_y \right)$. Through the updating of the gradient direction pool, the selected direction will be more reliable than the original one without continuous bias. That is the reason unfixed labels can be robust to noise.

To analyze the gradient bias in existing methods, we assume there is a method to perfectly distinguish all the noise, which means $\mathcal{W} = 0$ for mislabeled samples. So that the gradient bias of the ideal sample learning method Δg_{sc} equals the gradients with clean labels and can be derived as:

$$\begin{aligned} \Delta g_{sc} &= \mu a \otimes \left| \mathcal{E} \big(- \sum_{k} \frac{\partial \mathcal{L}(y)}{\partial z_k} d_k \big) \right| \\ &= \mu a \otimes \left| \mathcal{E} \big(\sum_{k \neq y} p_k d_k - (1 - p_y) d_y \big) \right|. \end{aligned} \tag{13}$$

And the gradient bias of SSL methods Δg_{ssl} can be derived as:

$$\Delta g_{ssl} = \mu a \otimes \left| \sum_{k} \Big(\big(\frac{\partial \mathcal{L}_{ssl}}{\partial z_k} - \frac{\partial \mathcal{L}(\widetilde{y})}{\partial z_k} \big) \frac{\partial z_k}{\partial m^l} \Big) \right|, \tag{14}$$

where $\mathcal{L}_{ssl}$ denotes the semi-supervised loss for unlabeled samples with MixMatch [6].

Experimental results are conducted in Table 2 to compare the gradient biases among Δg_{ori}, Δg_{sc}, Δg_{ssl}, and Δg_{gss}. The same term μa is omitted for simplicity of calculation. The results indicate that GSS has less bias than sample cleaning (Δg_{sc}) and SSL (Δg_{ssl}). It's worth noting that we assume that samples are perfectly distinguished in Δg_{sc}, so in fact the gradient bias of existing methods is even larger. SSL has a relatively low bias at the early stage, but the bias increases more compared to Δg_{gss} and Δg_{sc}. It might be due to the added noise by using predictions as targets for mislabeled samples.

	Dataset	CIFAR-10			CIFAR-100		
	Epochs	50	100	150	50	100	150
Gradient Bias ($\times 10^2$)	Δg_{ori}	2.15	5.65	17.62	5.26	13.26	26.91
	Δg_{sc}	1.22	2.70	6.67	3.36	7.31	14.52
	Δg_{ssl}	**1.18**	2.64	6.71	3.29	7.20	18.34
	Δg_{gss}	1.20	**2.61**	**6.53**	**3.27**	**7.04**	**12.19**

Table 2. The experimental analysis of various methods' gradient biases in different training stages. The results are calculated by adding the absolute values of gradient biases, and the average biases of mislabeled samples are shown. The experiments are conducted on CIFAR-10 and CIFAR-100 with 40% symmetric noise. The minimum biases are marked in **bold**.

6. Experiments

Dataset. We evaluate the proposed method on various datasets, including CIFAR-10, CIFAR-100 [19], Clothing1M [47], and WebVision [24]. The first two datasets are widely used benchmarks that only contain clean labels, so we generate simulated noise by symmetric and asymmetric approaches. The symmetric noise is generated by randomly flipping labels with uniform distribution, while the asymmetric noise only occurs between specific categories. Details of the asymmetric noise are provided in *supplementary materials*. The symmetric noisy labels are generated with a ratio from 40% to 80%, and the asymmetric ones are generated with a ratio from 20% to 40%. Clothing1M and WebVision are datasets with real-world noisy labels. Clothing1M contains a million images with 14 classes of clothing. The dataset is collected from online shopping websites with an overall label accuracy of 61.54%. There are several pairs of confusing classes, making this dataset very challenging. WebVision contains 2.4 million images of 1,000 same classes in ImageNet ILSVRC12 [12], which are crawled from the web with lots of noise. Based on previous works [7,23], the first 50 classes of the ImageNet subset are used for comparison.

Implementation Details. ResNet18 [11] is used as the backbone for CIFAR-10 and CIFAR-100, and ResNet50 is used as the backbone for Clothing1M and WebVision. All methods use the same backbones with pre-trained weights. The SGD with a momentum of 0.9 and weight decay of 5×10^{-4} is adopted. The batch size is set to 128 for all datasets. The initial learning rate is set to 0.01 for CIFAR-10 and CIFAR-100 and 0.001 for Clothing1M and WebVision. The learning rate is decayed in a cosine annealing manner. The hyper-parameters of existing methods are adjusted according to their papers. For the proposed GSS, the

Dataset	Method \Ratio	Symmetric				Asymmetric		
		20%	40%	60%	80%	20%	30%	40%
CIFAR-10	GCE [54]	88.77±0.18	84.66±0.30	78.43±0.25	66.11±0.27	87.28±0.13	84.63±0.15	82.15±0.27
	SL [44]	88.98±0.20	84.65±0.28	78.22±0.25	68.53±0.26	84.94±0.19	80.90±0.22	78.71±0.21
	ELR+ [25]	87.77±0.30	83.87±0.28	79.19±0.30	62.01±0.32	84.35±0.20	82.36±0.22	80.56±0.29
	Co-teaching [10]	89.59±0.09	87.20±0.20	81.40±0.15	72.94±0.21	85.99±0.12	84.23±0.11	79.48±0.12
	JoCoR [53]	86.82±0.24	85.31±0.22	76.50±0.23	66.94±0.33	86.73±0.18	79.84±0.17	77.19±0.24
	DivideMix [22]	94.26±0.14	92.85±0.19	92.26±0.21	90.07±0.17	92.98±0.15	91.57±0.13	90.59±0.16
	GSS-SSL (Ours)	**94.31±0.12**	**94.20±0.11**	**92.84±0.25**	**91.61±0.21**	**93.42±0.10**	**92.44±0.12**	**91.82±0.10**
CIFAR-100	GCE	69.19±0.24	63.17±0.35	52.45±0.32	22.60±0.40	67.19±0.30	55.41±0.28	49.75±0.28
	SL	70.43±0.29	62.28±0.31	53.20±0.45	25.79±0.42	69.11±0.28	57.63±0.30	52.06±0.27
	ELR+	66.77±0.33	63.89±0.26	49.93±0.26	19.81±0.33	64.10±0.28	51.89±0.36	46.78±0.35
	Co-teaching	70.35±0.19	64.54±0.20	52.99±0.22	27.05±0.24	69.96±0.23	58.84±0.39	55.74±0.35
	JoCoR	65.36±0.27	61.70±0.24	50.33±0.31	18.44±0.40	64.01±0.41	53.40±0.49	48.99±0.48
	DivideMix	75.89±0.14	73.90±0.16	67.41±0.16	45.82±0.15	72.20±0.20	69.04±0.19	59.16±0.19
	GSS-SSL (Ours)	**76.71±0.19**	**76.10±0.20**	**71.92±0.21**	**55.04±0.25**	**73.81±0.22**	**72.20±0.27**	**65.84±0.20**

Table 3. Classification results on CIFAR-10 and CIFAR-100 with different ratios of symmetric/asymmetric noise. The experiment compares our GSS-SSL with GCE [54], SL [44], ELR+ [25], Co-teaching [10], JoCoR [46], and DivideMix [22]. The mean accuracies over five experiments are shown, and the best results are marked in **bold** (All scores are in %).

Method	Clothing1M	WebVision		ILSVRC12	
	Top1	Top1	Top5	Top1	Top5
GCE [54]	71.73	61.22	80.81	59.13	79.09
SL [44]	72.05	63.78	84.29	61.56	84.08
ELR+ [25]	71.48	63.61	83.50	60.10	83.13
Co-teaching [10]	72.50	64.09	85.01	62.94	84.76
JoCoR [53]	71.74	60.79	82.48	57.15	81.33
DivideMix [22]	74.59	77.21	91.60	**75.23**	90.76
GSS-SSL (Ours)	**74.88**	**77.35**	**93.09**	75.18	**92.84**

Table 4. The classification accuracies (%) of various methods trained by real-world noisy datasets Clothing1M and WebVision. The mean accuracies over five experiments are shown, and the best results are marked in **bold** (All scores are in %).

weights of updating the gradient direction pool are set as $\lambda_1 = 2.0$, $\lambda_2 = 1.0$ for GSS-SB and GSS-DB. In GSS-SSL, the weights are set as $\lambda_1 = 0.2$, $\lambda_2 = 0.5$ for unlabeled samples, and $\lambda_1 = 0.1$, $\lambda_2 = 0.0$ for labeled samples.

6.1. Quantitative Evaluation

This section evaluates GSS on various benchmark datasets. The synthetic noisy labels are generated randomly on CIFAR-10 and CIFAR-100 to evaluate the effects with different noise ratios. And Clothing1M and WebVision are used to evaluate the performance with real-world noise.

Experiments on CIFAR-10/CIFAR-100: Tabel 3 shows test accuracy on CIFAR-10 and CIFAR-100 with synthetic noisy labels. The experiments are conducted with various ratios of symmetric and asymmetric noise to evaluate the performance under different conditions. Overall, the proposed GSS significantly improves over state-of-the-art methods with different noise ratios. Although DivideMix has achieved high accuracy, our method still greatly improves the performance by up to 9.22%.

Particularly, the performances of robust loss functions and sample cleaning methods decrease considerably with the increase of the noise ratio. Since a large amount of noise reduces the ability to identify clean samples, the model is more vulnerable to continuous damage. It can be seen that DivideMix is superior to other existing methods, because parts of the label noise are rectified through SSL. But there are still large gaps with the performance trained by clean labels. Our proposed GSS further reduces these gaps across all noise ratios. Especially for CIFAR-10 with symmetric noise less than 40% or asymmetric noise less than 30%, the gaps are reduced to within 0.5%. The possible explanation for the gap of DivideMix could be that the rectified labels by SSL add new noises, which are consistent with model predictions and regarded as clean samples. With GSS, the gradient directions are switched in each iteration. Thus continuous damage caused by wrong predictions can be effectively prevented.

Experiments on Clothing1M/WebVision: Table 4 shows the results of models trained on Clothing1M and WebVision. Clothing1M is a challenging dataset with a high ratio of real-world noise. Comparing to synthetic noise, some mislabeled samples in Clothing1M contain many similar features and are harder to identify. Thus the performances of existing methods are much closer. Still, GSS achieves 0.29% $\sim$ 3.40% improvement over these methods. For WebVision and ILSVRC12, the validation sets contain the same 50 classes of the ImageNet subset, and both top-1 and top-5 accuracy are shown. It can be seen the GSS improvement on top-5 accuracy is much higher than that on top-1 accuracy. This phenomenon is due to the gradient direction pool allowing the model to be trained in multiple directions for uncertain samples. Instead of minimizing predictions on other categories, GSS will be trained in various possible directions, even if the annotated labels are wrong. Consequently, for uncertain samples that cannot accurately predict, the model has a high probability of predicting cor-

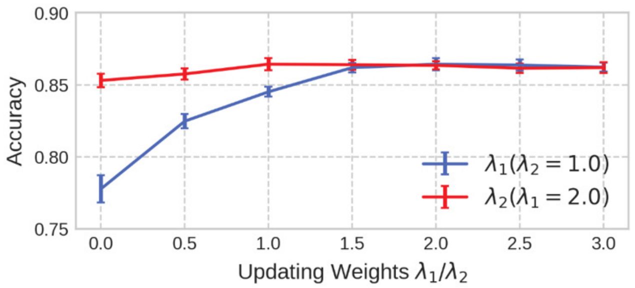
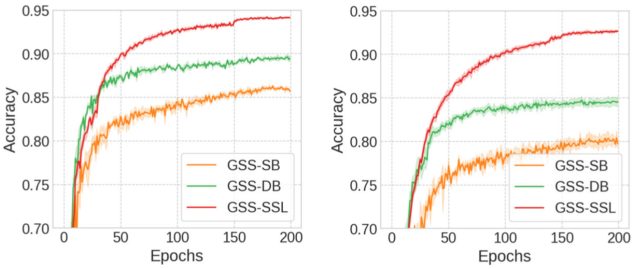


Figure 3. The ablation results with different weights of gradient direction pool updating. The experiments are conducted on CIFAR-10 with 40% noisy labels with GSS-SB. All results are the average accuracy with the error bar on the test set over five experiments.



(a) 40% Noisy labels

(b) 60% Noisy labels

Figure 4. The ablation results of GSS combinations with various frameworks. The experiments are conducted on CIFAR-10 with 40% (a) and 60% (b) noisy labels. All results are the average accuracy on the validation set over five experiments.

rectly with the top-5 predictions.

6.2. Ablation Study

Effects of Gradient Direction Pool Updating: In this section, we conduct an ablation study of different strategies for updating the gradient direction pool. Experiments are conducted with different weights (λ_1 and λ_2) to evaluate the sensitivity of these parameters. λ_1 and λ_2 are the weights of predicted labels and random labels, respectively. With weights $\lambda_1/\lambda_2 = 0$, the corresponding labels are removed to explore the effects of each component in updating the direction pool. The results are illustrated in Fig. 3.

In Fig. 3, the result with weight $\lambda_1 = 0.0$ ($\lambda_2 = 1.0$) is much lower than others, which reflects the importance of predicted labels in updating the direction pool. Without adding predicted labels, the training of each sample loses the principal directions, making the model hard to converge. With the addition of predicted labels ($\lambda_1 > 0$), the performance improves greatly. From the curve of λ_2, it can be seen the performance is less sensitive than λ_1. The result with weight $\lambda_2 = 0.0$ is also the lowest, indicating that the randomness of gradient direction is crucial to GSS. With random labels, the direction pool can select various directions for the model rather than a fixed one, which help prevent the continuous damage of label noise.

Effects of Different Combinations: Aiming at the combinations of GSS with various frameworks, experiments are conducted to provide insights into the effects of each component, shown in Fig. 4. GSS-SB, GSS-DB, and GSS-SSL denote GSS with the single branch, double branch, and semi-supervised learning, respectively. On both datasets with 40% and 60% noisy labels, GSS-DB achieves improvement compared with GSS-SB. The reason is the dual gradient direction pool further achieves damage prevention by increasing randomness in selecting gradient directions.

Moreover, GSS-SSL further improves the accuracy, and this improvement is more pronounced with 60% noisy labels. GSS-SSL solves the problem of continuous damage from incorrect predictions and enables more efficient learning of uncertain samples through semi-supervised learning. For datasets with severe noise, damage from incorrect predictions and uncertain samples is more common, so GSS-SSL can have a significant improvement for high-noise data.

7. Conclusion

This paper makes a deep analysis from a new perspective of gradient directions, demonstrating that label noise can cause continuous damage throughout the model training. Although existing methods improve the performance of noisy-label learning, the damage of misidentified noise leads to suboptimal performance. To address this problem, we put forward GSS to reduce the impact of mislabeled samples. GSS devises a dynamic gradient direction pool for each sample, which contains various gradient directions with different probabilities. With GSS, uncertain samples are forced to explore in different directions rather than mislead model optimization in a fixed direction. Our theoretical analysis demonstrates that GSS can effectively reduce the gradient biases caused by label noise. Validated by comprehensive experiments, our GSS achieves superior accuracy over all existing methods on both synthetic and real-world noisy datasets. Moreover, GSS trained with noisy labels obtains comparable performance with the model trained with clean labels. The proposed GSS effectively prevents noise damage by switching gradient directions, providing a new perspective for future noisy-label learning.

Acknowledgements. This work is funded by the National Key Research and Development Project (Grant No: 2022YFB2703100), Starry Night Science Fund of Zhejiang University Shanghai Institute for Advanced Study (Grant No. SN-ZJU-SIAS-001), Ningbo Natural Science Foundation (2022J182), Fundamental Research Funds for the Central Universities (2021FZZX001-23), Alibaba Group through Alibaba Innovative Research Program, Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, and Zhejiang Lab (No.2019KD0AD01/014).

References

[1] Ehsan Amid, Manfred KK Warmuth, Rohan Anil, and Tomer Koren. Robust bi-tempered logistic loss based on bregman divergences. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[2] Ehsan Amid, Manfred K Warmuth, and Sriram Srinivasan. Two-temperature logistic regression based on the tsallis divergence. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2388–2396. PMLR, 2019. 2

[3] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. Mcguinness. Unsupervised label noise modeling and loss correction. *International Conference on Machine Learning*, 2019. 2

[4] Noga Bar, Tomer Koren, and Raja Giryes. Multiplicative reweighting for robust neural network optimization. *arXiv preprint arXiv:2102.12192*, 2021. 2

[5] Alan Joseph Bekker and Jacob Goldberger. Training deep neural-networks based on unreliable labels. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2682–2686. IEEE, 2016. 2

[6] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019. 6

[7] Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pages 1062–1070. PMLR, 2019. 6

[8] Aritra Ghosh, Himanshu Kumar, and PS Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. 2

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1

[10] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. volume 31, 2018. 3, 4, 5, 7

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6

[12] D. Jia, D. Wei, R. Socher, L. J. Li, L. Kai, and F. F. Li. Imagenet: A large-scale hierarchical image database. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 248–255, 2009. 6

[13] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313. PMLR, 2018. 2

[14] Yongcheng Jing, Yining Mao, Yiding Yang, Yibing Zhan, Mingli Song, Xinchao Wang, and Dacheng Tao. Learning graph neural networks for image style transfer. In *ECCV*, 2022. 1

[15] Yongcheng Jing, Yiding Yang, Xinchao Wang, Mingli Song, and Dacheng Tao. Amalgamating knowledge from heterogeneous graph neural networks. In *CVPR*, 2021. 1

[16] Yongcheng Jing, Yiding Yang, Xinchao Wang, Mingli Song, and Dacheng Tao. Meta-aggregator: learning to aggregate for 1-bit graph neural networks. In *ICCV*, 2021. 1

[17] Nazmul Karim, Mamshad Nayeem Rizve, Nazanin Rahnavard, Ajmal Mian, and Mubarak Shah. Unicon: Combating label noise through uniform selection and contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9676–9686, 2022. 3

[18] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. Nlnl: Negative learning for noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 101–110, 2019. 1

[19] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009. 6

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1

[21] Abhishek Kumar and Ehsan Amid. Constrained instance and class reweighting for robust learning under label noise. *arXiv preprint arXiv:2111.05428*, 2021. 2, 4

[22] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. 2019. 3, 4, 5, 7

[23] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5051–5059, 2019. 6

[24] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017. 6

[25] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in Neural Information Processing Systems*, 33:20331–20342, 2020. 2, 4, 7

[26] Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International Conference on Machine Learning*, pages 6226–6236. PMLR, 2020. 4

[27] Yueming Lyu and Ivor W Tsang. Curriculum loss: Robust learning and generalization against label corruption. *arXiv preprint arXiv:1905.10045*, 2019. 1

[28] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*, pages 6543–6553. PMLR, 2020. 2

[29] Negin Majidi, Ehsan Amid, Hossein Talebi, and Manfred K Warmuth. Exponentiated gradient reweighting for robust training under label noise and beyond. *arXiv preprint arXiv:2104.01493*, 2021. 2, 4

[30] Eran Malach and Shai Shalev-Shwartz. Decoupling ”when to update” from ”how to update”. volume 30, 2017. 1

[31] Naresh Manwani and PS Sastry. Noise tolerance under risk minimization. *IEEE transactions on cybernetics*, 43(3):1146–1151, 2013. 2

[32] Ishan Misra, C Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2939, 2016. 2

[33] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. Self: Learning to filter noisy labels with self-ensembling. 2019. 3

[34] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015. 1

[35] Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. 2021. 1

[36] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016. 1

[37] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning*, pages 5907–5915. PMLR, 2019. 1

[38] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 2

[39] Haoliang Sun, Chenhui Guo, Qi Wei, Zhongyi Han, and Yilong Yin. Learning to rectify for robust learning with noisy labels. *Pattern Recognition*, 124:108467, 2022. 2

[40] Cheng Tan, Jun Xia, Lirong Wu, and Stan Z Li. Co-learning: Learning from noisy labels with self-supervision. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1405–1413, 2021. 1

[41] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa. Joint optimization framework for learning with noisy labels. *IEEE*, 2018. 2

[42] Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11244–11253, 2019. 2

[43] X. Wang, Y. Hua, E. Kodirov, and N. M. Robertson. Imae for noise-robust learning: Mean absolute error does not treat examples equally and gradient magnitude’s variance matters. 2019. 2

[44] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019. 1, 2, 4, 7

[45] Zhuowei Wang, Jing Jiang, Bo Han, Lei Feng, Bo An, Gang Niu, and Guodong Long. Seminll: A framework of noisy-label learning by semi-supervised learning. *arXiv preprint arXiv:2012.00925*, 2020. 3

[46] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13726–13735, 2020. 3, 7

[47] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6

[48] Youjiang Xu, Linchao Zhu, Lu Jiang, and Yi Yang. Faster meta update strategy for noise-robust deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 144–153, 2021. 2

[49] Xingyi Yang, Zhou Daquan, Songhua Liu, Jingwen Ye, and Xinchao Wang. Deep model reassembly. In *Advances in Neural Information Processing Systems*. 1

[50] Xingyi Yang, Jingwen Ye, and Xinchao Wang. Factorizing knowledge in neural networks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 73–91. Springer, 2022. 1

[51] K. Yi and J. Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[52] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7017–7025, 2019. 2

[53] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? pages 7164–7173, 2019. 3, 7

[54] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. volume 31, 2018. 1, 4, 7

[55] Guoqing Zheng, Ahmed Hassan Awadallah, and Susan Dumais. Meta label correction for noisy label learning. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 2021. 2

[56] Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Robust curriculum learning: from clean label detection to noisy label self-correction. In *International Conference on Learning Representations*, 2020. 3