Supplementary Materials: Phase-Amplitude Spectrum Disentangled Early Stopping for Learning with Noisy Labels

Huaxi Huang[1*†], Hui Kang[2†], Sheng Liu[3], Olivier Salvado[1],
Thierry Rakotoarivelo[1], Dadong Wang[1], Tongliang Liu[2]
[1] Data61, CSIRO, [2] The University of Sydney, [3] NYU Center for Data Science

1. Study of Deep Features on the Frequency Domain

To investigate the impact of label noise on CNNs trained using different frequency components from different layers, we conducted experiments by training a ResNet-18 model [5] with different label noises. We generated three label noises: 50% symmetric noise [18, 4], 40% instance noise [19], and 45% pairflip noise [4]. We trained the ResNet-18 model under these label noises using PADDLES (Algorithm 1 in our paper) to disentangle and detach the AS/PS components of the deep features from different ResNet-18 blocks during CNNs training. As the ResNet-18 has four blocks, we present all deep features extracted from those blocks under three label noises in Figure 2 (50% symmetric label noise), Figure 3 (40% instance label noise), and Figure 4 (45% pairflip label noise). As shown in these figures, the deep features extracted by different ResNet-18 blocks share similar behavior with original images, where PS components of deep features can help the CNNs become more robust towards label noises than AS or raw deep features. These results strongly support the rationality and correctness of our solution for disentangling and manipulating the model training in the deep image features.

Moreover, we observe that this behavior is more evident for deeper features (features from Block-4 and Block-3) than for shallower ones (features from Block-2 and Block-1). An intuitive explanation is the gradient vanishing phenomenon of CNNs [6]. Due to the gradients being back-propagated, repeated multiplication and convolution with small weights render the gradient information ineffectively small in shallower blocks. Therefore, detaching the AS or PS-related gradient propagation in the shallower layers (Block-1 or Block-2) can result in a smaller impact on the model updating than in deeper layers (Block-3 or Block-4). These observations also guide the principle of disentangle point selection. A later disentangle point can achieve better performance in resisting label noise, which is supported by the following study of the disentangle position.

2. Study of Disentangle Position & Hyper-parameter

An important component of PADDLES is the frequency disentangling position j, as presented in Algorithm 2. We chose ResNet models as the backbone and disentangled the deep features at each ResNet block, with 'P1' indicating decomposition before block 1, 'P5' after block 4, and 'ALL' representing decomposition at all five positions. As shown in Figure 1a, we observed that the performance of PADDLES was more stable on CIFAR-10 than on CIFAR-100 at different positions, with the best performances achieved at P3 and P4.

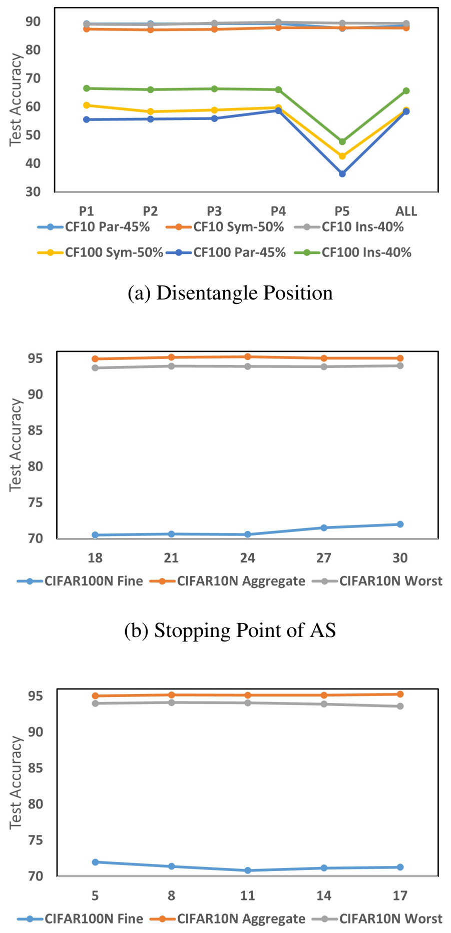
We investigated the hyper-parameter sensitivity of the early stopping points for amplitude spectrum T_A and phase spectrum T_P in Figures 1b and 1c. All experiments were conducted on CIFAR-N datasets with a ResNet-34 backbone. We varied T_A from 18 to 30 with $T_P = 5$ in Figure 1b, and set T_P from 5 to 17 with $T_A = 30$ in Figure 1c. We observed that with a fixed T_P, the performance generally increased as T_A grew for both Fine noise on CIFAR-100N and Worst noise on CIFAR-10N. When T_A was fixed, very large training steps for PS resulted in performance degradation, as the model started to overfit the label noises. Moreover, the performances of our model on CIFAR-10N dataset with Aggregate noise remained comparatively stable compared to other noises. The model achieved the best performance with $T_A = 30$ and $T_P = 5$.

3. Additional Experiments

In this section, we provide more experimental results to further demonstrate the effectiveness of our methods, including results on the WebVision dataset, training curves under different kinds of noise, and confident samples quality evaluation.

[*] Project lead.

[†] Co-first authors.



(a) Disentangle Position

(b) Stopping Point of AS

(c) Stopping Point of PS

Figure 1: Sensitivity analysis for different choices of disentangle positions, early stopping points of AS, and early stopping points of PS. The Y-axis of each figure represents the testing accuracy (%).

3.1. Experiment on WebVision Dataset

In this section, we present an experiment that we conducted on the WebVision 1.0 dataset [9] to evaluate the effectiveness of PADDLES on a large-scale dataset. The WebVision dataset comprises 2.4 million images sourced from the internet and categorized into 1,000 semantic concepts in ImageNet ILSCRC12 [15]. To enable easy comparison with prior research, we have followed the approach of [3, 8], which focused on the top 50 classes from the Google image subset of WebVision 1.0. To conduct the experiment, we employed InceptionResNetV2 [17] as the backbone, following prior work [3, 8, 10]. Additionally, we trained the model using two sub-networks ensembles, utilizing the SGD optimizer with a momentum of 0.9, and setting the weight decay at 10^{-3}, with a batch size of 32. We adopted a one-epoch warm-up training strategy, trained the model for 100 formal epochs, initialized the learning rate at 0.01, and used a three-phase OneCycle [16] learning rate scheduler. We set the disentangling point before the last convolutional layer of InceptionResNetV2, and T_A and T_P were set at 10 and 5, respectively. The results of the experiment, as shown in Table 1, indicate that PADDLES outperforms other baseline methods, demonstrating its effectiveness in handling the WebVision dataset.

Table 1: Comparison with different methods on mini WebVision dataset. Top-1 and Top-5 test accuracies on the WebVision validation set and the ImageNet ILSVRC12 validation set are given. The results of the baseline methods are taken from [8].

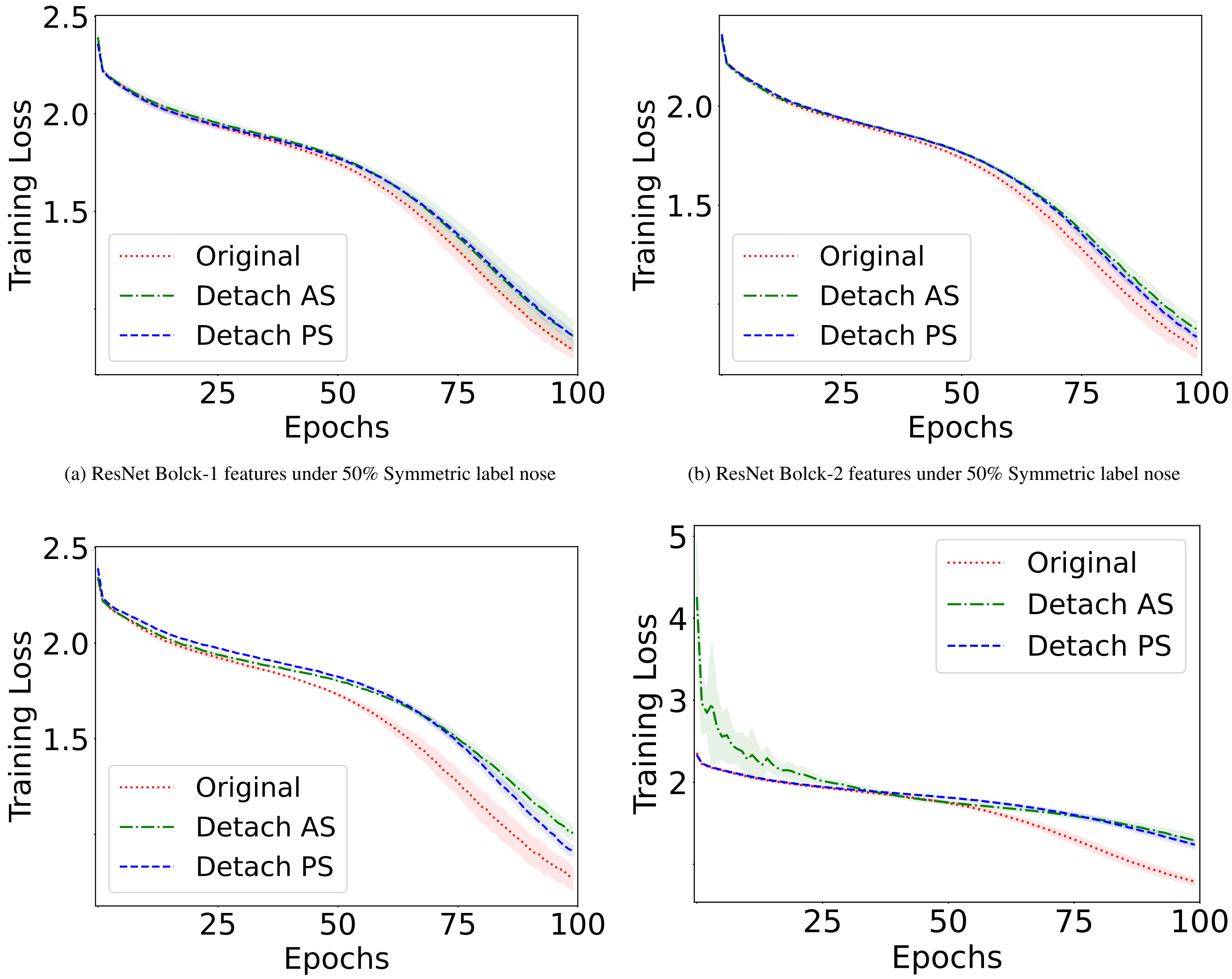
Method	WebVision		ILSVRC12	
	Top1	Top5	Top1	Top5
F-correction[14]	61.12	82.68	57.36	82.36
Decoupling[12]	62.54	84.74	58.26	82.26
D2L[11]	62.68	84.00	57.80	81.36
MentorNet[7]	63.00	81.40	57.80	79.92
Co-teaching[4]	63.58	85.20	61.48	84.70
Iterative-CV[3]	65.24	85.34	61.60	84.98
DivideMix[8]	77.32	91.64	75.20	90.84
PADDLES	77.64	92.08	75.48	91.20

3.2. Training Curves with Different Label Noises

More illustration is provided in Figure 5 regarding the impact of different kinds of label noises on deep models. Two additional kinds of label noises, the Pairflip [4] with a 45% noise rate and the Instance [19] with a 40% noise rate, were generated. It can be observed that the inflection point of AS's loss decline is earlier than that of PS components, indicating that the converging speed of CNN on AS is faster than PS. Furthermore, the curves of AS and PS become closer as the training epochs increase, indicating that the PS is more robust than AS with different label noises. Another difference between AS and PS is that the number of training steps to achieve optimal performance is not the same. Figures 5c and 5f demonstrate that AS achieves the best performance faster than PS. Both Figure 1 in our paper and Figure 5 in this material provide inspiration for decomposing the AS and PS from the input images and designing different stopping points to obtain a more robust deep network over previous ES models.

3.3. Confident Samples Quality

The quality of the extracted labels is a pivotal aspect of any machine-learning model, especially in the realm of image classification. In our study, we examined this quality us-



(a) ResNet Bolck-1 features under 50% Symmetric label nose

(b) ResNet Bolck-2 features under 50% Symmetric label nose

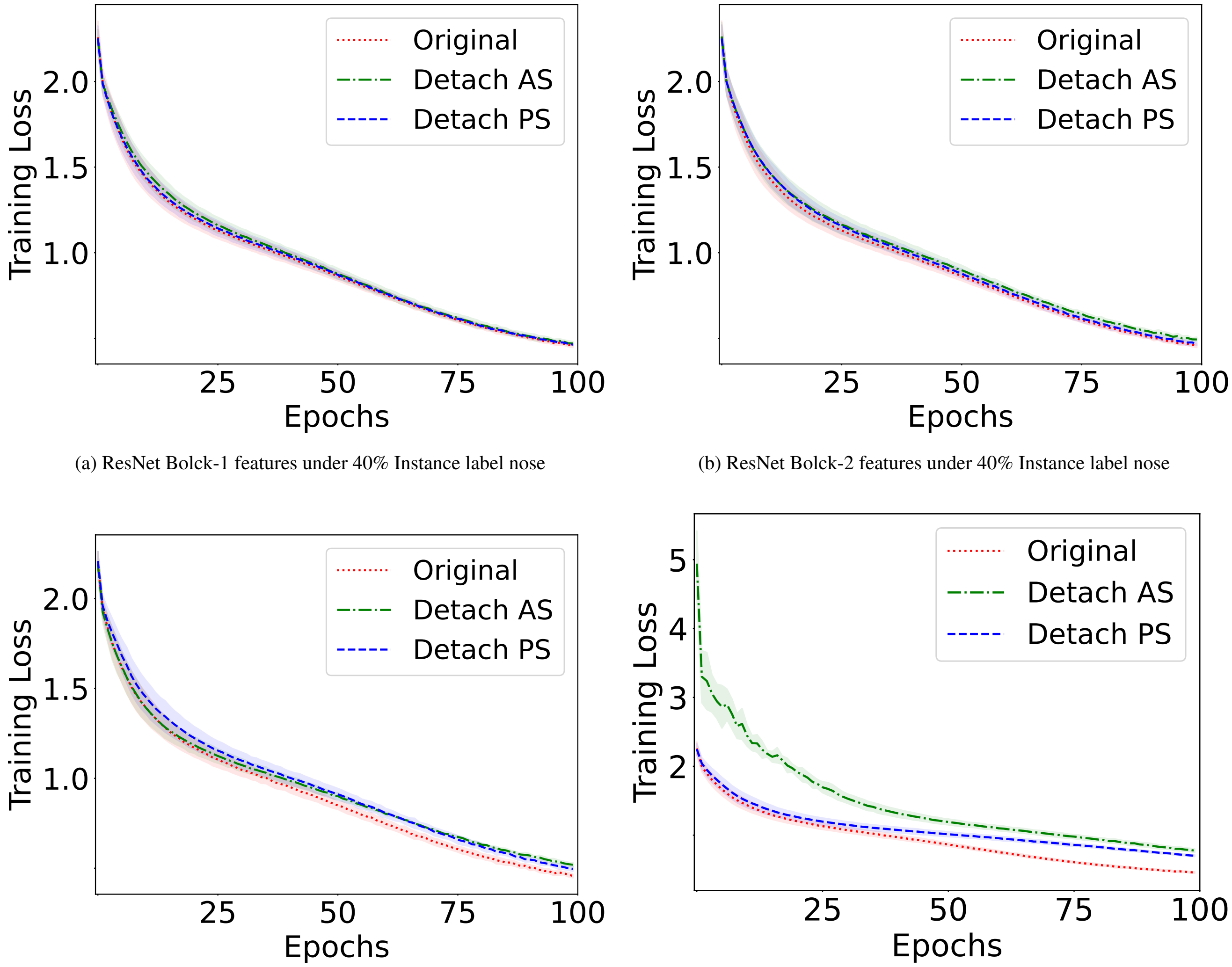
(c) ResNet Bolck-3 features under 50% Symmetric label nose

(d) ResNet Bolck-4 features under 50% Symmetric label nose

Figure 2: To evaluate the impact of Symmetric label noise on deep models with different frequency components extracted from different CNNs layers, we train a ResNet-18 model on CIFAR-10 using original image, amplitude spectrum (detach the gradients computing on phase spectrum), and phase spectrum (detach the gradients computing on phase spectrum) from different ResNet blocks. The X-axis illustrates the training epochs. Figure 2a presents the training losses of detach AS and PS components from the first ResNet block, indicated as "Detach AS" and "Detach PS" separately, and the "Original" represents train the ResNet-18 without any manipulation in the frequency domain. Figure 2b, Figure 2c, and Figure 2d show the corresponding training losses of the ResNet block 2, block 3 and block 4. The curves are based on five random runs.

ing the CIFAR-10 dataset, a well-established benchmark in the field. Our evaluation metrics, which are crucial for understanding model performance, included test accuracy, label recall, and label precision. Our methodology was deeply inspired by the approach detailed in PES [1].

Label recall, an essential metric, refers to the ratio of extracted confident samples with correct labels to the entire set of correctly labeled samples. On the other hand, label precision, equally crucial, quantifies the ratio of confidently extracted samples with correct labels to all confident samples. These metrics, when combined, provide a holistic view of the model's performance, ensuring both robustness and precision. For the experimental setup, we employed a state-of-the-art neural network architecture based on ResNet-18. We trained the networks 25 epochs. During this training phase, the model was exposed to a diverse range of label noise, simulating real-world scenarios and challenges. The point is located between the 3rd and 4th ResNet blocks. Further-



(a) ResNet Bolck-1 features under 40% Instance label nose

(b) ResNet Bolck-2 features under 40% Instance label nose

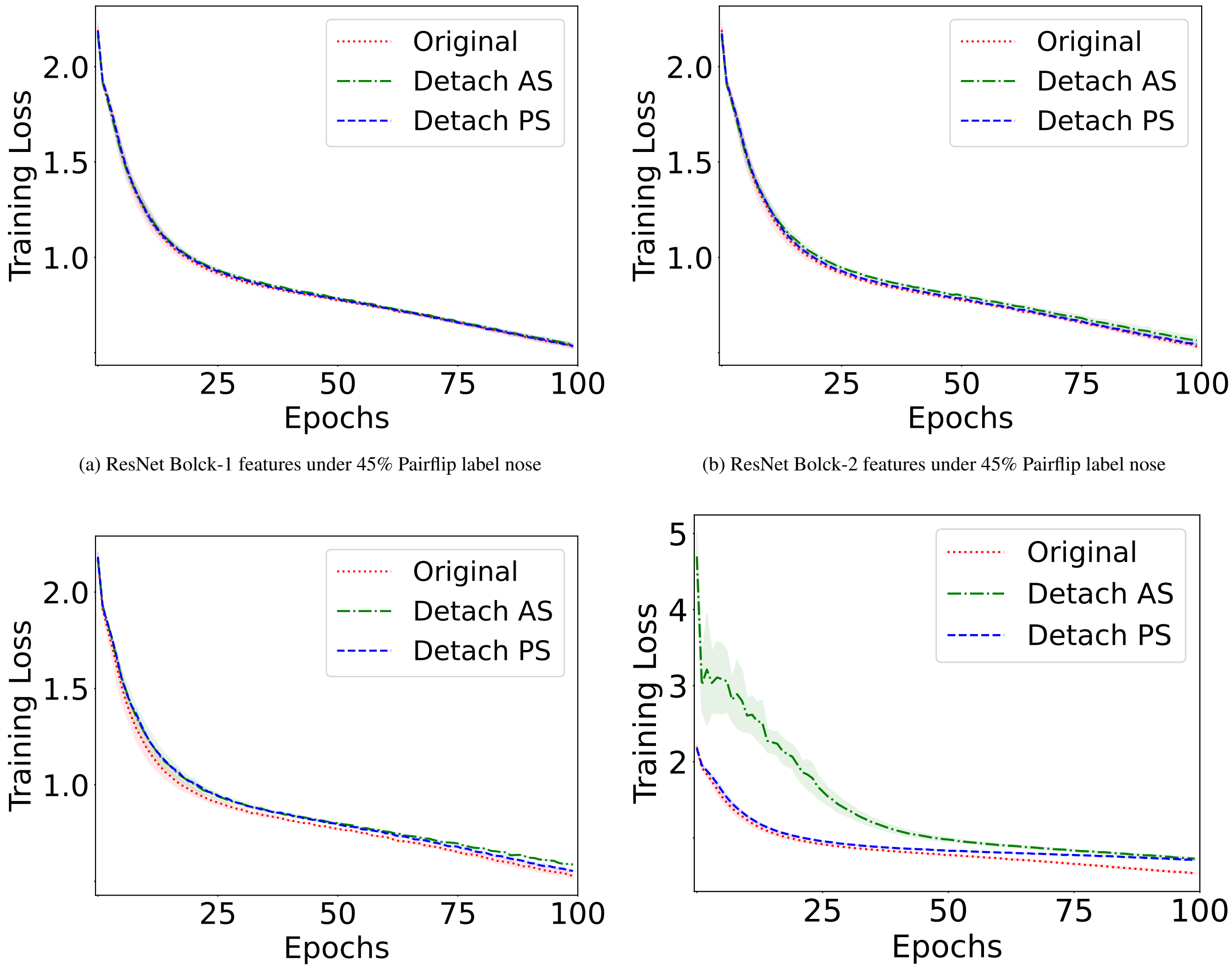
(c) ResNet Bolck-3 features under 40% Instance label nose

(d) ResNet Bolck-4 features under 40% Instance label nose

Figure 3: To evaluate the impact of Instance label noise on deep models with different frequency components extracted from different CNNs layers, we train a ResNet-18 model on CIFAR-10 using original image, amplitude spectrum (detach the gradients computing on phase spectrum), and phase spectrum (detach the gradients computing on phase spectrum) from different ResNet blocks. The X-axis illustrates the training epochs. Figure 3a presents the training losses of detach AS and PS components from the first ResNet block, indicated as "Detach AS" and "Detach PS" separately, and the "Original" represents train the ResNet-18 without any manipulation in the frequency domain. Figure 3b, Figure 3c, and Figure 3d show the corresponding training losses of the ResNet block 2, block 3 and block 4. The curves are based on five runs.

more, the stopping points of $\mathcal{AS}_{\mathcal{X}}$ and $\mathcal{PS}_{\mathcal{X}}$ were set to 23 and 25, respectively. The results are presented in Table 2.

Upon analyzing the data in Table 2, it becomes evident that our methods consistently outshine the conventional CE and PES methods. This superiority suggests that our techniques are adept at achieving higher accuracy and recall, while also maintaining a commendable level of precision. Such performance is indicative of our methods' capability to extract a larger volume of confident samples, a factor of paramount importance for semi-supervised learning. In the broader perspective, models boasting high recall values are inherently advantageous as they can amass a wealth of confident samples. This abundance benefits both supervised and semi-supervised training regimes, invariably leading to enhanced final classification outcomes. The empirical evidence from our experiments robustly supports this conclusion, highlighting the significance of our work.



(a) ResNet Bolck-1 features under 45% Pairflip label nose

(b) ResNet Bolck-2 features under 45% Pairflip label nose

(c) ResNet Bolck-3 features under 45% Pairflip label nose

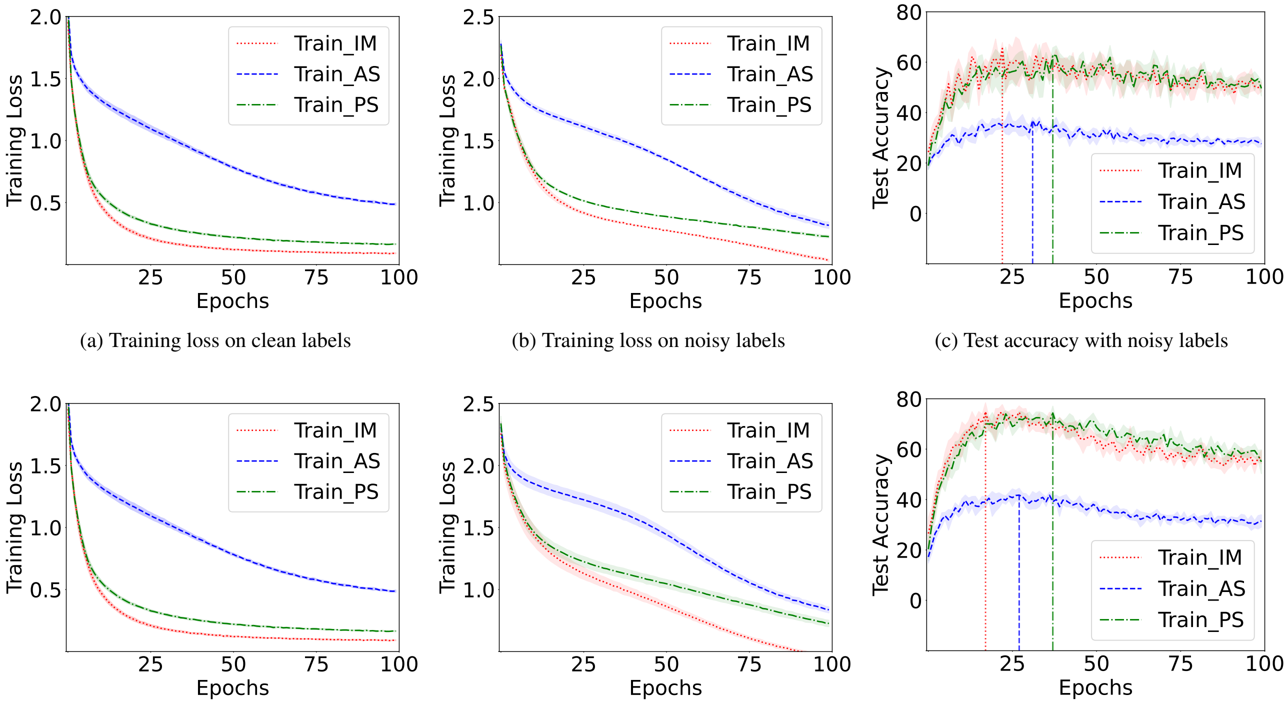
(d) ResNet Bolck-4 features under 45% Pairflip label nose

Figure 4: To evaluate the impact of Pairflip label noise on deep models with different frequency components extracted from different CNNs layers, we train a ResNet-18 model on CIFAR-10 using original image, amplitude spectrum (detach the gradients computing on phase spectrum), and phase spectrum (detach the gradients computing on phase spectrum) from different ResNet blocks. The X-axis illustrates the training epochs. Figure 4a presents the training losses of detach AS and PS components from the first ResNet block, indicated as "Detach AS" and "Detach PS" separately, and the "Original" represents train the ResNet-18 without any manipulation in the frequency domain. Figure 4b, Figure 4c, and Figure 4d show the corresponding training losses of the ResNet block 2, block 3 and block 4. The curves are based on five random runs.

4. Training Details

In this section, we give more implementation details about our experiments. We use three kinds of synthetic label noises for CIFAR-10 and CIFAR-100: symmetric class-dependent label noise [18] (Symmetric), pairflip class-dependent label noise [4] (Pairflip), and instance-dependent label noise [19] (Instance). We follow the implementation of ([4, 19, 1]) to generate these label noises with different levels, which can be found in PES.

Data preprocessing For learning with confident samples (Table 1 in the paper), we apply the random crop and random horizontal flip as data augmentations. We further add MixUp [20] data augmentation for semi-supervised settings (Table 2 in the paper). For CIFAR-N dataset (Table 3 in the paper), we use random crop, random horizontal, and a CIFAR-10 augmentation policy from [13]. The input im-



(a) Training loss on clean labels

(b) Training loss on noisy labels

(c) Test accuracy with noisy labels

(d) Training loss on clean labels

(e) Training loss on noisy labels

(f) Test accuracy with noisy labels

Figure 5: To evaluate the impact of label noise on deep models with different image components, we train a ResNet-18 model on CIFAR-10 using original images, amplitude spectrum, and phase spectrum under clean and noisy labels. The training losses on two kinds of labels (Figure 5a and Figure 5b 5e) and testing accuracy with the noisy labels (Figure 5c 5f) are given. The X-axis illustrates the training epochs. Figure 5b 5c are based on the 45% Pairflip label noises and Figure 5e 5f are based on the 40% Instance label noises. The curves are based on five random experiments, and the dotted vertical lines indicate the best performance steps of different image components.

Table 2: Analysis of the performance and the quality of the confident samples extracted from CIFAR-10. Mean and standard deviation over five runs are reported.

Metric	Method	Symmetric		Pairflip	Instance	
		20%	50%	45%	20%	40%
Test Accuracy	CE	82.55±2.46	70.76±1.24	60.62±5.59	84.41±0.90	74.73±2.65
	PADDLES_Base	**84.73±0.65**	**74.34±2.06**	**63.68±1.59**	**85.63±1.16**	**76.70±3.60**
	PES	85.87±1.59	75.87±1.33	62.40±2.34	86.58±0.45	77.07±1.18
	PADDLES	**86.98±0.56**	**76.62±1.66**	**64.39±1.79**	**86.79±0.78**	**78.44±2.17**
Label Recall	CE	88.51±2.26	75.18±1.00	67.84±5.06	90.37±1.01	82.15±3.17
	PADDLES_Base	**91.48±0.88**	**79.18±2.25**	**70.14±3.34**	**91.99±0.89**	**84.02±4.87**
	PES	92.67±1.43	81.03±1.83	71.06±2.27	93.24±0.60	**85.91±0.68**
	PADDLES	**93.29±1.26**	**82.10±2.12**	**74.28±5.45**	**93.90±1.02**	84.90±2.93
Label Precision	CE	98.81±0.15	94.65±0.19	72.53±5.26	**98.70±0.43**	**90.77±1.87**
	PADDLES_Base	**98.83±0.08**	**95.01±0.27**	**72.97±3.01**	98.52±0.26	89.83±2.73
	PES	**98.96±0.09**	**95.46±0.14**	72.99±2.27	**98.52±0.19**	**90.63±0.92**
	PADDLES	98.89±0.08	95.34±0.29	**73.38±5.28**	98.30±0.32	88.68±3.00

age size of CIFAR-like datasets is set as 32 × 32. For the Clothing-1M dataset (Table 4 in the paper), we first resize input images to the size of 256 × 256, then randomly crop the image as 224 × 224, and random horizontal flip last.

Hyper-parameters of PADDLES In learning with confident sample settings, we adopt ResNet-18 as the backbone for CIFAR-10 and ResNet-34 for CIFAR-100. We set the learning rate as 0.1, the weight decay as 10^{-4}, the batch size as 128, and the training epochs is 110. For PES training parameters, we use Adam optimizer, and set the PES learning rate is 10^{-4}, T_2, T_3 in [1] are 7 and 5 separately. Different types and levels of label noises result in different converge points of deep model on AS and PS. Therefore, we set different stopping points of T_A and T_P for different kinds and levels of label noises. For CIFAR-10, the T_A for 20%/40% Instance noise, 45% Pairflip noise, and 20%/50% Symmetric noise are [17, 20, 19, 18, 19]. The corresponding T_P are [13, 25, 16, 21, 20]. For CIFAR-100, the T_A for 20%/40% Instance noise, 45% Pairflip noise, and 20%/50% Symmetric noise are [20, 20, 19, 29, 20]. The corresponding T_P are [22, 22, 26, 11, 13]. The T_0 in Algorithm 2 is set as 0, and the training loss is the cross-entropy loss.

In semi-supervised learning, we adopt PreAct ResNet-18 as the backbone. The learning rate is 0.02 with a SGD optimizer, and we use cosine annealing learning rate scheduler to control the update of the learning rate. We set the weight decay as 5×10^{-4}, the batch size as 128, the training epochs as 500, and T_2 in [1] as 5. We train the semi-supervised models using MixMatch [2] loss with same parameters (λ_u, T, K) in [1]. Moreover, we set T_0 in Algorithm 2 as 0.

For CIFAR-N datasets, we use the ResNet-34 architecture. We set the learning rate as 0.02, the batch size as 128, the weight decay as 5×10^{-4}, the training epochs as 300, the T_2 in PES as 5. We also employ the MixMatch loss to train the semi-supervised model with MixMatch parameter λ_u as 5 and 75 for CIFAR-10N and CIFAR-100N, respectively. We set T_0 in Algorithm 2 as 1, and we do observe further performance improvement with a bigger T_0 like 5 in our CIFAR-N settings.

For Clothing-1M dataset, we employ the ResNet-50 as the backbone, which is pre-trained on the ImageNet. We set the batch size as 64, and the training epochs as 150. During training, we adopt the SGD optimizer with the learning rate as 4.5×10^{-3}, the weight decay as 0.001, and the momentum as 0.9. We also use a three phase OneCycle [16] scheduler to dynamic adjust the learning rate with the max learning rate as 8.55×10^{-3}. The corresponding PES learning rate is set as 5×10^{-6} and the T_2 is 7. Moreover, the training loss is the weighted cross-entropy loss, and T_0 in Algorithm 2 is as 0. More details will be found in our scheduled released codes.

References

[1] Yingbin Bai, Erkun Yang, Bo Han, Yanhua Yang, Jiatong Li, Yinian Mao, Gang Niu, and Tongliang Liu. Understanding and improving early stopping for learning with noisy labels. *NeurIPS*, 34:24392–24403, 2021.

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *NeurIPS*, 2019.

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

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

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, June 2016.

[6] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, pages 646–661. Springer, 2016.

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

[8] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, 2020.

[9] Wen et al. Li. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

[10] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *NeurIPS*, 2020.

[11] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pages 3355–3364. PMLR, 2018.

[12] Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from" how to update". *NeurIPS*, 30, 2017.

[13] Kento Nishi, Yi Ding, Alex Rich, and Tobias Hollerer. Augmentation strategies for learning with noisy labels. In *CVPR*, pages 8022–8031, 2021.

[14] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pages 1944–1952, 2017.

[15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[16] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019.

[17] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[18] Brendan Van Rooyen, Aditya Menon, and Robert C Williamson. Learning with symmetric label noise: The importance of being unhinged. *NeurIPS*, 28, 2015.

[19] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *NeurIPS*, 33:7597–7610, 2020.

[20] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.