Learning from Noisy Labels with Complementary Loss Functions Supplementary Material

Deng-Bao Wang,[1,2] **Yong Wen,**[3] **Lujia Pan,**[3] **Min-Ling Zhang**[1,2,4*]

[1]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[2]Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China
[3]Noah's Ark Lab, Huawei Technologies
[4]Collaborative Innovation Center of Wireless Communications Technology, China
wangdb@seu.edu.cn, {wenyong4, panlujia}@huawei.com, zhangml@seu.edu.cn

Appendix A. Noise Generation

Uniform noise was generated by replacing a true label with a random label through uniform sampling for all datasets.

Class-conditional noise was generated by mapping TRUCK $\rightarrow$ AUTOMOBILE, BIRD $\rightarrow$ AIRPLANE, DEER $\rightarrow$ HORSE, and CAT $\leftrightarrow$ DOG with probability η for CIFAR-10. For CIFAR-100 and TinyImageNet, we simulated class-conditional noise by flipping each class into the next circularly with probability η.

Appendix B. *Mixup* Augmentation

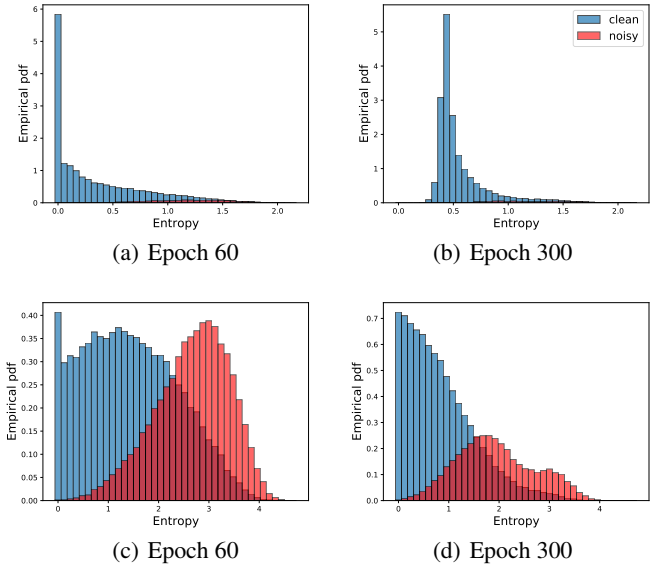
We use *Mixup* technique to train DNNs on convex combinations of sample pairs ($\boldsymbol{x}_1$ and $\boldsymbol{x}_2$) and their corresponding labels (y_1 and y_2). Specifically, a mixed sample pair can be computed by:

$$\begin{aligned} \boldsymbol{x}' &= \lambda' \boldsymbol{x}_1 + (1-\lambda')\boldsymbol{x}_2 \\ y' &= \lambda' y_1 + (1-\lambda') y_2 \end{aligned} \tag{1}$$

where λ' is randomly sampled from a beta distribution: $\lambda' = \text{Beta}(\sigma, \sigma)$. In this paper, we set $\sigma = 4$ for all data sets and noise levels.

Appendix C. Noise Reduction

In Methodology Section, we have present entropy distributions of correctly and incorrectly labeled examples on CIFAR-100. Here we also present entropy distributions on CIFAR-10 and TinyImageNet datasets in Figure 1. Furthermore, we report the mislabeling rates of the generated pseudo supervision over different phases in Table 1. As we can see, after filtering out a small set of hard samples with large entropy, the mislabeling rates drop to a low to relatively low level. Moreover, as shown in Table 1, the mis-labeling rates at the end of training phase are relatively lower than that at the end of warm-up phase, this indicates that our method can iteratively reduce the mislabeling rate in learning phase.



(a) Epoch 60 (b) Epoch 300

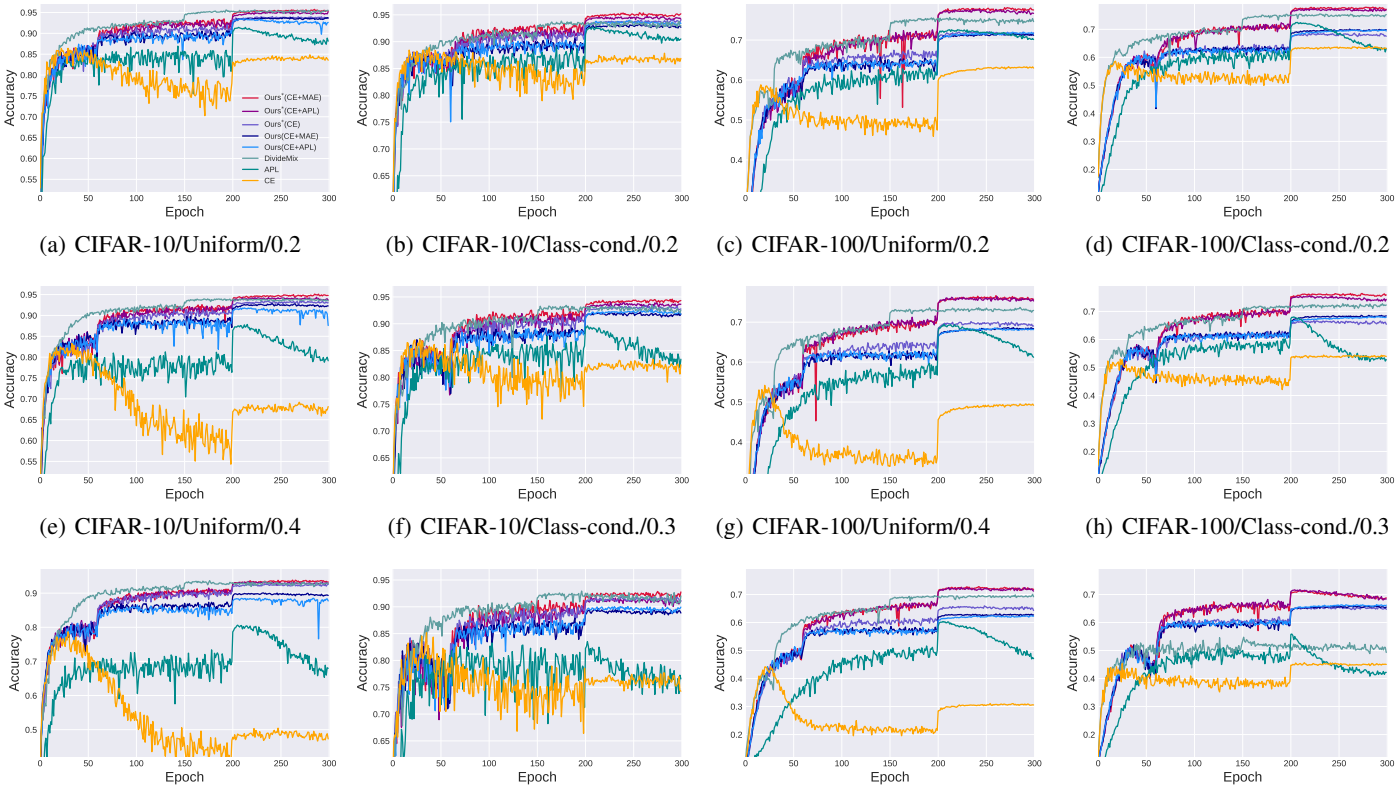
(c) Epoch 60 (d) Epoch 300

Figure 1. Entropy distributions of the average predictions of correctly and incorrectly pseudo labeled samples (trained on CIFAR-10 (top) and TinyImageNet (bottom) under 0.5 uniform noise).

Appendix D. Additional Comparison with DivideMix

DivideMix need to tune the hyperparameters λ_u for different noise types and rates. In our experiments, we use the suggested values of the original paper. The results of DivideMix can be significatly influence by the changing of λ_u. For example, in the case of 0.2 uniform noise on CIFAR-10, if we set $\lambda_u = 25$ then the obtained accuracy is 91.50, which drops about 4%. DivideMix trains two networks simultaneously, thus we can choose to average the predictions from both networks in test phase. For fair comparison, we use the prediction from a single network, and here we show that we can improve our method by using same averaging strategy. We train two networks

*Corresponding author



(a) CIFAR-10/Uniform/0.2 (b) CIFAR-10/Class-cond./0.2 (c) CIFAR-100/Uniform/0.2 (d) CIFAR-100/Class-cond./0.2

(e) CIFAR-10/Uniform/0.4 (f) CIFAR-10/Class-cond./0.3 (g) CIFAR-100/Uniform/0.4 (h) CIFAR-100/Class-cond./0.3

(i) CIFAR-10/Uniform/0.6 (j) CIFAR-10/Class-cond./0.4 (k) CIFAR-100/Uniform/0.6 (l) CIFAR-100/Class-cond./0.4

Figure 2. Test accuracy curves of different methods for different noise types and levels.

Table 1: Mis-labeling rate of the generated pseudo supervision over different phase.

		CIFAR-10	CIFAR-100	Tiny
epoch 60	all	8.25%	28.35%	41.38%
	filter 30%	1.11%	14.03%	26.29%
last epoch	all	4.96%	18.56%	31.77%
	filter 30%	0.36%	6.11%	16.04%

independently using complementary-loss on same synthetic noisy dataset and test the performance of both the first network and the ensemble of these two networks. As is shown in Table 2, the results of our method can be also improved by using this averaging strategy. In fact, for the 4 cases presented in Table 2, the single network learned using our method outperforms the ensemble strategy of DivideMix.

Appendix E. Performance Curves

We record the test accuracy at each training epoch to provide more detailed visualized results in Fiugre 5 (we randomly choose 1 trial from 3 runs). Note that for our method, we reduce the learning rate by a factor of 10 after 200 epochs, and we maintain the learning policy of DivieMix reported in their papers (in which the learning rate is reduced after 150 epoches). As shown in Figure 2, our proposed method consistently outperform other compared methods. After reducing the learning rate at epoch 200, the test accuracies of our method over different epochs are stable.

Table 2: Additional Comparison with DivideMix on `CIFAR-10` (top) and `CIFAR-100` (bottom). 2×Ours* means we train two networks using complementary-loss separately and average the predictions from both networks. Here we use the combination of CE and MAE as complementary losses.

Noise ratio	Uniform (0.4)	Class-cond. (0.3)
DivideMix(θ^1)	93.62±0.12	92.38±0.24
DivideMix(θ^1+θ^2)	94.23±0.12	93.05±0.26
Ours*	94.69±0.20	94.33±0.36
2×Ours*	**95.29± 0.14**	**95.23± 0.30**
DivideMix(θ^1)	72.92±0.20	72.14±0.31
DivideMix(θ^1+θ^2)	74.96±0.23	74.41±0.35
Ours*	75.80±0.33	75.45±0.42
2×Ours*	**78.34±0.13**	**78.03±0.43**