



Self-Filtering: A Noise-Aware Sample Selection for Label Noise with Confidence Penalization

Qi Wei , Haoliang Sun  , Xiankai Lu , and Yilong Yin  

School of Software, Shandong University, Jinan, China
haolsun.cn@gmail.com , ylyin@sdu.edu.cn

Abstract. Sample selection is an effective strategy to mitigate the effect of label noise in robust learning. Typical strategies commonly apply the small-loss criterion to identify clean samples. However, those samples lying around the decision boundary with large losses usually entangle with noisy examples, which would be discarded with this criterion, leading to the heavy degeneration of the generalization performance. In this paper, we propose a novel selection strategy, **Self-Filtering (SFT)**, that utilizes the fluctuation of noisy examples in historical predictions to filter them, which can avoid the selection bias of the small-loss criterion for the boundary examples. Specifically, we introduce a memory bank module that stores the historical predictions of each example and dynamically updates to support the selection for the subsequent learning iteration. Besides, to reduce the accumulated error of the sample selection bias of SFT, we devise a regularization term to penalize the confident output distribution. By increasing the weight of the misclassified categories with this term, the loss function is robust to label noise in mild conditions. We conduct extensive experiments on three benchmarks with variant noise types and achieve the new state-of-the-art. Ablation studies and further analysis verify the virtue of SFT for sample selection in robust learning.

Keywords: Label noise · Sample selection · Confidence penalization

1 Introduction

Neural networks exhibit notorious vulnerability to low-quality annotation. Especially, the generalization performance would heavily degrade when label noise arises. However, most existing datasets [18, 19] are commonly collected by Web crawlers, which inevitably contains label noise. Therefore, learning with noisy labels (LNL) poses great challenges for modern deep models [28].

Sample selection [3, 14, 23, 32, 44] is an effective strategy to mitigate the effect of label noise in LNL. The main idea is to select the clean instances from the

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-20056-4_30.



Fig. 1. Illustration of *fluctuation* during model learning. Given a sample (x, y) , the fluctuation event is **correlated with predicted results and can be defined as the prediction $p_{t-1} = y$ at $t - 1$ moment while $p_{t^*} \neq y$ ($t^* > t - 1$) at subsequent moments.** The line with the arrow indicates the *fluctuation* arises.

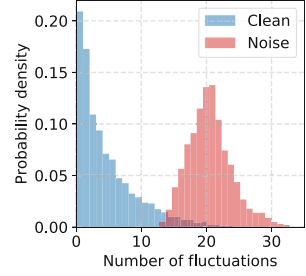


Fig. 2. Distributions of fluctuations number across synthetic CIFAR-10 with 40% symmetric label noise after 100 epochs.

corrupted dataset by using a certain criterion and reduce the bias of the training set. Inspired by the *memorization effect* that DNNs learn simple patterns shared by majority examples before fitting the noise [9], existing works [3, 9, 12, 14, 23, 32, 41, 44] commonly apply the small-loss criterion that selects samples with loss value lower than a pre-defined threshold and treats them as clean instances. Although these methods exhibit favorable properties for LNL, selecting samples with small losses would discard those boundary examples, since they are with large losses and usually entangled with noisy instances. However, those boundary examples are essential for learning a good decision boundary. Moreover, the selection bias would be accumulated and further degenerate the generalization performance as the learning proceeds [9, 32]. Besides, the loss threshold in this strategy is a crucial hyper-parameter that is usually carefully tuned by cross-validation, suffering from the issue of scalability and sensitivity.

In this paper, we have an interesting observation of the *fluctuation* in LNL, which has the potential for identifying noisy samples and retaining boundary samples. As shown in Fig. 1, a fluctuation event occurs when *a sample classified correctly at the current moment is misclassified in the following learning step*. Intuitively, as the learning proceeds, the discriminability of the classifier will be enhanced gradually, leading to a more accurate prediction for boundary examples rather than the noisy ones. Thus, the fluctuation would frequently occur for noisy samples but gradually disappear for boundary examples. On the other hand, more and more boundary samples are added to the training set via the fluctuation criterion, which can subsequently promote the learning of the classifier. To show the disparity between clean examples and noisy examples, we plot the distribution of the number of the fluctuation event in Fig. 2. The clean and noisy sample can be essentially separated by the proposed fluctuation criterion.

Based on the above observation, we propose a novel sample selection strategy, Self-Filtering (SFT), that filters out noisy samples with the fluctuation in historical predictions. We store the historical predictions of each example for different training iterations in an external memory bank and detect the fluctuation event for each sample in the current moment. By applying this criterion, SFT can filter out noisy samples and retain boundary samples for current optimization. Meanwhile, the memory bank updates dynamically as the learning proceeds, which can support selecting more reliable samples to boost the model learning. To reduce the accumulated error stemming from the selection bias of SFT, we design a regularization term to penalize the confident output distribution. By assigning the weight of the misclassified categories for the loss function, the model can avoid overconfidence of the correct prediction and further improve the robustness of our framework for severe label noise. We also integrate a semi-supervised method, FixMatch [26], into our framework to explore the useful information in the abandoned noisy samples, which has significantly advanced the model performance for noise-robust learning.

The contribution can be summarized in four aspects.

- We rethink the sample selection in LNL and propose a new strategy to select clean samples by using the fluctuation of historical predictions.
- We design a regularization term to penalise the confident output of the network, which can faithfully mitigate the sample selection bias.
- We build a novel learning framework, Self-Filtering (SFT), and achieve the new state-of-the-art on three benchmarks with variant noise types.
- We apply the proposed strategy to the prevailing learning framework and achieve a significant performance promotion, demonstrating its considerable versatility in LNL.

2 Related Work

Sample Selection. The majority of previous works exploit the memorization of DNNs and utilize the small-loss criterion to select clean samples [9, 12, 32, 41]. One representative work is MentorNet [12] that proposes a teacher network to select clean samples for a student network with the small-loss trick. Similarly, Co-teaching [9, 32, 41] constructs a double branches network to select clean samples for each branch. A surrogate loss function [22] is introduced to identify clean samples and theoretically guaranteed in [6]. To avoid tuning the threshold in the small-loss trick, Beta Mixture Model [2] or Gaussian Mixture Model [14] is introduced to separate clean and noisy examples among the training loss automatically. Zhang et al. [43] designs a meta-network trained with extra clean meta-data to identify noisy samples. Recently, to achieve the stable prediction for sample selection, Model Ensemble (or Mean Teacher) [23] is introduced to compute the exponential moving-average predictions over past iterations and replace current predictions, which performs well to confront the more complex noise type (*e.g.* instance-dependent label noise). As a homologous approach that

also aims to select the boundary samples, Me-Momentum [3] modifies the training strategy and introduces two-loop training in curriculum learning.

Robust Loss Function. A majority of robust loss functions have been theoretically analysed. Compared with the categorical cross-entropy (CCE) loss, mean absolute error (MAE) has been theoretically guaranteed to be robust to label noise [7]. Based on this analysis, a novel generalized cross-entropy loss that combined CCE and MAE has been proposed in [42] where its convergence and robustness are also analysed. Inspired by the symmetric Kullback-Leibler Divergence, a symmetric cross entropy (SCE) [31] has been designed to mitigate the effect of noisy labels. To exploit the virtue of variant noise-robust losses, a meta-learning method is designed to learn to combine the four loss functions [1, 7, 8, 31] adaptively. Recently, a new family of loss functions, the peer loss [17] are proposed to punish the agreement between outputs of the network and noisy labels by adding a regularization term to cross-entropy. It has also been proved that minimizing those robust loss functions with corrupted labels is equivalent to minimise the cross entropy loss on the clean set under mild noise ratios.

Label Correction. The pioneering methods correct the noisy labels by an additional inference phase (*e.g.* knowledge graphs [15] or graphical models [37]). Recently, two types of correction functions are proposed to correction. Firstly, transition matrix approaches [11, 36, 40] aim to construct a matrix that stores the flipping probability between the true label and the noisy one and is estimated with a set of clean data. Secondly, another family of methods utilize the output of the network to rectify labels. For example, Song et al. [27] proposes to select the clean samples by co-teaching framework [9] and progressively refurbish noisy samples with the prediction confidence. To achieve learning the correction function in a data-driven way, Wu et al. [33] builds a meta-correcter to generate labels with the input of the true label and previous predictions of meta-net.

Compared with Me-Momentum [3] that selects samples in current epoch, we construct the memory bank and propose a novel criterion based on fluctuation by leveraging the historical predictions. Also, compared with existing works that select clean samples based on historical predictions (SELFIE [27] computes the entropy value from predictions histories and RoCL [44] utilizes the variance of the training loss), our framework exhibits two advantages: (i) Less hyper-parameter. Our criterion contains only one hyper-parameter, namely, the size of historical predictions. However, the selection criteria in these works both contain another statistic threshold except the history size (*e.g.*, an entropy threshold in [27] and a loss variance threshold in [44]). (ii) Less sensitive. The setting of these thresholds is related to noise ratios, requiring more cross-validation processes. Those merits facilitate the application of our criterion to more general scenarios.

3 Methods

3.1 Problem Definition and Overall

Supposing that we have the training set $\mathcal{D} = \{(\mathbf{x}^i, y^i)\}_{i=1}^n \in (\mathcal{X}, \mathcal{Y})$ with corrupted labels and $y^i \in [\mathbf{K}] = \{1, 2, \dots, k\}$, the output distribution of a classifier f_θ

after t training epochs can be written as $\mathbf{p}^{(t)} = f(\mathbf{x}; \theta)$. Here, θ is the learnable parameters. Learning with label noise (LNL) aims to find the optimal parameter θ^* which can achieve admirable generalization performance on the clean testing set.

For a majority of sample selection methods [3, 9, 14, 23, 32, 41, 44], the robust training process for classifier f can be summarized as the following phases.

- Phase 1: select the reliable set $\tilde{\mathcal{D}}$ from the polluted dataset \mathcal{D} via a certain selection strategy. (*e.g.* small-loss criterion is designed to select the top $\tau\%$ of samples with the smaller loss values in the current mini-batch as the clean samples, where $\tau\%$ is the noise ratio estimated by cross-validation).
- Phase 2: train the classifier f on the selected set $\tilde{\mathcal{D}}$, and update the parameter as $\theta^{(t+1)} = \theta^t - \eta \nabla \left(\frac{1}{|\tilde{\mathcal{D}}|} \sum_{(\mathbf{x}, y) \in \tilde{\mathcal{D}}} \mathcal{L}(\mathbf{x}, y; \theta^t) \right)$, where η and \mathcal{L} are the given learning rate and loss function, respectively.
- Phase 3: repeat the above phases until finding the optimal parameter θ^* , then return the classifier f .

Our approach modifies the aforementioned two phases to render the network more robust on noisy labels. First, a novel criterion is proposed to select more boundary examples, which provides more decision information in the subset set $\tilde{\mathcal{D}}$. Second, we introduce a confidence regularization term to enable the loss function \mathcal{L} more robust while tackling noisy labels. Eventually, we present the learning framework, Self-Filtering (SFT), for LNL that contains two stages of warming-up and main learning. To further exploit the useful knowledge in the discarded examples, we adopt the idea of semi-supervised learning and incorporate FixMatch into our SFT.

3.2 Selecting with the Fluctuation

The key step in our selection strategy is to go through the historical prediction stored in the dynamically updated memory bank module. As shown in Fig. 3(b), we collect all predictions of the training set \mathcal{D} for the epoch t and store them in the memory bank \mathcal{M} . Specifically, this module contains those predictions of T epochs in the memory bank, where the size of \mathcal{M} is $n \times T$. \mathcal{M} maintains a queue data structure with the principle of first-in-first-out (FIFO). Therefore, for epoch t , \mathcal{M} stores predictions of the last T epochs. Finally, for the example (\mathbf{x}, y) in the current epoch t , the criterion for identifying it as the fluctuated sample can be formulated as

$$\beta = (\arg \max(\mathbf{p}^{t_1}) = y) \wedge (\arg \max(\mathbf{p}^{t_2}) \neq y), \quad (1)$$

when $t_1, t_2 \in \{t-T, \dots, t\}$, $T \geq 2$ and $t_1 < t_2$. A fluctuation event occurs when the sample classified correctly at the epoch t_1 is misclassified in the epoch t_2 . By computing β for each sample with \mathcal{M} , we discard these examples where $\beta = 1$. Therefore, the clean samples selected by fluctuation criterion can be represented as:

$$\tilde{\mathcal{D}} = \{(\mathbf{x}^i, y^i) \in \mathcal{D} | \beta^i \neq 1\}_{i=1}^n. \quad (2)$$

The selected clean samples will be utilized in the following learning stage.

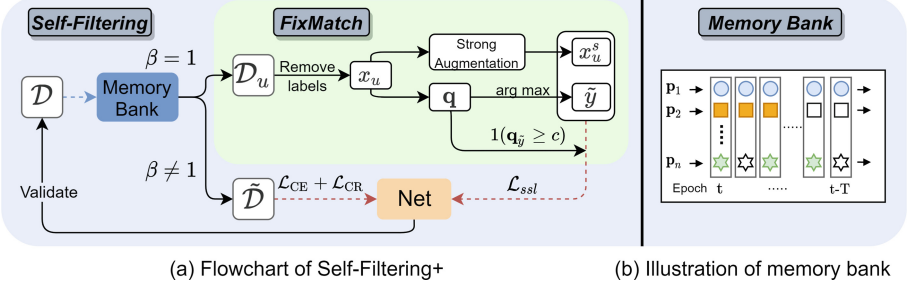


Fig. 3. (a) The blue and red dashed line denote the update of memory bank and model parameter, respectively. For selected set $\tilde{\mathcal{D}}$, the objective function is $\mathcal{L}_{CE} + \mathcal{L}_{CR}$. For the set \mathcal{D}_u , we remove their labels and input them to *FixMatch* framework. (b) MB stores the past T epoch prediction of each sample, and select the samples without fluctuation event to train the model. The updated classifier produces the prediction of the whole dataset to update MB. (Color figure online)

3.3 Learning with Selected Examples

SFT contains two stages: warming-up and main learning. We firstly conduct a warming-up of a few epochs to gain the basic discriminability for the network and achieve initialization for the memory bank module. Then, the main learning proceeds with clean samples selected via the fluctuation criterion. Also, the external memory bank module is dynamically updated for each epoch.

Warming Up. It is necessary to warm up with the whole training set for the network before the main learning stage. However, it is usually vulnerable for pair noise that has a certain pattern for noise transition, especially under the extreme ratio (*e.g.* 40%). Therefore, we penalize the confidence for the output of the network to avoid radically moving forward in warm-up stage. Let \mathbf{p}_y denotes the element in \mathbf{p} for the label y , we formulate the confident-penalization regularization term as

$$\mathcal{R} = -\alpha(\mathbf{p}_j) \cdot \log \mathbf{p}_j \quad (3)$$

where \mathbf{p}_j is the element with the second largest confidence of prediction \mathbf{p} . Under the condition of the pair noise, we consider that the class j could be the correct category for the noisy instance. $\alpha(\mathbf{p}_j)$ is an adaptive weight that can be computed by

$$\alpha(\mathbf{p}_j) = \max(0, \Gamma - \frac{\mathbf{p}_j}{\mathbf{p}_y}). \quad (4)$$

Here, Γ is a hyper-parameter of the confidence threshold. If the network is over-confident in class y , it would be penalized with a larger $\alpha(\mathbf{p}_j)$. Finally, we can write our objective function of cross-entropy loss with the regularizer in the warming-up stage as

$$\mathcal{L} = \mathbb{E}_{(\mathbf{x}, y) \in \mathcal{D}} [\mathcal{L}_{CE}(f(\mathbf{x}, \theta), y) + \mathcal{R}(f(\mathbf{x}, \theta), y)], \quad (5)$$

Main Learning. By leveraging the proposed memory bank (MB) module, we can train the classification network with the selected clean examples. Meanwhile, the MB module is updated dynamically as the learning proceeds. Specifically, the clean sample is selected via the fluctuation criterion and subsequently utilized for training the network in the current epoch. After that, we collect the predictions for the whole examples and store them into the MB module for facilitating the selection in the next epoch. Since the classification network is weak in the early learning stage, the error would be accumulated in the following iteration step [9], leading to the selection bias. To tackle this problem, we propose a regularization term \mathcal{L}_{CR} to penalize the confident output distribution for an example (\mathbf{x}, y) as following

$$\mathcal{L}_{\text{CR}} = -\frac{1}{K} \sum_{k \in [K]} \alpha(\mathbf{p}_k) \cdot \log \mathbf{p}_k, \quad (6)$$

where the coefficient $\alpha(\mathbf{p}_k)$ can be computed by Eq. (4). The regularizer penalizes the confident output of the model by minimizing the expectation of the loss for each class. This is similar with the label smoothing (LS) [20] term. Recall LS of

$$\mathcal{L}_{\text{LS}} = -\log \mathbf{p}_y - \sum_{k \in [K]} \varepsilon \log \mathbf{p}_k, \quad (7)$$

where ε is a fixed smoothing coefficient and the later term can be regarded as a confidence regularization term. Compared with LS, our coefficient $\alpha(\mathbf{p}_k)$ is adaptively computed by using the predictive value for each class, which can avoid tuning the hyper-parameter and can be more robust to the variant noise ratios.

Let $\tilde{\mathcal{D}}$ denotes the selected samples, the loss function for the main learning stage can be formulated as

$$\mathcal{L} = \mathbb{E}_{(\mathbf{x}, y) \in \tilde{\mathcal{D}}} [\mathcal{L}_{\text{CE}}(f(\mathbf{x}, \theta), y) + \lambda \mathcal{L}_{\text{CR}}(f(\mathbf{x}, \theta), y)], \quad (8)$$

where λ is a hyper-parameter set by cross-validation.

3.4 Improving with FixMatch

Our framework for sample selection is flexible, which can be combined with the state-of-the-art semi-supervised method. Hence, to further explore the knowledge in the discarded noise set, we introduce FixMatch [26] to the main learning stage. Since FixMatch is play-and-plug for SFT, we denote Self-Filtering with FixMatch as SFT+ in the following section. The SFT framework is flexible, which can be implemented by commonly-used differentiation tools. The whole learning framework is summarised in Fig. 3(a). More details can be found in the supplemental material.

4 Experiments

To evaluate the performance of our proposed method, we implement experiments in several dimensions: (1) **Task variety:** we select three visual tasks with

various dataset including CIFAR-10 [13], CIFAR-100 [13] and a real-world task Clothing1M [37]. (2) **Noise condition:** we manually corrupt the partial labels with three noise types (*e.g.* symmetric, pair and instance dependent noise) on CIFAR-10&100 and various noise ratios ranging from 20% to 80%. The code is available at <https://github.com/1998v7/Self-Filtering>.

4.1 Noise Types

To simulate the actual noise condition in real-world, we refer to [3] manually construct three noise types: symmetric, pair and instance-dependent label noise. Specially, we introduce a transition matrix T to corrupt the clean label y into a wrong label \hat{y} . Given a noise ratio τ , for each sample (\mathbf{x}, y) , the T is defined as $T_{ij}(\mathbf{x}) = P(\hat{y} = j | y = i)$, where T_{ij} denotes that the true label transits from clean label i to noisy label j .

- (1) For **symmetric** label noise, the diagonal entries of the symmetric transition matrix are $1 - \tau$ and the off-diagonal entries are $\tau / (k - 1)$, where k denotes the number of categories.
- (2) For **pair-flipped** label noise, the diagonal entries of the symmetric transition matrix are $1 - \tau$ and there exists other value τ in each row.
- (3) For **instance-dependent** label noise, we stay the same construct algorithm with [3, 34]. The actual flip rate relies on the pre-setting noise ratio τ and the representation of images. The detail algorithm is provided in Appendix 1.
- (4) For **open-set** label noise, it's reported as the combination of aforementioned type noise. We select a real-world datasets to verify the effectiveness of our framework. Clothing1M [37] contains one million images of 14 categories and its noise ratio is around 39.46%.

4.2 Network Structure and Experimental Setup

We adopt ResNet-18 [10] and ResNet-34 [10] to implement SFT on CIFAR-10 and CIFAR-100, respectively. The setting for the optimizer is listed as follows that SGD is with the momentum 0.9, the weight decay is $5e-4$, the batch size is 32, and the initial learning rate is 0.02, and decayed with the factor 10 at 60 epoch. The number of epoch is set to be 75 for CIFAR-10 and 100 for CIFAR-100. For the warming-up stage, we train the network for 10 epochs and 30 epochs for CIFAR-10 and CIFAR-100, respectively, which is similar to Me-Momentum [3]. Typical data augmentations including randomly cropping and horizontally flipping are applied in our experiments.

For Clothing1M, we utilize the same architecture of ResNet-50 pre-trained on ImageNet. For image preprocessing and data augmentations, we resize the image to 256×256 and crop them into 224×224 . The horizontally flipping is adopted. We train the classifier network for 15 epochs using SGD with 0.9 momentum, weight decay of 0.0005, and the batch size of 32. The warming-up stage is one epoch. The learning rate is set as 0.02 and decayed with the factor of 10 after 10

Table 1. Test accuracy (%) on CIFAR-10 (the *top*) and CIFAR-100 (the *bottom*). The mean accuracy (\pm std) over 5 repetitions are reported.

Method	Symm.		Pair.		Inst.	
	20%	40%	20%	40%	20%	40%
DMI [39]	88.18 \pm 0.36	83.98 \pm 0.48	89.44 \pm 0.41	84.37 \pm 0.78	89.14 \pm 0.36	84.78 \pm 1.97
Peer Loss [17]	88.97 \pm 0.47	84.29 \pm 0.52	89.61 \pm 0.66	85.18 \pm 0.87	89.94 \pm 0.51	85.77 \pm 1.19
Co-teaching [9]	87.16 \pm 0.11	83.59 \pm 0.28	86.91 \pm 0.37	82.77 \pm 0.57	86.54 \pm 0.11	80.98 \pm 0.39
JoCoR [32]	88.69 \pm 0.19	85.44 \pm 0.29	87.75 \pm 0.46	83.91 \pm 0.49	87.31 \pm 0.27	82.49 \pm 0.57
SELFIE [27]	90.18 \pm 0.25	86.27 \pm 0.31	89.29 \pm 0.19	85.71 \pm 0.30	89.24 \pm 0.27	84.16 \pm 0.44
CDR [35]	89.68 \pm 0.38	86.13 \pm 0.44	89.19 \pm 0.29	85.79 \pm 0.41	90.24 \pm 0.39	83.07 \pm 1.33
Me-Momentum [3]	91.44 \pm 0.33	88.39 \pm 0.34	90.91 \pm 0.45	87.49 \pm 0.56	90.86 \pm 0.21	86.66 \pm 0.91
PES [4]	92.38 \pm 0.41	87.45 \pm 0.34	91.22 \pm 0.42	89.52 \pm 0.91	92.69 \pm 0.42	89.73 \pm 0.51
SFT (ours)	92.57 \pm 0.32	89.54 \pm 0.27	91.53 \pm 0.26	89.93 \pm 0.47	91.41 \pm 0.32	89.97 \pm 0.49
DMI [39]	58.73 \pm 0.70	49.81 \pm 1.22	59.41 \pm 0.69	48.13 \pm 0.52	58.05 \pm 0.20	47.36 \pm 0.68
Peer Loss [17]	58.41 \pm 0.55	50.53 \pm 1.31	58.73 \pm 0.51	50.17 \pm 0.42	58.91 \pm 0.41	48.61 \pm 0.78
Co-teaching [9]	59.28 \pm 0.47	51.60 \pm 0.49	58.07 \pm 0.61	49.79 \pm 0.69	57.24 \pm 0.69	49.39 \pm 0.99
JoCoR [32]	64.17 \pm 0.19	55.97 \pm 0.46	60.42 \pm 0.35	50.97 \pm 0.58	61.98 \pm 0.39	50.59 \pm 0.71
SELFIE [27]	67.19 \pm 0.30	61.29 \pm 0.39	65.18 \pm 0.23	58.67 \pm 0.51	65.44 \pm 0.43	53.91 \pm 0.66
CDR [35]	66.52 \pm 0.24	60.18 \pm 0.22	66.12 \pm 0.31	59.49 \pm 0.47	67.06 \pm 0.50	56.86 \pm 0.62
Me-Momentum [3]	68.03 \pm 0.53	63.48 \pm 0.72	68.42 \pm 0.19	59.73 \pm 0.47	68.11 \pm 0.57	58.38 \pm 1.28
PES [4]	68.89 \pm 0.41	64.90 \pm 0.57	69.31 \pm 0.25	59.08 \pm 0.81	70.49 \pm 0.72	65.68 \pm 0.44
SFT (ours)	71.98 \pm 0.26	69.72 \pm 0.31	71.23 \pm 0.29	69.29 \pm 0.42	71.83 \pm 0.42	69.91 \pm 0.54

Table 2. Comparison results with SSL with symmetric (S), pair (P) and instance (I) label noise.

Methods	SSL	CIFAR-10			CIFAR-100		
		S 50%	P 40%	I 40%	S 50%	P 40%	I 40%
SELF [23]	Mean teacher	91.4	90.9	90.4	71.8	70.7	69.1
CORES ^{2*} [5]	UDA	93.1	92.4	92.2	73.1	72.0	71.9
DivideMix [14]	MixMatch	94.6	<u>93.4</u>	93.0	74.6	72.1	71.7
ELR+ [16]	MixMatch	93.8	92.7	92.2	72.4	<u>74.4</u>	72.6
SFT+	FixMatch	<u>94.8</u>	92.9	94.4	75.4	74.2	<u>74.1</u>
SFT+*	MixMatch	94.9	93.7	<u>94.1</u>	<u>75.2</u>	74.9	74.6

epochs. Following the convention from [14], we sample 1000 mini-batches from the training data while ensuring the labels (noisy) are balanced.

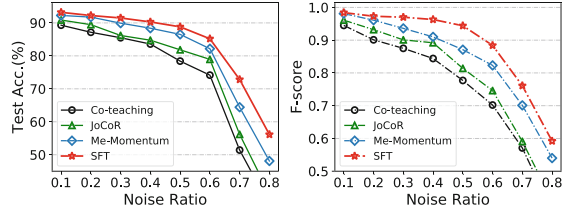
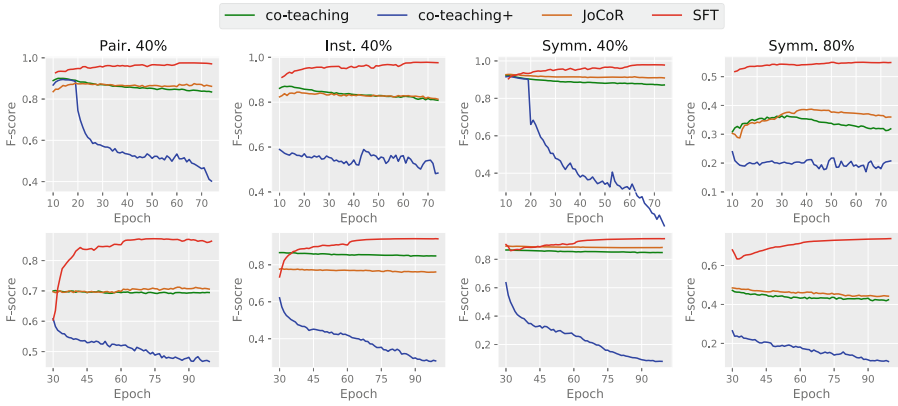
Hyper-parameter Setup. We set memory bank size $T = 3$ and confidence threshold $\Gamma = 0.2$ for all experiments. We set the trade-off coefficient λ in loss function as 1 and the threshold c in FixMatch as 0.95 following [26].

4.3 Comparison with State-of-the-Arts

Baseline. We evaluate our method against the following state-of-the-art methods. (1) Robust loss function: DMI [39], Peer loss [17]. (2) Sample selection

Table 3. Test accuracy (%) on the Clothing1M.

Method		Acc.
No SSL	Cross entropy	64.54
	MentorNet [12]	67.14
	Co-teaching [9]	68.51
	JoCoR [32]	70.30
	Forward [24]	69.84
	Joint Optim [29]	72.23
	Me-Momentum [3]	73.13
	SFT	74.16
SSL	DivideMix [14]	74.76
	ELR+ [16]	74.81
	SFT+*	75.08

**Fig. 4.** Illustration of the performance of variant methods as the noise ratio changes. Notably, SFT can produce considerable performance under extreme noise ratios.**Fig. 5.** SFT achieves the highest F-score for sample selection results on CIFAR-10 (the *top*) and CIFAR-100 (the *bottom*).

methods: Co-teaching [9], JoCoR [32], SELFIE [27], Me-Momentum [3]. Note that both of them explore the memorization effect and utilize the small-loss criterion to select clean examples. (3) Sample selection methods with SSL: SELF [23], CORES^{2*} [5], DivideMix [14] and ELR+ [16]. Specially, SELF and CORES^{2*} use the SSL methods of Mean Teacher [30] and UDA [38], respectively. DivideMix and ELR+ both utilize MixMatch. (4) Others: CDR [35], PES [4].

Results on CIFAR-10 & 100. To evaluate the performance of SFT, we conduct experiments on CIFAR-10 and CIFAR-100 under three noise types with variant noise ratios $\tau \in \{0.2, 0.4\}$. Since SSL can dramatically improve the performance, we split the table of each benchmark into two parts for a fair comparison. The bottom part includes methods using SSL, while the top part does not. As shown in Table 1, our SFT outperforms the almost state-of-the-art on two datasets with three noise types. Compared with the homologous approach Me-Momentum, SFT achieves the higher accuracy of 89.97% and gains the sig-

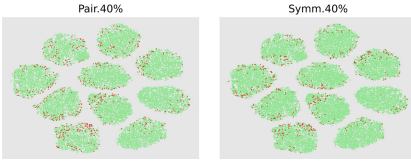


Fig. 6. Most boundary examples can be selected by SFT. The clusters with green dots are samples in the same categories. The red dots are the selected clean samples in the last ten epochs. (Color figure online)

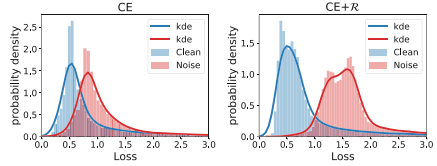


Fig. 7. The losses distribution of noisy CIFAR-10 with Pair.40% after warming-up. There exists an obvious disparity between clean and noisy samples by using the regularizer \mathcal{R} during warming-up.

nificant improvement of **6.24%** in *Symm.-40%*, **9.56%** in *Pair.-40%* and **11.53%** in *Inst.-40%*. The results exhibit the superiority of SFT in handling the noise issue for learning with more categories.

Our framework is flexible that can easily integrate the semi-supervised technology (SSL) to further boost the generalization performance, denoting SFT+ (with FixMatch) or SFT+* (with MixMatch). Table 2 shows the result when combining vanilla models with recent SSL techniques. The hybrid approach, SFT+ (or SFT+*) consistently outperforms other methods. Especially, for the instance-dependent label noise, our approach achieves the average of **1.7%** improvement compared with the state-of-the-art.

Results on Clothing1M. To demonstrate the effectiveness of our method on real-world noisy labels, we evaluate our approach on Clothing1M dataset, which is a real-world benchmark in LNL tasks. As shown in Table 3, our proposal obtains the state-of-the-art performance. For a fair comparison, we divide the table into two parts according to using the semi-supervised technique or not. Our SFT+* (with MixMatch) outperforms other methods and achieves the improvement of **0.27%** over ELR+, demonstrating its effectiveness for the real-world application.

4.4 Further Analysis

Robustness. To validate the robustness of SFT in a more challenging noisy environment, we compare it with three robust methods [3, 9, 32] under a more noise ratio setting $\tau \in \{0.1, \dots, 0.8\}$. We plot the test accuracy and F-score as the *Symm.* noise ratio increases. As shown in Fig. 4, SFT produces a good performance, even on the challenging condition with a far higher noise ratio. Compared with the homologous approach Me-Momentum, SFT exhibits the favorable property of selecting the clean samples and consistently outperforms Me-Momentum on variant noise ratios. SFT also exhibits considerable robustness under extreme noise conditions as shown in Fig. 4.

Effectiveness. We evaluate the effectiveness of SFT from these three aspects.

Table 4. The significant gain with our selection criterion for other methods.

Method		CIFAR-10				CIFAR-100			
		P 40%	I 40%	S 40%	S 80%	P 40%	I 40%	S 40%	S 80%
Co-teaching	Base	82.7	80.8	83.5	34.6	49.8	49.3	51.6	19.2
	After	84.9 (2.2)	83.9 (3.1)	85.2 (1.7)	39.7 (5.1)	52.1 (2.3)	51.9 (2.6)	54.1 (2.5)	23.6 (4.4)
JoCoR	Base	83.9	82.4	85.4	38.0	51.0	50.6	56.0	22.7
	After	86.8 (2.9)	85.6 (3.2)	87.1 (1.7)	41.6 (3.6)	53.6 (2.6)	53.6 (3.0)	58.4 (2.4)	26.8 (4.1)
DivideMix	Base	93.4	94.5	94.7	93.0	71.0	70.9	72.9	57.9
	After	94.1 (0.7)	94.7 (0.2)	94.6 (0.1)	92.8 (0.2)	72.4 (1.4)	72.0 (1.1)	73.3 (0.4)	59.1 (1.2)

Table 5. Comparison of the total training time (hours) on CIFAR-10.

Co-teaching [9]	MW-Net [25]	MSLC [33]	CORES ² [5]	DivideMix [14]	Me-Momentum [3]	SFT	SFT+
2.9 h	4.7 h	4.1 h	2.6 h	4.4 h	4.8 h	2.4 h	4.1 h

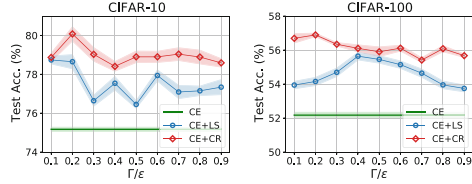
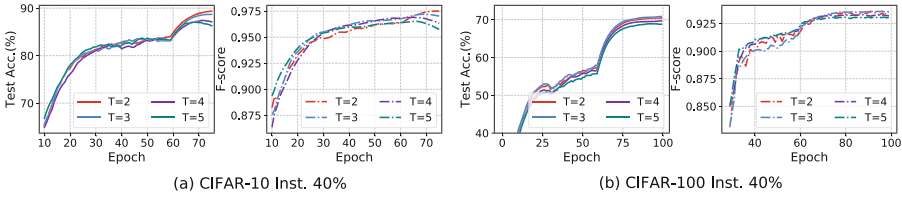
- (1) *How accurate is the sample selection strategy?* We conduct the comparison experiments on CIFAR-10 & 100 and plot the curves of F-score for the selection result. As shown in Fig. 5, SFT achieves the highest F-score on all noise types as the training proceeds. Under 40% noise ratios, SFT attains average 0.97 F1-score, indicating it obtains high selection accuracy and recall scores.
- (2) *Are the boundary examples selected?* We conduct the experiment with *Inst.* & *Pair*-40% and record the selected set in different epoch to illustrate the dynamic selecting process by t-SNE [21]. As illustrated in Fig. 6, most of selected samples (red) lie around the decision boundary, demonstrating the effectiveness of the fluctuation criterion for selecting boundary examples.
- (3) *How effective is the framework in the warming-up stage?* We plot the distribution of training losses for all instances after warming-up in Fig. 7. The blue and red parts represent the losses of clean and noise labels, respectively. By introducing the regularizer during the warming-up stage, there exists an obvious disparity between clean and noisy samples, verifying the effect of \mathcal{R} in mitigating the overconfidence during warming-up.

Versatility. Our fluctuation criterion is flexible and play-and-plug that can be applied on other modern methods [9, 14, 32]. We replace the sample selection phase with our selection module and conduct experiments on CIFAR-10 & 100 under four settings of noise. As shown in Table 4, by introducing the fluctuation criterion, these three methods almost outperform the basic version that uses the small-loss criterion. Even in the current SOTA work DivideMix, the improvement of performance on CIFAR-100 is remarkable. The fluctuation criterion gains a significant average improvement of almost **2.0%** under all settings. These results demonstrate the great flexibility of our proposed selection strategy.

Efficiency. We compare the training time with typical methods to show its efficiency. We evaluate them on CIFAR-10 and obtain the mean value of train-

Table 6. The hyper-parameter selection of Γ . Test acc. is reported.

CIFAR-10	0.2	0.4	0.6	0.8
Symm. 40%	89.67	89.51	89.27	89.29
Pair. 40%	89.74	90.11	89.51	89.43
Inst. 40%	89.93	89.61	89.37	89.29
CIFAR-100	0.2	0.4	0.6	0.8
Symm. 40%	69.88	69.61	69.17	69.34
Pair. 40%	69.23	69.07	68.71	68.66
Inst. 40%	69.84	69.91	69.52	69.37

**Fig. 8.** Comparison confidence regularization (CR) with label smooth (LS). Γ , ε denote the confidence threshold and smooth coefficient respectively.**Fig. 9.** The hyper-parameter selection of memory bank size T . Test accuracy (%) and F-score are reported.

ing time with the 40% rate of three noise types. All models are trained on a single Geforce-3090. As shown in Table 5, SFT is consistently faster than other methods since it can directly back-propagate with selected and does not rely on sophisticated learning strategies, *e.g.*, two-loop training in Me-Momentum [3].

4.5 Hyper-parameter Selection

The confidence threshold Γ and the size T of the memory bank are two hyper-parameters that need tuning with cross-validation. To study the impact of T and Γ , we conduct ablation studies and compared the confidence regularization (CR) with label smoothing (LS) as shown in Table 6, Fig. 8 and Fig. 9.

Confidence threshold Γ . We conduct the experiments from two aspects. First, in Table 6, we compare different setting of Γ on our framework and the value of Γ belongs to $\{0.2, 0.4, 0.6, 0.8\}$. As shown, the different values of Γ slightly affect the generalization performance of the model and a relatively small Γ is more preferred in our learning framework.

Further, to verify that this confidence regularization can be regarded as a robust loss function in a mild noisy condition, we compare CR with LS on two benchmarks with 40% symm. label noise. As shown in Fig. 8, our model achieves the best performance when $\Gamma = 0.2$ for all benchmarks. For the smooth coefficient ε , the setting is variant for different benchmarks (*e.g.* 0.1 for CIFAR-10, 0.4 for CIFAR-100). Therefore, we recommend $\Gamma = 0.2$ for most cases.

Memory Bank Size T . We conduct experiments with variant settings and plot the testing accuracy and F-score for sample selection in Fig. 9. Intuitively,

Table 7. Ablation study of each components. The results of *test accuracy* and *F-score* on variant noise labels are reported.

CIFAR-10	Pair. 40%		Inst. 40%	
	Acc.	F-score	Acc.	F-score
SFT	89.74	0.963	90.06	0.969
w. Voting	85.11	0.862	83.92	0.846
w/o. \mathcal{R}	88.79	0.952	88.83	0.961
w/o. \mathcal{L}_{CR}	87.14	0.944	87.36	0.948
w/o. \mathcal{R} & \mathcal{L}_{CR}	86.05	0.940	86.31	0.946

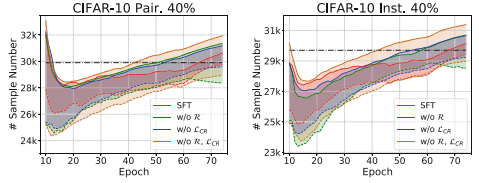


Fig. 10. Ablation study of selected samples number. The solid lines and dashed lines denote the number of selected set $\tilde{\mathcal{D}}$ and the clean labels in $\tilde{\mathcal{D}}$, respectively. The horizontal dashed line denotes the actual clean samples number.

the larger size T tends to detect more fluctuation events in the memory bank and further mitigate the selection of boundary examples. As we expected, the classification results and selection accuracy illustrate that our model attains the best performance with a smaller T . Therefore, we set $T = 3$ for all experiments.

4.6 Ablation Study

Selection Criterion. The majority voting strategy can select samples with high probability of consistent prediction results (right or wrong) in memory bank. Thus, we conduct an ablation study that replaces our criterion with the voting strategy. In Table 7, the significant improvement of test accuracy and F-score compared with voting strategy verifies the superiority of the fluctuation criterion.

Regularization Terms. To directly validate the effectiveness of regularization terms for warming-up and main learning, we remove them in different stages and retrain the model. We evaluate them with the classification accuracy and F-score for the selected results. As shown in Table 7, by removing each component, the performance of the model is degraded. Specifically, test accuracy and F-score averagely decrease **3.72%** and **0.023** on 40% noise ratio without \mathcal{R} and \mathcal{L}_{CR} . We also plot the selection curve in Fig. 10. With the support of the two terms, the selected subset contains less noisy labels. Meanwhile, red dashed lines in two figures are close to the horizontal dashed line, indicating the boundary samples are almost selected by our framework.

5 Conclusion

In this paper, we propose a simple but effective framework, Self-Filtering, to select clean samples from the noisy training set. We build a memory bank module to store the historical predictions and design the fluctuation criterion for selection. Compared with the small-loss criterion, the fluctuation strategy takes

the boundary sample into account and improves the generalization performance for learning with noisy labels. To reduce the accumulated error of the sample selection bias, we propose a confidence-penalization term. By increasing the weight of the misclassified categories with this term, we mitigate the noise effect in learning proceeding and thus the algorithm is robust to label noise. Extensive experiments and studies exhibit the great properties of the proposed framework.

Acknowledgment. This work was supported in part by National Natural Science Foundation of China (No. 62106129, 62106128, 62176139), Natural Science Foundation of Shandong Province (No. ZR2021ZD15, ZR2021QF053, ZR2021QF001), Young Elite Scientists Sponsorship Program by CAST (No. 2021QNRC001), and China Postdoctoral Science Foundation (No. 2021TQ0195, 2021M701984).

References

1. Amid, E., Warmuth, M.K., Anil, R., Koren, T.: Robust bi-tempered logistic loss based on Bregman divergences. In: *NeurIPS* (2019)
2. Arazo, E., Ortego, D., Albert, P., O'Connor, N., McGuinness, K.: Unsupervised label noise modeling and loss correction. In: *ICML* (2019)
3. Bai, Y., Liu, T.: ME-MOMENTUM: extracting hard confident examples from noisily labeled data. In: *ICCV* (2021)
4. Bai, Y., et al.: Understanding and improving early stopping for learning with noisy labels. In: *NeurIPS* (2021)
5. Cheng, H., Zhu, Z., Li, X., Gong, Y., Sun, X., Liu, Y.: Learning with instance-dependent label noise: a sample sieve approach. In: *ICLR* (2021)
6. Cheng, J., Liu, T., Ramamohanarao, K., Tao, D.: Learning with bounded instance and label-dependent label noise. In: *ICML* (2020)
7. Ghosh, A., Kumar, H., Sastry, P.: Robust loss functions under label noise for deep neural networks. In: *AAAI* (2017)
8. Gong, M., Li, H., Meng, D., Miao, Q., Liu, J.: Decomposition-based evolutionary multiobjective optimization to self-paced learning. *IEEE Trans. Evol. Comput.* **23**, 288–302 (2018)
9. Han, B., et al.: Co-teaching: robust training of deep neural networks with extremely noisy labels. In: *NeurIPS* (2018)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *CVPR* (2016)
11. Hendrycks, D., Mazeika, M., Wilson, D., Gimpel, K.: Using trusted data to train deep networks on labels corrupted by severe noise. In: *NeurIPS* (2018)
12. Jiang, L., Zhou, Z., Leung, T., Li, L.J., Fei-Fei, L.: MentorNet: learning data-driven curriculum for very deep neural networks on corrupted labels. In: *ICML* (2018)
13. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
14. Li, J., Socher, R., Hoi, S.C.: DivideMix: learning with noisy labels as semi-supervised learning. In: *ICLR* (2020)
15. Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., Li, L.J.: Learning from noisy labels with distillation. In: *ICCV* (2017)
16. Liu, S., Niles-Weed, J., Razavian, N., Fernandez-Granda, C.: Early-learning regularization prevents memorization of noisy labels. In: *NeurIPS* (2020)

17. Liu, Y., Guo, H.: Peer loss functions: learning from noisy labels without knowing noise rates. In: ICML (2020)
18. Lu, X., Ma, C., Shen, J., Yang, X., Reid, I., Yang, M.H.: Deep object tracking with shrinkage loss. *IEEE Trans. Pattern Anal. Mach. Intell.* (2020)
19. Lu, X., Wang, W., Shen, J., Crandall, D., Van Gool, L.: Segmenting objects from relational visual data. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 7885–7897 (2021)
20. Lukasik, M., Bhojanapalli, S., Menon, A., Kumar, S.: Does label smoothing mitigate label noise? In: ICML (2020)
21. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**(11) (2008)
22. Natarajan, N., Dhillon, I.S., Ravikumar, P.K., Tewari, A.: Learning with noisy labels. In: *NeurIPS* (2013)
23. Nguyen, D.T., Mummadi, C.K., Ngo, T.P.N., Nguyen, T.H.P., Beggel, L., Brox, T.: SELF: learning to filter noisy labels with self-ensembling. In: *ICLR* (2020)
24. Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., Qu, L.: Making deep neural networks robust to label noise: a loss correction approach. In: *CVPR* (2017)
25. Shu, J., et al.: Meta-weight-net: learning an explicit mapping for sample weighting. In: *NeurIPS* (2019)
26. Sohn, K., et al.: FixMatch: Simplifying semi-supervised learning with consistency and confidence. In: *NeurIPS* (2020)
27. Song, H., Kim, M., Lee, J.G.: SELFIE: refurbishing unclean samples for robust deep learning. In: *ICML* (2019)
28. Sun, H., Guo, C., Wei, Q., Han, Z., Yin, Y.: Learning to rectify for robust learning with noisy labels. *Pattern Recogn.* **124**, 108467 (2022)
29. Tanaka, D., Ikami, D., Yamasaki, T., Aizawa, K.: Joint optimization framework for learning with noisy labels. In: *CVPR* (2018)
30. Tarvainen, A., Valpola, H.: Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results. In: *NeurIPS* (2017)
31. Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., Bailey, J.: Symmetric cross entropy for robust learning with noisy labels. In: *ICCV* (2019)
32. Wei, H., Feng, L., Chen, X., An, B.: Combating noisy labels by agreement: a joint training method with co-regularization. In: *CVPR* (2020)
33. Wu, Y., Shu, J., Xie, Q., Zhao, Q., Meng, D.: Learning to purify noisy labels via meta soft label corrector. In: *AAAI* (2021)
34. Xia, X., et al.: Robust early-learning: hindering the memorization of noisy labels. In: *ICLR* (2020)
35. Xia, X., Liu, T., Han, B., Gong, C., Wang, N., Ge, Z., Chang, Y.: Robust early-learning: hindering the memorization of noisy labels. In: *ICLR* (2021)
36. Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., Sugiyama, M.: Are anchor points really indispensable in label-noise learning? In: *NeurIPS* (2019)
37. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: *CVPR* (2015)
38. Xie, Q., Dai, Z., Hovy, E.H., Luong, T., Le, Q.: Unsupervised data augmentation for consistency training. In: *NeurIPS* (2020)
39. Xu, Y., Cao, P., Kong, Y., Wang, Y.: L_DMI: an information-theoretic noise-robust loss function. In: *NeurIPS* (2019)
40. Yao, Y., et al.: Dual T: reducing estimation error for transition matrix in label-noise learning. In: *NeurIPS* (2020)

41. Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., Sugiyama, M.: How does disagreement help generalization against label corruption? In: ICML (2019)
42. Zhang, Z., Sabuncu, M.R.: Generalized cross entropy loss for training deep neural networks with noisy labels. In: NeurIPS (2018)
43. Zhang, Z., Zhang, H., Arik, S.O., Lee, H., Pfister, T.: Distilling effective supervision from severe label noise. In: CVPR (2020)
44. Zhou, T., Wang, S., Bilmes, J.: Robust curriculum learning: from clean label detection to noisy label self-correction. In: ICLR (2020)