How does Early Stopping Help Generalization against Label Noise?

Hwanjun Song [1] **Minseok Kim** [1] **Dongmin Park** [1] **Jae-Gil Lee** [1]

Abstract

Noisy labels are very common in real-world training data, which lead to poor generalization on test data because of overfitting to the noisy labels. In this paper, we claim that such overfitting can be avoided by "early stopping" training a deep neural network before the noisy labels are severely memorized. Then, we resume training the early stopped network using a "maximal safe set," which maintains a collection of almost certainly true-labeled samples at each epoch since the early stop point. Putting them all together, our novel two-phase training method, called ***Prestopping***, realizes *noise-free* training under *any type* of label noise for practical use. Extensive experiments using four image benchmark data sets verify that our method significantly outperforms four state-of-the-art methods in test error by 0.4–8.2 percent points under the existence of real-world noise.

1. Introduction

By virtue of massive labeled data, deep neural networks (DNNs) have achieved a remarkable success in numerous machine learning tasks (Krizhevsky et al., 2012; Redmon et al., 2016). However, owing to their high capacity to memorize any label noise, the generalization performance of DNNs drastically falls down when noisy labels are contained in the training data (Jiang et al., 2018; Han et al., 2018; Song et al., 2019). In particular, Zhang et al. (2017) have shown that a standard convolutional neural network (CNN) can easily fit the entire training data with any ratio of noisy labels and eventually leads to very poor generalization on the test data. Thus, it is challenging to train a DNN robustly even when noisy labels exist in the training data.

A popular approach to dealing with noisy labels is "sample selection" that selects true-labeled samples from the noisy training data (Ren et al., 2018; Han et al., 2018; Yu et al., 2019). Here, $(1\text{–}\tau)\times 100\%$ of *small-loss* training samples are treated as true-labeled ones and then used to update a DNN robustly, where $\tau \in [0, 1]$ is a noise rate. This *loss-based separation* is well known to be justified by the *memorization effect* (Arpit et al., 2017) that DNNs tend to learn easy patterns first and then gradually memorize all samples.

Despite its great success, a recent study (Song et al., 2019) has argued that the performance of the loss-based separation becomes considerably worse depending on the type of label noise. For instance, the loss-based approach well separates true-labeled samples from false-labeled ones in *symmetric noise* (Figure 1(a)), but many false-labeled samples are misclassified as true-labeled ones because the two distributions overlap closely in *pair* and *real-world noises* (Figures 1(b) and 1(c)), both of which are more *realistic* than symmetric noise (Ren et al., 2018; Yu et al., 2019). This limitation definitely calls for a new approach that supports *any type* of label noise for practical use.

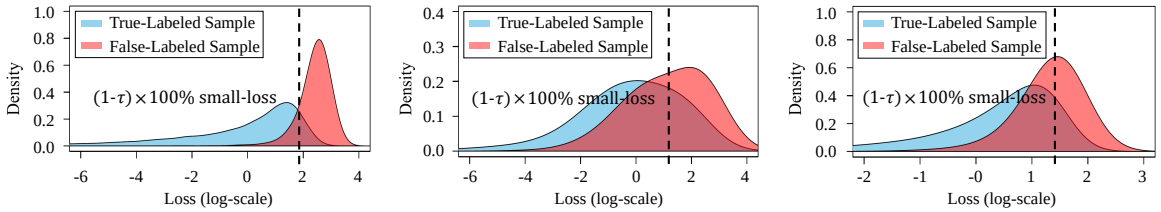
In this regard, as shown in Figure 2(a), we thoroughly investigated the memorization effect of a DNN on the two types of noises and found two interesting properties as follows:

- **A noise type affects the memorization rate for false-labeled samples:** The memorization rate for false-labeled samples is faster with pair noise than with symmetric noise. That is, the red portion in Figure 2(a) starts to appear earlier in pair noise than in symmetric noise. This observation supports the significant overlap of true-labeled and false-labeled samples in Figure 1(b). Thus, the loss-based separation performs well only if the false-labeled samples are scarcely learned at an early stage of training, as in symmetric noise.
- **There is a period where the network accumulates the label noise severely:** Regardless of the noise type, the memorization of false-labeled samples significantly increases at a late stage of training. That is, the red portion in Figure 2(a) increases rapidly after the dashed line, in which we call the *error-prone period*. We note that the training in that period brings *no benefit*. The generalization performance of "Default" deteriorates sharply, as shown in Figure 2(c).

Based on these findings, we contend that eliminating this error-prone period should make a profound impact on robust optimization. In this paper, we propose a novel approach, called **Prestopping**, that achieves *noise-free* training based

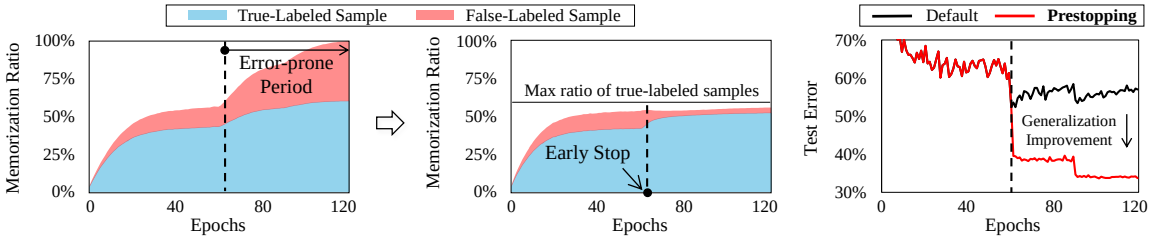
[1]Graduate School of Knowledge Service Engineering, Daejoen, Korea. Correspondence to: Jae-Gil Lee <jaegil@kaist.ac.kr>.

Presented at the ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning.



(a) Symmetric Noise. (b) Pair Noise. (c) Real-World Noise.

Figure 1. Loss distributions at a training accuracy of 50%: (a) and (b) show those on CIFAR-100 with two types of synthetic noises of 40%, where "symmetric noise" flips a true label into other labels with equal probability, and "pair noise" flips a true label into a specific false label; (c) shows those on FOOD-101N (Lee et al., 2018) with the real-world noise of 18.4%.



(a) Default. (b) Prestopping. (c) Test Error Convergence.

Figure 2. Key idea of *Prestopping*: (a) and (b) show how many true-labeled and false-labeled samples are memorized when training DenseNet (L=40, k=12)[2] on CIFAR-100 with the pair noise of 40%. "Default" is a standard training method, and "*Prestopping*" is our proposed one; (c) contrasts the convergence of test error between the two methods.

on the *early stopping* mechanism. Because there is no benefit from the error-prone period, *Prestopping* early stops training before that period begins. This early stopping effectively prevents a network from overfitting to false-labeled samples, and the samples memorized until that point are added to a *maximal safe set* because they are true-labeled (i.e., blue in Figure 2(a)) with high precision. Then, *Prestopping* resumes training the early stopped network *only* using the maximal safe set in support of noise-free training. Notably, our proposed merger of "early stopping" and "learning from the maximal safe set" indeed eliminates the error-prone period from the training process, as shown in Figure 2(b). As a result, the generalization performance of a DNN remarkably improves in both noise types, as shown in Figure 2(c).

2. Preliminaries

A k-class classification problem requires the training data $\mathcal{D} = \{x_i, y_i^*\}_{i=1}^N$, where x_i is a sample and $y_i^* \in \{1, 2, \dots, k\}$ is its *true* label. Following the label noise scenario, let's consider the noisy training data $\tilde{\mathcal{D}} = \{x_i, \tilde{y}_i\}_{i=1}^N$, where $\tilde{y}_i \in \{1, 2, \dots, k\}$ is a *noisy* label which may not be true. Then, in conventional training, when a mini-batch $\mathcal{B}_t = \{x_i, \tilde{y}_i\}_{i=1}^b$ consists of b samples randomly drawn from the noisy training data $\tilde{\mathcal{D}}$ at time t, the network parameter θ_t is updated in the descent direction of the expected loss on the mini-batch $\mathcal{B}_t$ as in Eq. (1), where α is a learning rate and $\mathcal{L}$ is a loss function.

$$\theta_{t+1} = \theta_t - \alpha \nabla \Big(\frac{1}{|\mathcal{B}_t|} \sum_{x \in \mathcal{B}_t} \mathcal{L}(x, \tilde{y}; \theta_t) \Big) \tag{1}$$

[2]The learning rate, as usual, was decayed at 50% and 75% of the total number of training epochs.

As for the notion of network memorization, a sample x is defined to be *memorized* by a network if the majority of its recent predictions at time t coincide with the given label, as in Definition 2.1.

Definition 2.1. (Memorized Sample) Let $\hat{y}_t = \Phi(x|\theta_t)$ be the predicted label of a sample x at time t and $H_x^t(q) = \{\hat{y}_{t_1}, \hat{y}_{t_2}, \dots, \hat{y}_{t_q}\}$ be the history of the sample x that stores the predicted labels of the recent q epochs, where Φ is a neural network. Next, $P(y|x,t;q)$ is formulated such that it provides the probability of the label $y \in \{1, 2, ..., k\}$ estimated as the label of the sample x based on H_x^t as in Eq. (2), where $[\cdot]$ is the Iverson bracket[3].

$$P(y|x,t;q) = \frac{\sum_{\hat{y} \in H_x^t(q)} [\hat{y} = y]}{|H_x^t(q)|} \tag{2}$$

Then, the sample x with its noisy label $\tilde{y}$ is a *memorized sample* of the network with the parameter θ_t at time t if $\text{argmax}_y P(y|x,t;q) = \tilde{y}$ holds. □

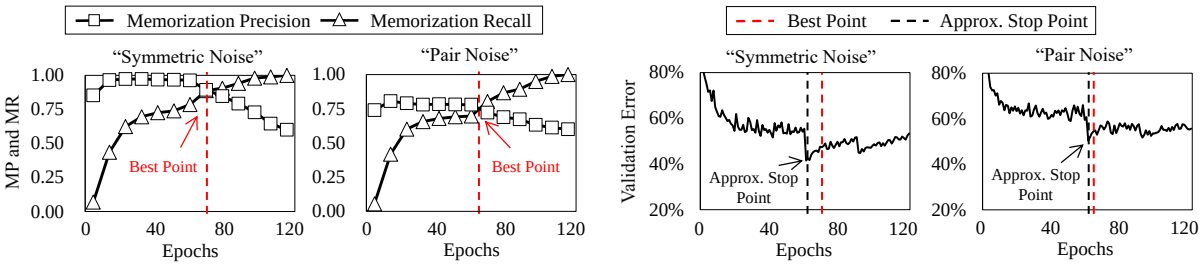
3. Robust training via *Prestopping*

The key idea of *Prestopping* is learning from a maximal safe set with an early stopped network. Thus, the two components of "early stopping" and "learning from the maximal safe set" respectively raise the questions about **(Q1)** when is the best point to early stop the training process? and **(Q2)** what is the maximal safe set to enable noise-free training during the remaining period?

3.1. Q1: Best point to early stop

It is desirable to stop the training process at the point when the network *(i)* not only accumulates *little* noise from the

[3]The Iverson bracket $[P]$ returns 1 if P is true; 0 otherwise.



(a) Ideal Method. (b) Validation Heuristic.

Figure 3. Early stop point estimated by ideal and heuristic methods when training DenseNet (L=40, k=12) on CIFAR-100 with two types of synthetic noises of 40%: (a) and (b) show the stop point derived by the ground-truth labels and the clean validation set, respectively.

false-labeled samples, *(ii)* but also acquires *sufficient* information from the true-labeled ones. Intuitively speaking, as indicated by the dashed line in Figure 3(a), the best stop point is the moment when *memorization precision* and *memorization recall* in Definition 3.1 cross with each other because it is the best trade-off between the two metrics. The period beyond this point is what we call the *error-prone period* because memorization precision starts decreasing rapidly.

If the ground truth y^* of a noisy label $\tilde{y}$ is known, the best stop point can be easily calculated by these metrics.

Definition 3.1. (Memorization Metrics) Let $\mathcal{M}_t \subseteq \tilde{\mathcal{D}}$ be a set of memorized samples at time t. Then, *memorization precision (MP)* and *recall (MR)* (Han et al., 2018) are formulated as Eq. (3).

$$MP = \frac{|\{(x,\tilde{y}) \in \mathcal{M}_t : \tilde{y} = y^*\}|}{|\mathcal{M}_t|}, \; MR = \frac{|\{(x,\tilde{y}) \in \mathcal{M}_t : \tilde{y} = y^*\}|}{|\{(x,\tilde{y}) \in \tilde{\mathcal{D}} : \tilde{y} = y^*\}|} \quad \square \tag{3}$$

However, it is not straightforward to find the exact best stop point *without* the ground-truth labels. Hence, we present two practical heuristics of approximating the best stop point with *minimal supervision*. These two heuristics require either a small clean validation set or a noise rate τ for the minimal supervision, where they are widely regarded as available in many studies (Veit et al., 2017; Ren et al., 2018; Han et al., 2018; Yu et al., 2019; Song et al., 2019).

- **Validation Heuristic:** If a clean validation set is given, we stop training the network when the validation error is the *lowest*. It is reasonable to expect that the lowest validation error is achieved near the cross point; after that point (i.e., in the error-prone period), the validation error likely increases because the network will be overfitted to the false-labeled samples. As shown in Figure 3(b), the estimated stop point is fairly close to the best stop point.
- **Noise-Rate Heuristic:** If a noise rate τ is known, we stop training the network when the training error reaches $\tau \times 100\%$. If we assume that all true-labeled samples of $(1-\tau)\times 100\%$ are memorized before any false-labeled samples of $\tau \times 100\%$ (Arpit et al., 2017), this point indicates the cross point of MP and MR with their values to be all 1. This heuristic performs worse than the validation heuristic because the assumption does not hold perfectly.

3.2. Q2: Criterion of a maximal safe set

Because the network is early stopped at the (estimated) best point, the set of memorized samples at that time is quantitatively sufficient and qualitatively less noisy. That is, it can be used as a safe and effective training set to resume the training of the early stopped network without accumulating the label noise. Based on this intuition, we define a *maximal safe set* in Definition 3.2, which is initially derived from the memorized samples at the early stop point and gradually increased as well as purified along with the network's learning progress. In each mini-batch $\mathcal{B}_t$, the network parameter θ_t is updated using the current maximal safe set $\mathcal{S}_t$ as in Eq. (4), and subsequently a more refined maximal safe set $\mathcal{S}_{t+1}$ is derived by the updated network.

Definition 3.2. (Maximal Safe Set) Let t_{stop} be the early stop point. A *maximal safe set* $\mathcal{S}_t$ at time t is defined to be the set of the memorized samples of the network $\Phi(x;\theta_t)$ when $t \geq t_{stop}$. The network $\Phi(x;\theta_t)$ at $t = t_{stop}$ is the early stopped network, i.e., $\mathcal{S}_{t_{stop}} = \mathcal{M}_{t_{stop}}$, and the network $\Phi(x;\theta_t)$ at $t > t_{stop}$ is obtained by Eq. (4).

$$\theta_{t+1} = \theta_t - \alpha \nabla \Big(\frac{1}{|\mathcal{B}'_t|} \sum_{x \in \mathcal{B}'_t} \mathcal{L}(x, \tilde{y}; \theta_t) \Big) \tag{4}$$
$$\mathcal{B}'_t = \{x | x \in \mathcal{S}_t \cap \mathcal{B}_t\} \quad \square$$

Hence, *Prestopping* iteratively updates its maximal safe set at each iteration, thereby adding more hard-yet-informative clean samples, which was empirically proven by the gradual increase in MR after the transition point in Appendix A

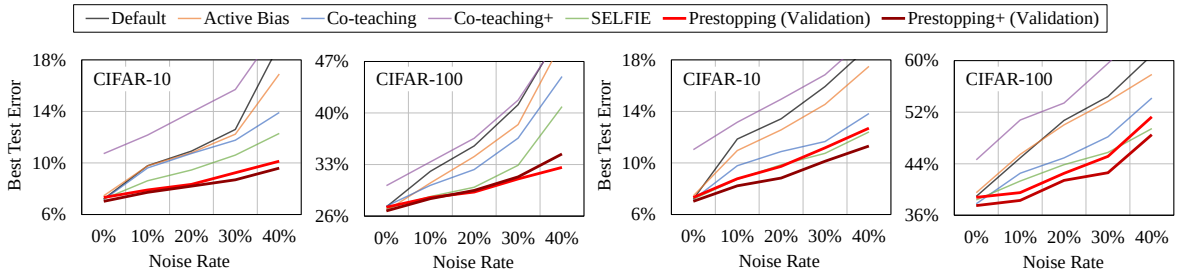
3.3. Algorithm

Because of the lack of space, we describe the overall procedure of *Prestopping* with the *validation* and *noise-rate* heuristics in Appendix B.

4. Evaluation

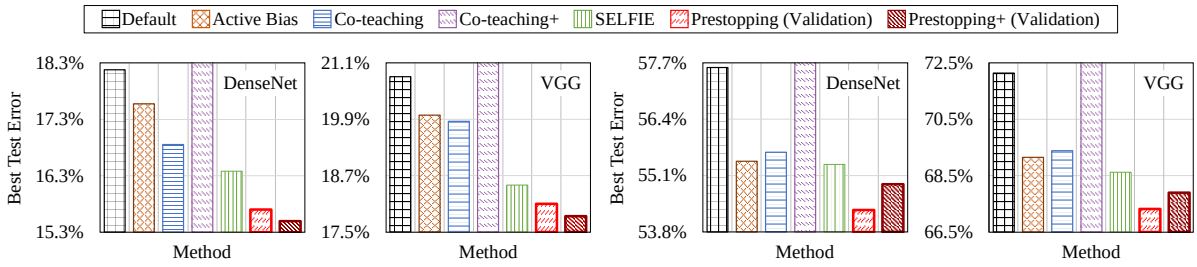
We compared *Prestopping* and *Prestopping+*[4] with not only a baseline algorithm but also the *four* state-of-the-art robust training algorithms: *Default* trains the network without any

[4]Collaboration with sample refurbishment in *SELFIE* (Song et al., 2019). See Appendix C for details.



(a) Pair Noise. (b) Symmetric Noise.

Figure 4. Best test errors using DenseNet on two data sets with varying **pair** and **symmetric** noise rates.



(a) ANIMAL-10N ($\tau \approx 8.0\%$). (b) FOOD-101N ($\tau \approx 18.4\%$).

Figure 5. Best test errors using two CNNs on two data sets with **real-world** noises.

processing for noisy labels; *Active Bias* re-weights the loss of training samples based on their prediction variance; *Co-teaching* selects a certain number of small-loss samples to train the network based on the co-training (Blum & Mitchell, 1998); *Co-teaching+* is similar to *Co-teaching*, but its small-loss samples are selected from the disagreement set; *SELFIE* selectively refurbishes noisy samples and exploits them together with the small-loss samples. Related work and experimental setup are detailed in Appendix D and E.

4.1. Performance Comparison

Figures 4 show the test error of the seven training methods using two CNNs on two data sets with varying *pair* and *symmetric* noise rates. Figure 5 shows the test error on two *real-world* noisy data sets with different noise rates. The results of *Prestopping* and *Prestopping+* in Section 4 were obtained using the *validation* heuristic. See Appendix B.2 for the results of the *noise-rate* heuristic.

4.1.1. Result with Pair Noise (Figure 4(a))

In general, either *Prestopping* or *Prestopping+* achieved the lowest test error in a wide range of noise rates on both CIFAR data sets. With help of the refurbished samples, *Prestopping+* achieved a slightly better performance than *Prestopping* in CIFAR-10. However, an opposite trend was observed in CIFAR-100. Although *SELFIE* achieved relatively lower test error among the existing methods, the test error of *SELFIE* was still worse than that of *Prestopping*. *Co-teaching* did not work well because many false-labeled samples were misclassified as clean ones; *Co-teaching+* was shown to be even worse than *Co-teaching* despite it being an improvement of *Co-teaching* (see Section F.2 for details). The test error of *Active Bias* was not comparable to that of *Prestopping*. The performance improvement of ours over the others increased as the label noise became heavier. In particular, at a heavy noise rate of 40%, *Prestopping* or *Prestopping+* significantly reduced the *absolute* test error by $2.2pp$–$18.1pp$ compared with the other robust methods.

4.1.2. Result with Symmetric Noise (Figure 4(b))

Similar to the pair noise, both *Prestopping* and *Prestopping+* generally outperformed the others. Particularly, the performance of *Prestopping+* was the best at any noise rate on all data sets, because the synergistic effect was higher in symmetric noise than in pair noise. At a heavy noise rate of 40%, our methods showed significant reduction in the *absolute* test error by $0.3pp$–$11.7pp$ compared with the other robust methods. Unlike the pair noise, *Co-teaching* and *SELFIE* achieved a low test error comparable to *Prestopping*.

4.1.3. Result with Real-World Noise (Figure 5)

Both *Prestopping* and *Prestopping+* maintained their dominance over the other methods under *real-world* label noise as well. *Prestopping+* achieved the lowest test error when the number of classes is small (e.g., ANIMAL-10N), while *Prestopping* was the best when the number of classes is large (e.g., FOOD-101N) owing to the difficulty in label correction of *Prestopping+*. (Thus, practitioners are recommended to choose between *Prestopping* and *Prestopping+* depending on the number of classes in hand.) Specifically, they improved the *absolute* test error by $0.4pp$–$4.6pp$ and $0.5pp$–$8.2pp$ in ANIMAL-10N and FOOD-101N, respectively. Therefore, we believe that the advantage of our methods is unquestionable even in the real-world scenarios.

5. Conclusion

In this paper, we proposed a novel two-phase training strategy for the noisy training data, which we call *Prestopping*. The first phase, "early stopping," retrieves an initial set of true-labeled samples as many as possible, and the second phase, "learning from a maximal safe set," completes the rest training process only using the true-labeled samples with high precision. *Prestopping* can be easily applied to many real-world cases because it additionally requires only either a small clean validation set or a noise rate. Furthermore, we combined this novel strategy with sample refurbishment to develop *Prestopping+*. Through extensive experiments using various real-world and simulated noisy data sets, we verified that either *Prestopping* or *Prestopping+* achieved the lowest test error among the seven compared methods, thus significantly improving the robustness to diverse types of label noise. Overall, we believe that our work of dividing the training process into two phases by early stopping is a new direction for robust training and can trigger a lot of subsequent studies.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00862, DB4DL: High-Usability and Performance In-Memory Distributed DBMS for Deep Learning).

References

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. A closer look at memorization in deep networks. In *ICML*, pp. 233–242, 2017.

Blum, A. and Mitchell, T. Combining labeled and unlabeled data with co-training. In *COLT*, pp. 92–100, 1998.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101–mining discriminative components with random forests. In *ECCV*, pp. 446–461, 2014.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. LOF: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pp. 93–104. ACM, 2000.

Chang, H.-S., Learned-Miller, E., and McCallum, A. Active Bias: Training more accurate neural networks by emphasizing high variance samples. In *NeurIPS*, pp. 1002–1012, 2017.

Chen, P., Liao, B. B., Chen, G., and Zhang, S. Understanding and utilizing deep neural networks trained with noisy labels. In *ICML*, pp. 1062–1070, 2019.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pp. 8536–8546, 2018.

Hendrycks, D., Lee, K., and Mazeika, M. Using pre-training can improve model robustness and uncertainty. In *ICML*, pp. 2712–2721, 2019.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pp. 2309–2318, 2018.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *NeurIPS*, pp. 1097–1105, 2012.

Krizhevsky, A., Nair, V., and Hinton, G. CIFAR-10 and CIFAR-100 datasets, 2014. https://www.cs.toronto.edu/~kriz/cifar.html.

Lee, K.-H., He, X., Zhang, L., and Yang, L. Cleannet: Transfer learning for scalable image classifier training with label noise. In *CVPR*, pp. 5447–5456, 2018.

Li, M., Soltanolkotabi, M., and Oymak, S. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. *arXiv preprint arXiv:1903.11680*, 2019.

Ma, X., Wang, Y., Houle, M. E., Zhou, S., Erfani, S. M., Xia, S.-T., Wijewickrema, S., and Bailey, J. Dimensionality-driven learning with noisy labels. In *ICML*, pp. 3361–3370, 2018.

Malach, E. and Shalev-Shwartz, S. Decoupling "when to update" from "how to update". In *NeurIPS*, pp. 960–970, 2017.

Oymak, S., Fabian, Z., Li, M., and Soltanolkotabi, M. Generalization guarantees for neural networks via harnessing the low-rank structure of the jacobian. *arXiv preprint arXiv:1906.05392*, 2019.

Patrini, G., Rozza, A., Menon, A. K., Nock, R., and Qu, L. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pp. 2233–2241, 2017.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *CVPR*, pp. 779–788, 2016.

Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. In *ICLR*, 2015.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *ICML*, pp. 4334–4343, 2018.

Shen, Y. and Sanghavi, S. Learning with bad training data via iterative trimmed loss minimization. In *ICML*, pp. 5739–5748, 2019.

Song, H., Kim, M., and Lee, J.-G. SELFIE: Refurbishing unclean samples for robust deep learning. In *ICML*, pp. 5907–5915, 2019.

Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., and Belongie, S. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, pp. 839–847, 2017.

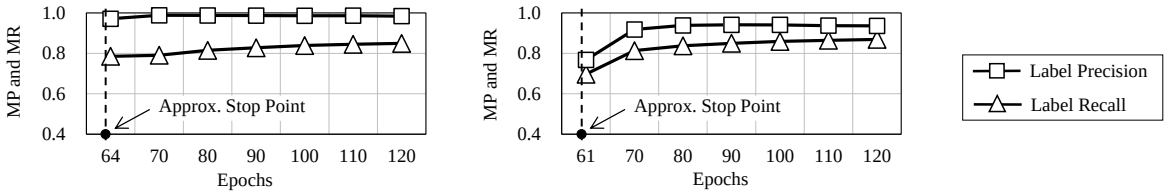
Wang, Y., Liu, W., Ma, X., Bailey, J., Zha, H., Song, L., and Xia, S.-T. Iterative learning with open-set noisy labels. In *CVPR*, pp. 8688–8696, 2018.

Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., and Sugiyama, M. How does disagreement help generalization against label corruption? In *ICML*, pp. 7164–7173, 2019.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

A. Ablation Study on Learning from the Maximal Safe Set

Figure 6 shows the main advantage of learning from the maximal safe set during the remaining period. Even when the noise rate is quite high (e.g., 40%), this learning paradigm exploits most of true-labeled samples, in considering that the label recall of the maximal safe set was maintained over 0.81 in symmetric noise and over 0.84 in pair noise after the 80th epoch. Further, by excluding unsafe samples that might accumulate the label noise, the label precision of the maximal safe set increases rapidly at the beginning of the remaining period; it was maintained over 0.99 in symmetric noise and over 0.94 even in pair noise after the 80th epoch, which could *not* be realized by the small-loss separation (Han et al., 2018; Song et al., 2019).



(a) Symmetric Noise. (b) Pair Noise.

Figure 6. Memorization precision and Memorization recall of the maximal safe set during the remaining epochs when training DenseNet (L=40, k=12) on CIFAR-100 with two types of synthetic noises of 40%.

B. Algorithm Description

Algorithm 1 *Prestopping* with **Validation** Heuristic

INPUT: $\tilde{\mathcal{D}}$: data, $\mathcal{V}$: clean validation data, $epochs$: total number of epochs, q: history length
OUTPUT: θ_t: network parameters

1: $t \leftarrow 1$; $\theta_t \leftarrow$ Initialize the network parameter;
2: $\theta_{t_{stop}} \leftarrow \emptyset$; /* The parameter of the stopped network */
3: **for** $i = 1$ **to** $epochs$ **do** /* **Phase I: Learning from a noisy training data set** */
4: **for** $j = 1$ **to** $|\tilde{\mathcal{D}}|/|\mathcal{B}_t|$ **do**
5: Draw a mini-batch $\mathcal{B}_t$ from $\tilde{\mathcal{D}}$;
6: $\theta_{t+1} = \theta_t - \alpha\nabla\big(\frac{1}{|\mathcal{B}_t|}\sum_{x\in\mathcal{B}_t}\mathcal{L}(x,\tilde{y};\theta_t)\big)$; /* Update by Eq. (1) */
7: $val\_err \leftarrow$ Get_Validation_Error($\mathcal{V}$, θ_t); /* A validation error at time t */
8: **if** $isMin(val\_err)$ **then** $\theta_{t_{stop}} \leftarrow \theta_t$; /* Save the network when val_error is the lowest */
9: $t \leftarrow t + 1$;
10: $\theta_t \leftarrow \theta_{t_{stop}}$; /* Load the network stopped at t_{stop} */
11: **for** $i = stop\_epoch$ **to** $epochs$ **do** /* **Phase II: Learning from a maximal safe set** */
12: **for** $j = 1$ **to** $|\tilde{\mathcal{D}}|/|\mathcal{B}_t|$ **do**
13: Draw a mini-batch $\mathcal{B}_t$ from $\tilde{\mathcal{D}}$;
14: $\mathcal{S}_t \leftarrow \{x|\text{argmax}_y P(y|x,t;q) = \tilde{y}\}$; /* A maximal safe set in Definition 3.2 */
15: $\theta_{t+1} = \theta_t - \alpha\nabla\big(\frac{1}{|\mathcal{S}_t\cap\mathcal{B}_t|}\sum_{x\in\mathcal{S}_t\cap\mathcal{B}_t}\mathcal{L}(x,\tilde{y};\theta_t)\big)$; /* Update by Eq. (4) */
16: $t \leftarrow t + 1$;
17: **return** θ_t, $\mathcal{S}_t$;

B.1. *Prestopping* with Validation Heuristic (Main Algorithm)

Algorithm 1 describes the overall procedure of *Prestopping* with the *validation* heuristic, which is self-explanatory. First, the network is trained on the noisy training data $\tilde{\mathcal{D}}$ in the *default* manner (Lines 3–6). During this first phase, the validation data $\mathcal{V}$ is used to evaluate the best point for the early stop, and the network parameter is saved at the time of the lowest validation error (Lines 7–8). Then, during the second phase, *Prestopping* continues to train the early stopped network for the remaining learning period (Lines 10–12). Here, the maximal safe set $\mathcal{S}_t$ at the current moment is retrieved, and each sample $x \in \mathcal{S}_t \cap \mathcal{B}_t$ is used to update the network parameter. The mini-batch samples not included in $\mathcal{S}_t$ are no longer used in pursuit of robust learning (Lines 14–15).

Algorithm 2 *Prestopping* with **Noise-Rate** Heuristic

INPUT: $\tilde{\mathcal{D}}$: data, $epochs$: total number of epochs, q: history length, τ: noise rate
OUTPUT: θ_t: network parameters

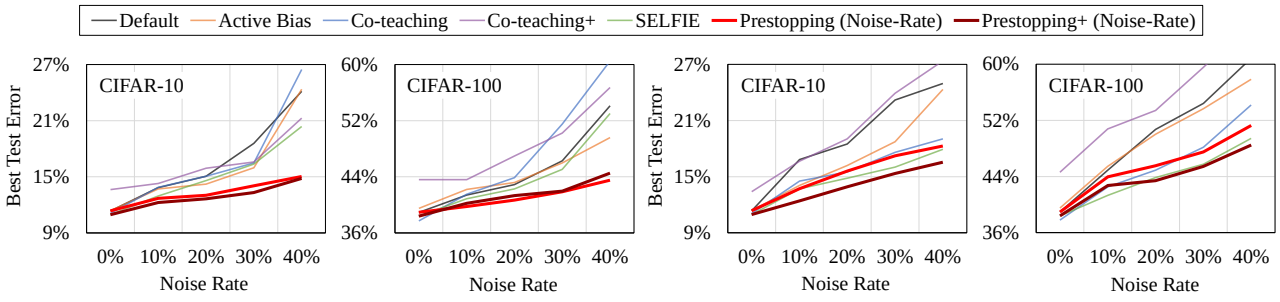
1: $t \leftarrow 1$; $\theta_t \leftarrow$ Initialize the network parameter;
2: $\theta_{t_{stop}} \leftarrow \emptyset$; /* The parameter of the stopped network */
3: **for** $i = 1$ **to** $epochs$ **do** /* **Phase I: Learning from a noisy training data set** */
4: **for** $j = 1$ **to** $|\tilde{\mathcal{D}}|/|\mathcal{B}_t|$ **do**
5: Draw a mini-batch $\mathcal{B}_t$ from $\tilde{\mathcal{D}}$;
6: $\theta_{t+1} = \theta_t - \alpha\nabla\big(\frac{1}{|\mathcal{B}_t|}\sum_{x\in\mathcal{B}_t}\mathcal{L}(x,\tilde{y};\theta_t)\big)$; /* Update by Eq. (1) */
7: $train\_err \leftarrow$ Get_Training_Error($\tilde{\mathcal{D}}$, θ_t); /* A training error at time t */
8: **if** $train\_err \leq \tau$ **then** /* Save the network when $train\_err \leq \tau$ */
9: $\theta_{t_{stop}} \leftarrow \theta_t$; break;
10: $t \leftarrow t + 1$;
11: $\theta_t \leftarrow \theta_{t_{stop}}$; /* Load the network stopped at t_{stop} */
12: **for** $i = stop\_epoch$ **to** $epochs$ **do** /* **Phase II: Learning from a maximal safe set** */
13: **for** $j = 1$ **to** $|\tilde{\mathcal{D}}|/|\mathcal{B}_t|$ **do**
14: Draw a mini-batch $\mathcal{B}_t$ from $\tilde{\mathcal{D}}$;
15: $\mathcal{S}_t \leftarrow \{x|\text{argmax}_y P(y|x,t;q) = \tilde{y}\}$; /* A maximal safe set in Definition 3.2 */
16: $\theta_{t+1} = \theta_t - \alpha\nabla\big(\frac{1}{|\mathcal{S}_t\cap\mathcal{B}_t|}\sum_{x\in\mathcal{S}_t\cap\mathcal{B}_t}\mathcal{L}(x,\tilde{y};\theta_t)\big)$; /* Update by Eq. (4) */
17: $t \leftarrow t + 1$;
18: **return** θ_t, $\mathcal{S}_t$;

B.2. *Prestopping* with Noise-Rate Heuristic

Algorithm 2 describes the overall procedure of *Prestopping* with the *noise-rate* heuristic, which is also self-explanatory. Compared with Algorithm 1, only the way of determining the best stop point in Lines 7–9 has changed.

B.2.1. Result with Synthetic Noise (Figure 7)

To verify the performance of *Prestopping* and *Prestopping+* with the *noise-rate* heuristic in Section 3.1, we trained a VGG-19 network on two simulated noisy data sets with the same configuration as in Section 4. Figure 7 shows the test error of our two methods using the noise-rate heuristic as well as those of the other five training methods. Again, the test error of either *Prestopping* or *Prestopping+* was the lowest at most error rates with any noise type. The trend of the noise-rate heuristic here was almost the same as that of the validation heuristic in Section 4.1. Especially when the noise rate was 40%, *Prestopping* and *Prestopping+* significantly improved the test error by $5.1pp$–$17.0pp$ in the pair noise (Figure 7(a)) and $0.3pp$–$17.3pp$ in the symmetric noise (Figure 7(b)) compared with the other robust methods.



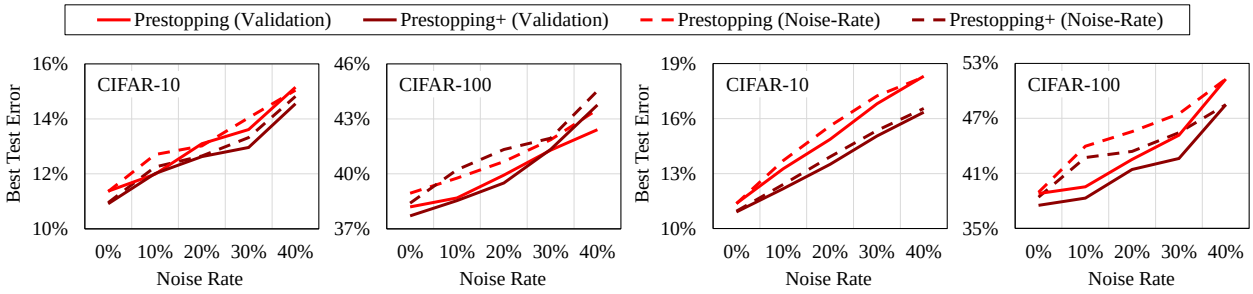
(a) Pair Noise. (b) Symmetric Noise.

Figure 7. Best test errors using VGG-19 on two simulated noisy data sets with varying noise rates.

B.2.2. Comparison with Validation Heuristic (Figure 8)

Figure 8 shows the difference in test error caused by the two heuristics. Overall, the performance with the noise-rate heuristic was worse than that with validation heuristic, even though the worse one outperformed the other training methods as shown in Figure 7. As the assumption of the noise-rate heuristic does not hold perfectly, a lower performance of the noise-rate

heuristic looks reasonable. However, we expect that the performance with this heuristic can be improved by stopping a little earlier than the estimated point, and we leave this extension as the future work.



(a) Pair Noise. (b) Symmetric Noise.

Figure 8. Difference in test error between two heuristics on two simulated noisy data sets.

C. Collaboration with Sample Refurbishment: *Prestopping+*

To further improve the performance of *Prestopping*, we introduce *Prestopping+* that employs the concept of selectively refurbishing false-labeled samples in *SELFIE* (Song et al., 2019). In detail, the final maximal safe set $\mathcal{S}_{t_{end}}$ is retrieved from the first run of *Prestopping*, and then it is used as the true-labeled set for the next run of *SELFIE* with initializing the network parameter. Following the update principle of *SELFIE*, the modified gradient update rule in Eq. (5) is used to train the network. For each sample x in the mini-batch $\mathcal{B}_t$, the given label $\tilde{y}$ of x is selectively refurbished into y^{refurb} if it can be corrected with high precision. However, the mini-batch samples included in $\mathcal{S}_{t_{end}}$ are omitted from the refurbishment because their label is already highly credible. Then, the mini-batch samples in the refurbished set $\mathcal{R}_t$ are provided to update the network together with those in the final maximal safe set $\mathcal{S}_{t_{end}}$. Refer to Song et al. (2019)'s work for more details.

$$\theta_{t+1} = \theta_t - \alpha\nabla\Big(\frac{1}{|\{x|x \in \mathcal{R}_t \cap \mathcal{S}_{t_{end}}\}|}\Big(\sum_{x\in\mathcal{R}_t\cap\mathcal{S}^c_{t_{end}}} \mathcal{L}(x, y^{refurb}; \theta_t) + \sum_{x\in\mathcal{S}_{t_{end}}} \mathcal{L}(x, \tilde{y}; \theta_t)\Big)\Big) \tag{5}$$

D. Related Work

Numerous studies have been conducted to address the problem of learning from noisy labels. A typical method is using "loss correction" that estimates the label transition matrix and corrects the loss of the samples in the mini-batch. *Bootstrap* (Reed et al., 2015) updates the network based on their own reconstruction-based objective with the notion of perceptual consistency. *F-correction* (Patrini et al., 2017) reweights the forward or backward loss of the training samples based on the label transition matrix estimated by a pre-trained normal network. *D2L* (Ma et al., 2018) employs a simple measure called local intrinsic dimensionality and then uses it to modify the forward loss in order to reduce the effects of false-labeled samples in learning.

Active Bias (Chang et al., 2017) heuristically evaluates uncertain samples with high prediction variances and then gives higher weights to their backward losses. Ren et al. (2018) include small clean validation data into the training data and re-weight the backward loss of the mini-batch samples such that the updated gradient minimizes the loss of those validation data. However, this family of methods is known to accumulate severe noise from the *false correction*, especially when the number of classes or the number of false-labeled samples is large (Han et al., 2018; Yu et al., 2019).

To be free from the false correction, many recent researches have adopted "sample selection" that trains the network on selected samples. These methods attempt to select true-labeled samples from the noisy training data for use in updating the network. *Decouple* (Malach & Shalev-Shwartz, 2017) maintains two networks simultaneously and updates the models only using the samples that have different label predictions from these two networks. Wang et al. (2018) proposed an iterative learning framework that learns deep discriminative features from well-classified noisy samples based on the local outlier factor algorithm (Breunig et al., 2000).

MentorNet (Jiang et al., 2018) introduces a collaborative learning paradigm that a pre-trained mentor network guides the training of a student network. Based on the small-loss criteria, the mentor network provides the student network with the samples whose label is probably correct. *Co-teaching* (Han et al., 2018) and *Co-teaching+* (Yu et al., 2019) also maintain

two networks, but each network selects a certain number of small-loss samples and feeds them to its peer network for further training. Compared with *Co-teaching*, *Co-teaching+* further employs the disagreement strategy of *Decouple*. *ITLM* (Shen & Sanghavi, 2019) iteratively minimizes the trimmed loss by alternating between selecting a fraction of small-loss samples at current moment and retraining the network using them. *INCV* (Chen et al., 2019) randomly divides the noisy training data and then utilizes cross-validation to classify true-labeled samples with removing large-loss samples at each iteration. However, their general philosophy of selecting *small-loss* samples works well only in some cases such as symmetric noise. Differently to this family, we exploit the maximal safe set initially derived from the memorized samples at the early stop point.

Most recently, a hybrid of "loss correction" and "sample selection" approaches was proposed by Song et al. (2019). Their algorithm called *SELFIE* trains the network on selectively refurbished false-labeled samples together with small-loss samples. *SELFIE* not only minimizes the number of falsely corrected samples but also exploits full exploration of the entire training data. Its component for sample selection can be replaced by our method to further improve the performance.

For the completeness of the survey, we mention the work that provides an empirical or a theoretical analysis on why early stopping helps learn with the label noise. Oymak et al. (2019) and Hendrycks et al. (2019) have argued that early stopping is a suitable strategy because the network eventually begins to memorize all noisy samples if it is trained too long. Li et al. (2019) have theoretically proved that the network memorizes false-labeled samples at a later stage of training, and thus claimed that the early stopped network is fairly robust to the label noise. However, they did not mention how to take advantage of the early stopped network for further training. Please note that *Prestopping* adopts early stopping to derive a *seed* for the maximal safe set, which is exploited to achieve noise-free training during the remaining learning period. Thus, our novelty lies in the "merger" of early stopping and learning from the maximal safe set.

E. Experimental Configuration

All the algorithms were implemented using TensorFlow $1.8.0$ and executed using 16 NVIDIA Titan Volta GPUs. For reproducibility, we provide the source code at `https://bit.ly/2l3g9Jx`. In support of reliable evaluation, we repeated every task *thrice* and reported the average and standard error of the best test errors, which are the common measures of robustness to noisy labels (Chang et al., 2017; Jiang et al., 2018; Ren et al., 2018; Song et al., 2019).

Data Sets: To verify the superiority of *Prestopping*, we performed an image classification task on *four* benchmark data sets: CIFAR-10 (10 classes)[5] and CIFAR-100 (100 classes)[4], a subset of 80 million categorical images (Krizhevsky et al., 2014); ANIMAL-10N (10 classes)[6], a real-world noisy data of human-labeled online images for confusing animals (Song et al., 2019); FOOD-101N (101 classes)[7], a real-world noisy data of crawled food images annotated by their search keywords in the FOOD-101 taxonomy (Bossard et al., 2014; Lee et al., 2018). We did not apply any data augmentation.

Noise Injection: As all labels in the CIFAR data sets are clean, we artificially corrupted the labels in these data sets using typical methods for the evaluation of synthetic noises (Ren et al., 2018; Han et al., 2018; Yu et al., 2019). For k classes, we applied the label transition matrix $\mathbf{T}$: *(i) symmetric noise*: $\forall_{j \neq i} \mathbf{T}_{ij} = \frac{\tau}{k-1}$ and *(ii) pair noise*: $\exists_{j \neq i} \mathbf{T}_{ij} = \tau \wedge \forall_{k \neq i, k \neq j} \mathbf{T}_{ik} = 0$, where $\mathbf{T}_{ij}$ is the probability of the true label i being flipped to the corrupted label j and τ is the noise rate. For the pair noise, the corrupted label j was set to be the next label of the true label i following the recent work (Yu et al., 2019; Song et al., 2019). To evaluate the robustness on varying noise rates from light noise to heavy noise, we tested five noise rates $\tau \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$. In contrast, we did not inject any label noise into ANIMAL-10N and FOOD-101N[8] because they contain real label noise estimated at 8.0% and 18.4% respectively (Lee et al., 2018; Song et al., 2019).

Clean Validation Data: Recall that a clean validation set is needed for the validation heuristic in Section 3.1. As for ANIMAL-10N and Food-101N, we did exploit their own clean validation data; $5,000$ and $3,824$ images, respectively, were included in their validation set. However, as no validation data exists for the CIFAR data sets, we constructed a small clean validation set by randomly selecting $1,000$ images from the *original* training data of $50,000$ images. Please note that the noise injection process was applied to only the rest $49,000$ training images.

Networks and Hyperparameters: For the classification task, we trained DenseNet (L=40, k=12) and VGG-19 with a momentum optimizer. Specifically, we used a momentum of 0.9, a batch size of 128, a dropout of 0.1, and batch

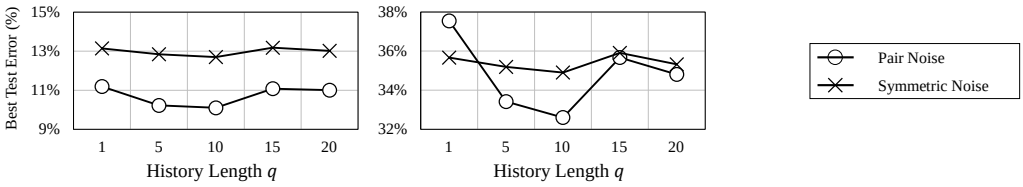
[5] https://www.cs.toronto.edu/ kriz/cifar.html
[6] https://dm.kaist.ac.kr/datasets/animal-10n
[7] https://kuanghuei.github.io/Food-101N
[8] In FOOD-101N, we used a subset of the entire training data marked with whether the label is correct or not.

normalization. *Prestopping* has only one unique hyperparameter, the history length q, and it was set to be 10, which was the best value found by the grid search (see Appendix F.1 for details). The hyperparameters used in the compared algorithms were favorably set to be the best values presented in the original papers. As for the training schedule, we trained the network for 120 epochs and used an initial learning rate 0.1, which was divided by 5 at 50% and 75% of the total number of epochs.

F. Supplementary Evaluation

F.1. Hyperparameter Selection (Figure 9)

Prestopping requires one additional hyperparameter, the history length q in Definition 2.1. For hyperparameter tuning, we trained DenseNet (L=40, k=12) on CIFAR-10 and CIFAR-100 with a noise rate of 40%, and the history length q was chosen in a grid $\in \{1, 5, 10, 15, 20\}$. Figure 9 shows the test error of *Prestopping* obtained by the grid search on two noisy CIFAR data sets. Regardless of the noise type, the lowest test error was typically achieved when the value of q was 10 in both data sets. Therefore, the history length q was set to be 10 in all experiments.

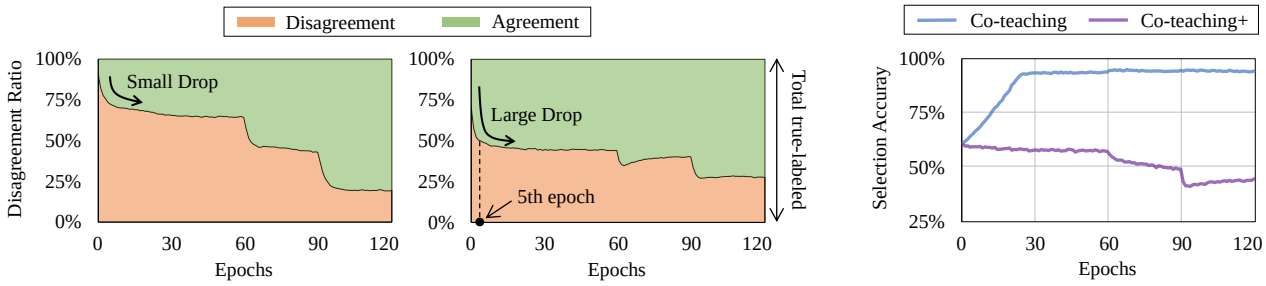


(a) CIFAR-10. (b) CIFAR-100.

Figure 9. Grid search on CIFAR-10 and CIFAR-100 with two types of noises of 40%.

F.2. Anatomy of *Co-teaching+* (Figure 10)

Although *Co-teaching+* is the latest method, its performance was worse than expected, as shown in Section 4.1. Thus, we looked into *Co-teaching+* in more detail. A poor performance of *Co-teaching+* was attributed to the fast consensus of the label predictions for true-labeled samples, especially when training a complex network. In other words, because two complex networks in *Co-teaching+* start making the same predictions for true-labeled samples too early, these samples are excluded too early. As shown in Figures 10(a) and 10(b), the disagreement ratio with regard to the true-labeled samples dropped faster with a complex network than with a simple network, in considering that the ratio drastically decreased to 49.8% during the first 5 epochs. Accordingly, it is evident that *Co-teaching+* with a complex network causes a *narrow exploration* of the true-labeled samples, and the selection accuracy of *Co-teaching+* naturally degraded from 60.4% to 44.8% for that reason, as shown in Figure 10(c). Therefore, we conclude that *Co-teaching+* may not suit a complex network.



(a) Simple Network. (b) Complex Network. (c) Selection Accuracy.

Figure 10. Anatomy of *Co-teaching+* on CIFAR-100 with 40% symmetric noise: (a) and (b) show the change in disagreement ratio for all true-labeled samples, when using *Co-teaching+* to train two networks with different complexity, where "simple network" is a network with seven layers used by Yu et al. (2019), and "complex network" is a DenseNet (L=40, k=12) used for our evaluation; (c) shows the accuracy of selecting true-labeled samples on the DenseNet.

G. Case Study: Noisy Labels in Original CIFAR-100

One interesting observation is a noticeable improvement of *Prestopping+* even when the noise rate was 0%. It was turned out that *Prestopping+* sometimes refurbished the labels of the false-labeled samples which were *originally* contained in the CIFAR data sets. Figure 11 shows a few successful refurbishment cases. For example, an image falsely annotated as a "Boy" was refurbished as a "Baby" (Figure 11(a)), and an image falsely annotated as a "Mouse" was refurbished as a "Hamster" (Figure 11(c)). Thus, this sophisticated label correction of *Prestopping+* helps overcome the residual label noise in well-known benchmark data sets, which are misconceived to be clean, and ultimately further improves the generalization performance of a network.

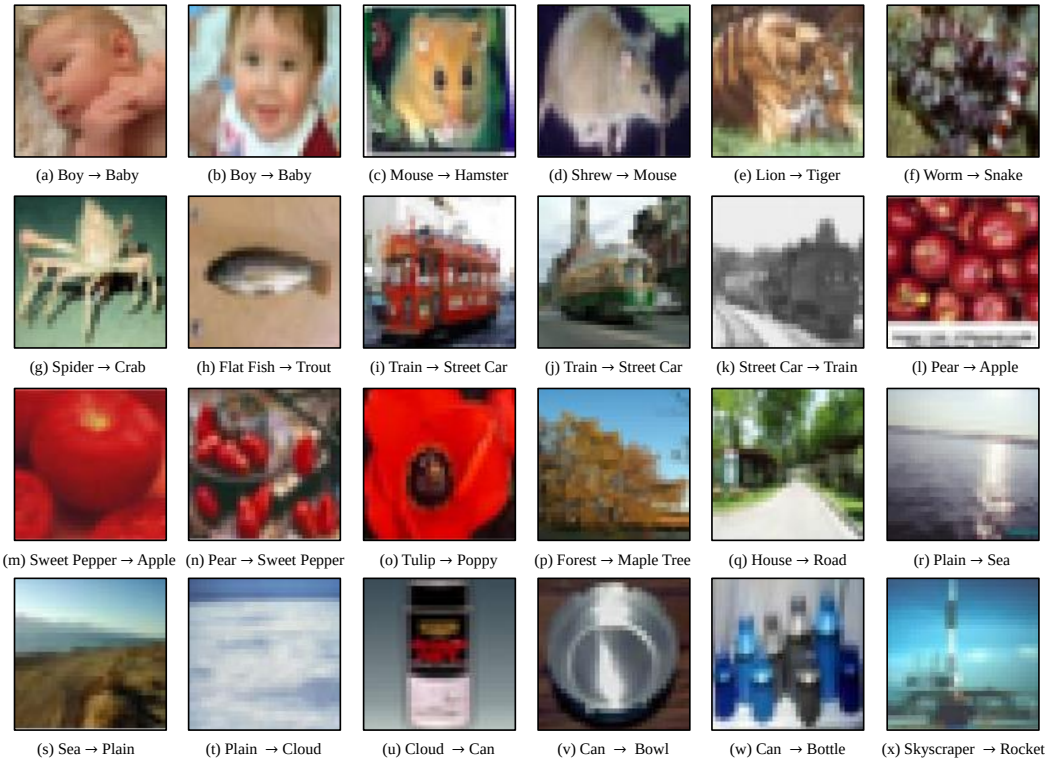


Figure 11. Refurbishing of false-labeled samples *originally* contained in CIFAR-100. The subcaption represents "original label" → "refurbished label" recognized by *Prestopping+*.