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

Anonymous authors
Paper under double-blind review

Abstract

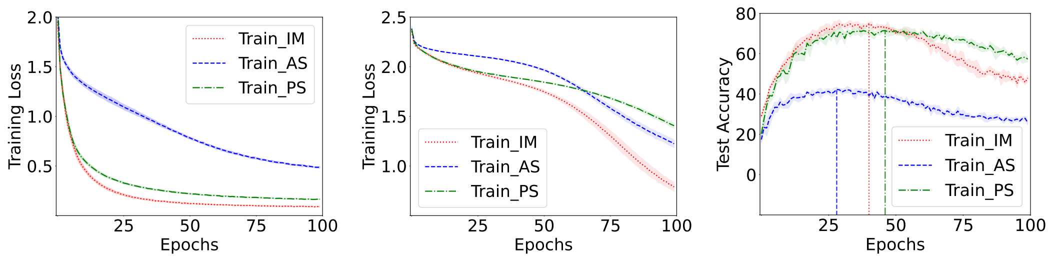
Deep Neural Networks (DNNs) have demonstrated superiority in learning various patterns. However, DNNs are sensitive to label noises and would easily overfit noisy labels during training. The early stopping strategy averts updating DNNs during the early training phase and is widely employed as an effective method when learning with noisy labels. Motivated by biological findings that the amplitude spectrum (AS) and phase spectrum (PS) in the frequency domain play different roles in the animal's vision system, we observe that PS, which captures more semantic information, is more resistant to label noise than AS. Performing the early stopping on AS and PS at the same time is therefore undesirable. In contrast, we propose early stops at different times for AS and PS. In order to achieve this, we disentangle the features of some layer(s) into AS and PS using Discrete Fourier Transform (DFT) during training. The AS and PS will be detached at different training stages from the gradient computational graph. The features are then restored via inverse DFT (iDFT) for the next layer. We term the proposed method Phase-AmplituDe DisentangLed Early Stopping (PADDLES). Simple yet effective, PADDLES outperforms other early stopping methods and obtains state-of-the-art performance on both synthetic and real-world label-noise datasets.

1 Introduction

Learning from noisy labels (LNL) (Angluin & Laird (1988)) has revived as a hot research topic with the development of deep learning (Reed et al. (2015); Goldberger & Ben-Reuven (2017); Malach & Shalev-Shwartz (2017); Patrini et al. (2017); Thekumparampil et al. (2018); Zhang & Sabuncu (2018); Kremer et al. (2018); Han et al. (2018); Ren et al. (2018); Yu et al. (2018); Jiang et al. (2018); Xu et al. (2019); Yu et al. (2019); Liu & Guo (2020); Li et al. (2020b;a); Hu et al. (2020); Lyu & Tsang (2020); Yao et al. (2020); Xia et al. (2020b); Yao et al. (2021); Cheng et al. (2021); Zhu et al. (2021); Ghazi et al. (2021); Paul et al. (2021); Yang et al. (2022); Wu et al. (2022); Liu et al. (2022b); Xia et al. (2022); Wei et al. (2022)). As noisy labels widely exist in real-world datasets (Welinder & Perona (2010); Vijayanarasimhan & Grauman (2014); Xiao et al. (2015); Sun et al. (2021)), a trustworthy AI system should be robust towards inaccurate labels or mislabels.

The memorization effect of deep models that DNNs learn the clean patterns first and then memorize (overfit) the noise patterns (Arpit et al. (2017)), inspired many breakthroughs (Han et al. (2018); Wang et al. (2018); Li et al. (2020a;b); Xia et al. (2020a); Liu et al. (2020; 2022a)) in LNL. A representative training strategy is early stopping (ES), which stops the gradient-based optimization at a particular early training step. Due to its effectiveness, ES is widely applied in current LNL models and has achieved promising performance (Tanaka et al. (2018); Li et al. (2020a); Nguyen et al. (2020); Bai et al. (2021); Liu et al. (2022a)).

The frequency and spatial domains are alternative codes for depicting signal data such as images and text (Oppenheim et al. (1997); Szeliski (2010)). Different frequency components contain different information. (Castleman (1996)) indicated that the amplitude spectrum (AS) prescribes how much of each sinusoidal component is present, while the phase spectrum (PS) stipulates the location of each sinusoidal component residing in the image. Biological justification and psychological patterns testing (Simoncelli & Schwartz (1999); Guo et al. (2008)) demonstrated that the response of cells



(a) Cleanly labeled examples (b) Wrongly labeled examples (c) Test accuracy with noisy labels

Figure 1: Results of training a ResNet-18 model on CIFAR-10 using original images, amplitude spectrum, and phase spectrum ("Train_IM", "Train_AS", and "Train_PS" in the Figure) on cleanly and noisily labeled subsets. The curves are averaged across five random runs. The dotted vertical lines indicate the best performance steps of different image components. The converging speed of the deep model trained on AS and PS differs, especially on wrongly labeled examples. Approaching the end of the training, when the wrong labels begin to be memorized, the model accelerates fitting to AS, resulting in an intersection on the training curves of AS and PS, shown in Figure 1(b). Hence, PS is more resistant to label noises than AS.

in the primary visual cortex (V1) is closely related to the local AS for specific image patterns (frequency and orientation). That is, the AS component usually represents the intensity of the patterns in the image. On the other hand, previous qualitative and quantitative studies (Castleman (1996); Guo et al. (2008)) indicated that the PS is the key to locating salient object areas and holds visible structured information for vision recognition (Oppenheim & Lim (1981); Ghiglia & Pritt (1998); Li et al. (2015)), thus containing more semantic information than the AS.

Current deep models, such as Convolutional Neural Networks (CNNs), profit from human unperceivable high-frequency components in images (Ilyas et al. (2019); Wang et al. (2020)). However, without adequate regulations, CNNs perceive more AS than PS (Chen et al. (2021)), which is inconsistent with the human vision system of focusing on semantic parts (Oppenheim & Lim (1981); Guo et al. (2008); Li et al. (2015)). The counterintuitive behavior of CNNs and the properties of different frequency components of images invoke an interesting question: How can we train a robust model using the frequency components when our supervision is the noisy-label data?

Directly adapting the ES strategy to stop the optimization of CNNs on all image components simultaneously will ignore their different sensitivity towards noisy labels, which may lead to sub-optimal solutions. However, solely depending on training with one component will lose the complementary information from other components, resulting in overall performance degradation. In Figure 1, we investigate the impact of label noise on deep models trained with different image components. We generate symmetric label noise (Van Rooyen et al. (2015); Han et al. (2018)) with a 50% noise rate. As shown in Figures 1(a) and 1(b), the convergence speed of CNNs on AS and PS is different. When CNNs start to overfit the noisy labels, they fit AS much faster than PS (Figure 1(b)), resulting in test performance degradation. Meanwhile, the learning speed on PS is slower than AS as well as the raw images, which indicates that PS maybe more robust than AS or raw inputs. Note that the model trained with only AS or PS performs worse than the one trained with the original images (Figure 1(c)). This is fair as either AS or PS could miss some information of the original image data. Therefore, how to utilize AS and PS separately but also prevent information loss is challenging.

To tackle this challenge, we propose to disentangle the deep image features into AS and PS at different training steps by Discrete Fourier Transform (DFT). We first detach the AS component from the gradient computational graph to stop its involvement in the model update, which can alleviate the potential negative effects of AS in the later training stage. With AS being detached, we continue train the deep model with PS components which are more robust to the noisy labels. We eventually stop optimization on the PS component as well after few training epochs. Notice that the detached components will regenerate the deep features in the spatial domain through inverse DFT (iDFT). This is efficient as there is no modification to the original architecture. Moreover, complete information is used for training. We call the proposed method as Phase-AmplituDe DisentangLed Early Stopping (PADDLES). To the best of our knowledge, PADDLES is the first method to consider features learned with noisy labels in the frequency domain and thus is orthogonal to existing methods that mainly focus on the spatial domain. Our contributions are summarized as follows:

- We study learning with noise labels from the frequency domain and find that the phase spectrum is more resistant to label noise than the amplitude spectrum.
- We propose to early stop training at different stages for amplitudes and phase spectrums. We show that this can utilize the robustness of the phase spectrum without losing information on phase during model training.
- Extensive experiments on benchmark datasets such as CIFAR-10, CIFAR-100, CIFAR-10N, CIFAR-100N, Clothing-1M, and NEWS datasets validate the effectiveness of the proposed method.

The rest of the paper is organized as follows. In Section 2, we introduce the proposed PADDLES method. In Section 3, we give empirical evaluations of our method, followed by Conclusions. Due to the page limit, we review the related works in Appendix A, and more details and additional experiments in Appendix B and C.

2 METHODOLOGY

In this section, we present the proposed Phase-AmplituDe DisentangLed Early Stopping (PADDLES). We first introduce the problem definition, followed by the detailed learning methods.

2.1 PROBLEM DEFINITION

In the learning with noisy labels, the real training data distribution can be defined as $\mathcal{D} = \{(x, y) \,|x \in \mathcal{X}, y \in \{1, \ldots, K\}\}$, where $\mathcal{X}$ is the sample space, and $\{1, \ldots, K\}$ denotes the label space with K classes. However, the actual distribution of the label space is usually inaccessible since the data collection and dataset construction will inevitably import label errors. We can only use the accessible noisy dataset $\widehat{\mathcal{D}} = \{(x, \widehat{y}) \,|x \in \mathcal{X}, \widehat{y} \in \{1, \ldots, K\}\}$ to train the model, where $\widehat{y}$ denotes the corrupted noisy labels. The goal of our algorithm is to learn a robust deep classifier from the noisy data that can perform accurately on the query samples.

2.2 PHASE-AMPLITUDE DISENTANGLED EARLY STOPPING

Training a deep model with a noisy dataset $\widehat{\mathcal{D}}$ is challenging as the model will fit the clean labels first and then overfit the noisy labels, as shown in Figure 1. This memorization effect motivates previous methods to adapt the early stopping to cease the optimization of deep models at a specific step. Namely, the early stopping method aims to choose a suitable step tp in training a deep model f_Θ. The training process is to learn an optimal Θ^*:

$$\Theta^* = \underset{\Theta=\{\Theta_T, \Theta_{T-1}, \ldots, \Theta_0\}}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(\hat{y}_i, f_{\Theta_T} \circ f_{\Theta_{T-1}} \circ \cdots \circ f_{\Theta_0}\left(x_i\right)\right), \tag{1}$$

where $\Theta = \{\Theta_T, \Theta_{T-1}, \ldots, \Theta_0\}$ denotes the parameter(s) of the deep model, and $\circ$ denotes the operator of the function composition. The deep model $f_\Theta(\cdot)$ is rewritten as $f_{\Theta_T} \circ f_{\Theta_{T-1}} \circ \cdots \circ f_{\Theta_0}(\cdot)$ since the deep neural networks can be viewed as a stack of non-linear functions. $x_i, \hat{y}_i$ represent the ith sample and its label, and $\mathcal{L}$ is the cross-entropy training loss.

To obtain Θ^*, previous works (Liu et al. (2020); Xia et al. (2020a); Bai et al. (2021)) developed various optimization policies from the perspective of robust loss function design (Liu et al. (2020)), gradient regulation (Xia et al. (2020a)), and progressive architecture selection (Bai et al. (2021)). These methods focus on the spatial domain, and treats the input data (images) as a whole. However, as discussed in the Introduction section, different image components play different roles in the vision system. It is unoptimized to stop the model optimization on these components simultaneously.

For this reason, we propose to investigate the early stopping on the input data components and select different stop points for different parts. It is natural to consider the frequency domain due to its equivalent representation of input data (Castleman (1996); Oppenheim et al. (1997)) on the

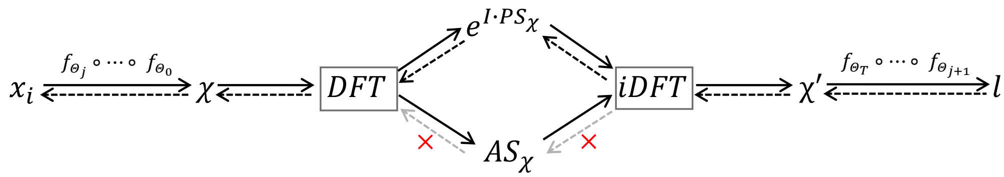


Figure 2: The illustration of the proposed PADDLES strategy stops the amplitude spectrum's involvement in model training. "⟶" denotes the forward propagation, while the "⇠" represents the backward propagation. Using Equations 2 3 and 4, we form a computational chain to disentangle the frequency domain representation, and then we can stop the backward propagation of the target component. In this way, we can control the model's optimization with each component and choose different stopping points.

spatial domain and the vision properties (Bian & Zhang (2008); Li et al. (2015)) of amplitude and phase spectrum, as discussed previously. Specifically, for an input sample x_i, the deep feature after jth operation in f_Θ can be represented as $\chi = f_{\Theta_j} \circ \cdots \circ f_{\Theta_0}(x_i)$, and its frequency domain representation $\mathcal{F}_\chi$ can be computed using DFT:

$$\mathcal{F}_\chi(u) = \sum_{p=0}^{M-1} \chi_p e^{\frac{-I \cdot 2\pi}{M} pu}, \tag{2}$$

which can be denoted as $\mathcal{F}_\chi = DFT(\chi)$. u represents a specific frequency, M is the number of sampled points, I is the imaginary unit, and χ_p denotes the value at the position p of χ. We consider one dimension here for simplicity, and the higher-dimensional DFT corresponds to successive Fourier transforms along each dimension in sequence. Notice that the $\mathcal{F}_\chi(u)$ is a complex-valued variable, its real part can be denoted as $Real_{\mathcal{F}_\chi}$, and the imaginary part is $Imag_{\mathcal{F}_\chi}$. We then disentangle the phase and amplitude components using the following rules:

$$\begin{aligned} \mathcal{PS}_\chi(u) &= \arctan(\frac{Imag_{\mathcal{F}_\chi}(u)}{Real_{\mathcal{F}_\chi}(u)}), \\ \mathcal{AS}_\chi(u) &= |\mathcal{F}_\chi(u)|, \end{aligned} \tag{3}$$

where $\mathcal{PS}_\chi$ represents the phase spectrum, $\mathcal{AS}_\chi$ represents the amplitude spectrum, $\arctan(\cdot)$ is the inverse trigonometric function, and $|\cdot|$ computes the absolute value. Using Equations 2 and 3, the deep features are decomposed into amplitude and phase components during the model training. Afterward, we restore the deep feature using iDFT:

$$\chi'_p = \frac{1}{M} \sum_{u=0}^{M-1} (e^{I \cdot PS_\chi(u)} \odot AS_\chi(u)) e^{\frac{I \cdot 2\pi}{M} pu}, \tag{4}$$

which can be represented with $\chi' = iDFT(e^{I \cdot PS_\chi} \odot AS_\chi)$. Notice that $\chi' = \chi$, $\odot$ indicates the element-wise multiplication operation.

Through Equation 2, 3, and 4, we construct a computation chain disentangling the phase spectrum $\mathcal{PS}_\chi$ and the amplitude spectrum $\mathcal{AS}_\chi$ from the original feature χ during the end-to-end model training. Therefore, we can control the deep model's optimization with each component. Specifically, the end-to-end training of the deep model f_Θ consists of the forward and the backward propagations, the forward propagation (right arrows in Figure 2) will generate the intermediate values $(\chi, \mathcal{PS}_\chi, \mathcal{AS}_\chi, \chi')$ with the input x_i, and the backward propagation (left arrows in Figure 2) will track the gradients for each intermediate value and model parameter. Finally, the model is updated using the gradient descent with the tracked gradients. For the backward propagation of f_Θ, we need to compute the partial derivatives of loss function $\mathcal{L}$ with respect to $\mathcal{PS}_\chi$ ($\frac{\partial \mathcal{L}}{\partial \mathcal{PS}_\chi}$) and $\mathcal{AS}_\chi$ ($\frac{\partial \mathcal{L}}{\partial AS_\chi}$) [1]. Stopping computing these derivatives can detach the phase-related gradient or amplitude-related gradient nodes from the gradient computational graph and thus control the model optimization on each frequency component, as illustrated in Figure 2.

[1] Thanks to the automatic differentiation engine of deep learning frameworks, *e.g.,* PyTorch and TensorFlow, it is convenient to obtain the derivatives and gradient for each variable. Therefore, we omit the derivatives computation of PS and AS here.

Algorithm 1: PADDLES

Input : A noisy set $\hat{\mathcal{D}}$, Deep Model $f_{\Theta=\{\Theta_T,\Theta_{T-1},\ldots,\Theta_0\}}$, Disentangle point j, AS_χ training epoch T_A, PS_χ training epoch T_P, Additional epoch T_0, Epochs for remaining part: $T_{j+1},\ldots,T_T$, Epoch T_r for learning with confident samples (semi).

1 **for** $i=1$ **to** T_A **do**
2   Update network parameter Θ using Equation 1;
3 **end**
4 **for** $i=1$ **to** T_P **do**
5   Extract χ at f_{Θ_j}, disentangle χ into $\mathcal{AS}_\chi$ and $\mathcal{PS}_\chi$ using Equation 2 and 3;
6   Detach gradient computation of $\mathcal{AS}_\chi$ in Equation 3;
7   Restore deep feature χ' using Equation 4;
8   Update network parameter Θ using Equation 1;
9 **end**
10 **for** $i=1$ **to** T_0 **do**
11   Extract χ at f_{Θ_j}, disentangle χ into $\mathcal{AS}_\chi$ and $\mathcal{PS}_\chi$ using Equation 2 and 3;
12   Detach gradient computation of $\mathcal{PS}_\chi$ in Equation 3;
13   Restore deep feature χ' using Equation 4;
14   Update network parameter Θ using Equation 1;
15 **end**
16 Hook $\mathcal{AS}_\chi$ and $\mathcal{PS}_\chi$ to the gradient computation graph during backpropagation;
17 **for** $l=j+1$ **to** T **do**
18   Freeze $\{\Theta_0,\ldots,\Theta_j\}$ and re-initialize $\{\Theta_{j+1},\ldots,\Theta_T\}$;
19   **for** $i=1$ **to** T_l **do**
20   Update network parameter $\{\Theta_{j+1},\ldots,\Theta_T\}$ using Equation 5;
21   **end**
22 **end**
23 Unfreeze f_Θ;
24 **for** $i=1$ **to** T_r **do**
25   Extract confident sample set $\mathcal{D}_{lb}$ and unlabeled set $\mathcal{D}_{ub}$ using Equation 6 and 7 ;
26   Update f_Θ using MixMatch loss on $\mathcal{D}_{lb}$ and $\mathcal{D}_{ub}$;
27 **end**

Output: The optimized model f_{Θ^*}.

2.3 Practical Implementation

The proposed PADDLES is summarized in Algorthm 1. In this section, we introduce the structure of our model and the corresponding learning settings.

Model Structure To reduce the difficulty of implementation and further improve the robustness of PADDLES, we incorporate progressive early stopping (PES) (Bai et al. (2021)) in our model training. Therefore, we need to add a copy of the PES optimization strategy.

After finishing the amplitude and phase spectrum training (Step 15 in Algorithm 1). The parameter parts $\{\Theta_0^*,\ldots,\Theta_j^*\}$ are well-optimized. PES model will continue update the remaining parts $\{\Theta_{j+1},\ldots,\Theta_T\}$ with previous parameters fixed. Training process will perform T_l steps using the following objective:

$$\min_{\{\Theta_l,\ldots,\Theta_T\}} \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}\left(\hat{y}_i, f_{\Theta_T}\circ\cdots\circ f_{\Theta_l}\circ f_{\Theta_{l-1}^*}\circ\cdots\circ f_{\Theta_0^*}(x_i)\right), l=j+1,\cdots,T. \tag{5}$$

After PES optimization, the final model $f_{\Theta^*=\{\Theta_0^*,\cdots,\Theta_T^*\}}$ is obtained.

Learning Settings Following (Li et al. (2020a); Bai et al. (2021)), we adopt PADDLES as a confident sample selector to boost noisy label learning with supervised and semi-supervised learning settings. The confident sample set $\mathcal{D}_{lb}$ is defined as

$$\begin{aligned}\mathcal{D}_{lb} &= \{(x_i, \hat{y}_i) | \hat{y}_i = \bar{y}_i, i = 1, \cdots, N\}, \\ \bar{y}_i &= \underset{\tau \in \{1, \cdots, K\}}{\arg\max} \frac{1}{2}[f^{\tau}_{\Theta^*}(\mathrm{A}(x_i)) + f^{\tau}_{\Theta^*}(\mathrm{A}'(x_i))],\end{aligned} \tag{6}$$

where A and A$'$ are data augmentation operators randomly sampled from the same augmentation set, $f^{\tau}_{\Theta^*}(x_i)$ indicates the classification probability of x_i belonging to class τ. For the supervised learning with confident samples, we adopt the weighted classification loss (Equitation (6) in Bai et al. (2021)).

For the semi-supervised setting, besides the confident label set $\mathcal{D}_{lb}$, the additional unlabeled set D_{ub} is defined as

$$\begin{aligned}\mathcal{D}_{ub} &= \{x_i | \hat{y}_i \neq \bar{y}_i, i = 1, \cdots, N\}, \\ \bar{y}_i &= \underset{\tau \in \{1, \cdots, K\}}{\arg\max} \frac{1}{2}[f^{\tau}_{\Theta^*}(\mathrm{A}(x_i)) + f^{\tau}_{\Theta^*}(\mathrm{A}'(x_i))].\end{aligned} \tag{7}$$

We adopt the MixMatch (Berthelot et al. (2019)) loss to train the semi-supervised learning task as previous works (Li et al. (2020a); Bai et al. (2021)).

3 Experiments

3.1 Experimental Setup

Datasets We demonstrate the effectiveness of our PADDLES on the two manually corrupted datasets: CIFAR-10 and CIFAR-100 (Krizhevsky et al. (2009)), and two real-world noisy datasets: CIFAR-N (Wei et al. (2022)) and Clothing-1M (Xiao et al. (2015)). Both CIFAR-10 and CIFAR-100 contain 50,000 training samples and 10,000 testing samples with the size of 32×32 for each image sample. CIFAR-10 has 10 classes, while CIFAR-100 contains 100 classes. The original labels of these two datasets are clean, and we generate three types of noisy labels, *i.e.,* symmetric, pairflip, and instance-dependent label noise, according to (Han et al. (2018); Liu et al. (2020); Xia et al. (2020a; 2019); Bai et al. (2021)). CIFAR-N consists of CIFAR-10N and CIFAR-100N, a re-annotation of CIFAR-10 and CIFAR-100 with real human annotators. Specifically, CIFAR-10N has five types of labels: *Random 1, Random 2, Random 3, Aggregate, and Worst*, which are derived from three submitted label sets. CIFAR-100N contains a single human annotated label set named *Noisy Fine*. Clothing-1M has 1,000,000 clothing images in 14 classes clawed from online shopping web sits. The labels of Clothing-1M are generated according to the context on the shopping web page, resulting in lots of mislabelled samples. This dataset also provides 14,313 and 10,526 images with clean labels for validation and testing. Besides the image datasets, we also validate our method on a text classificaiton dataset, NEWS (Joachims (1997); Yu et al. (2019)). Due to the page limit, the experimental analysis on NEWS can be found in *Appendix C.3*.

Comparison Methods We compare the proposed PADDLES with the following approaches: 1) Cross Entropy (CE) and MixUp as two baselines, which training deep models with cross-entropy loss and mixup (Zhang et al. (2018)) strategy separately. 2) Classic LNL methods: Co-teaching (Han et al. (2018)), Forward-T (Patrini et al. (2017)), JointOptim (Tanaka et al. (2018)), T-revision (Xia et al. (2019)), M-correction (Arazo et al. (2019)) and DMI (Xu et al. (2019)). 3) State-of-the art LNL methods: DivideMix (Li et al. (2020a)), CDR (Xia et al. (2020a)), ELR (Liu et al. (2020)), PES (Bai et al. (2021)), CORES (Cheng et al. (2021)) and SOP (Liu et al. (2022b)).

Network Structures and Hyperparameters We implement our method with PyTorch. The compared methods are implemented or re-implemented according to their original papers and open-source codes. We chose the same hyperparameters as their papers presented. We set network structures and hyperparameters for PADDLES on each noisy-label dataset as follows.

For the supervised learning setting, we follow (Xia et al. (2019); Bai et al. (2021)) to use ResNet-18 and ResNet-34 architectures for CIFAR-10 and CIFAR-100, respectively. The disentangle point j is between the 3rd and 4th ResNet blocks. The initial learning rate is 0.1 and decayed with a factor of 10 at the 100th epoch, the weight decay is 10^{-4}, and we train the networks 110 epochs. We list the details in the *Appendix B* including stopping points of $\mathcal{AS}_{\chi}$, $\mathcal{PS}_{\chi}$ and the related PES parameters.

Table 1: Comparison with different methods under supervised learning of confident samples on CIFAR-10 and CIFAR-100. The results of the baseline methods are taken from Bai et al. (2021). The best results are in bold. The mean and standard deviation computed over five runs are given.

Dataset	Method	Symmetric		Pairflip	Instance	
		20%	50%	45%	20%	40%
CIFAR-10	CE	84.00±0.66	75.51±1.24	63.34±6.03	85.10±0.68	77.00±2.17
	Co-teaching	87.16±0.11	72.80±0.45	70.11±1.16	86.54±0.11	80.98±0.39
	Forward-T	85.63±0.52	77.92±0.66	60.15±1.97	85.29±0.38	74.72±3.24
	JointOptim	89.70±0.11	85.00±0.17	82.63±1.38	89.69±0.42	82.62±0.57
	T-revision	89.63±0.13	83.40±0.65	77.06±6.47	90.46±0.13	85.37±3.36
	DMI	88.18±0.36	78.28±0.48	57.60±14.56	89.14±0.36	84.78±1.97
	CDR	89.72±0.38	82.64±0.89	73.67±0.54	90.41±0.34	83.07±1.33
	PES	92.38±0.40	87.45±0.35	88.43±1.08	92.69±0.44	89.73±0.51
	PADDLES	**92.43±0.18**	**87.94±0.22**	**89.32±0.21**	**92.76±0.30**	**89.87±0.51**
CIFAR-100	CE	51.43±0.58	37.69±3.45	34.10±2.04	52.19±1.42	42.26±1.29
	Co-teaching	59.28±0.47	41.37±0.08	33.22±0.48	57.24±0.69	45.69±0.99
	Forward-T	57.75±0.37	44.66±1.01	27.88±0.80	58.76±0.66	44.50±0.72
	JointOptim	64.55±0.38	50.22±0.41	42.61±0.61	65.15±0.31	55.57±0.41
	T-revision	65.40±1.07	50.24±1.45	41.10±1.95	60.71±0.73	51.54±0.91
	DMI	58.73±0.70	44.25±1.14	26.90±0.45	58.05±0.20	47.36±0.68
	CDR	66.52±0.24	55.30±0.96	43.87±1.35	67.33±0.67	55.94±0.56
	PES	68.89±0.45	58.90±2.72	57.18±1.44	70.49±0.79	65.68±1.41
	PADDLES	**69.19±0.88**	**59.78±3.15**	**58.68±1.28**	**70.88±0.55**	**66.11±1.19**

For the semi-supervised learning setting, we follow (Li et al. (2020a); Bai et al. (2021)) to use PreAct ResNet-18 for CIFAR-10 and CIFAR-100, and use ResNet-34 for CIFAR-N. For Clothing-1M, we adopt the ResNet-50 pretrained on the ImageNet. The disentangle point j is set between the 3rd and 4th ResNet blocks. We train the model 500 epochs using cosine annealing strategy for CIFAR-like datasets, and the initial learning rate is 0.02, with a weight decay of 5×10^{-4}, stopping points of $\mathcal{AS}_\chi$, $\mathcal{PS}_\chi$ are set as 30 ($T_A = 30$) and 35 ($T_P = 5$) separately. For Clothing-1M, we train the model with 150 epochs using OneCycleLR strategy (Smith & Topin (2019)) and set the learning rate to 4.5×10^{-3} with a weight decay of 0.001, stopping points of $\mathcal{AS}_\chi$, $\mathcal{PS}_\chi$ are set as 10 ($T_A = 10$) and 29 ($T_P = 19$), respectively. More details can be found in the *Appledix B*.

3.2 Classification Performance on Noisy Datasets

Results on Synthetic Datasets We evaluate PADDLES on CIFAR-10 and CIFAR-100 with different levels and types of label noise under supervised learning, as shown in Table 1. Under the same architectures, PADDLES consistently outperforms the other methods across different noisy types and noisy levels, which demonstrates the effectiveness of PADDLES.

In Table 2, we compare PADDLES with state-of-the-art semi-supervised LNL methods. PADDLES achieves a significant performance improvement of around 10% to 40% over the baseline methods such as CE and MixUp. Moreover, PADDLES beats the state-of-the-art LNL methods like ELR+ and PES on all settings. Specifically, with 80% Symmetric label noise on CIFAR-100, the classification accuracies are 62.9% vs. 61.6% PES (Bai et al. (2021)), indicating the superiority of PADDLES in using unlabelled data to boost classification performance.

Results on Real-world Datasets We compare the classification performance of different methods in Table 3. All the compared methods adopt a pre-trained ResNet-50 backbone on the ImageNet. Since PADDLES is equipped with a more nuanced optimization strategy from perspectives of frequency domain and progressive model construction, it achieves state-of-the-art performance.

Furthermore, we test our PADDLES model on a more challenging real-world noise-label dataset, as summarized in Table 4. CIFAR-N consists of CIFAR-10N and CIFAR-100N with six types of noisy labels annotated with human observers. We can observe a performance gain of PADDLES over comparing methods on five types of labels except for CIFAR-10N' Aggregate. PADDLES achieves comparable performance towards SOP+ on CIFAR-10N's Aggregate labels.

Table 2: Comparison with different methods under semi-supervised learning of confident samples on CIFAR-10 and CIFAR-100. The results of the baseline methods are taken from Bai et al. (2021). The best results are in bold. The mean and standard deviation computed over five runs are given.

Dataset	Method	Symmetric			Pairflip	Instance	
		20%	50%	80%	45%	20%	40%
CIFAR-10	CE	86.5±0.6	80.6±0.2	63.7±0.8	74.9±1.7	87.5±0.5	78.9±0.7
	MixUp	93.2±0.3	88.2±0.3	73.3±0.3	82.4±1.0	93.3±0.2	87.6±0.5
	DivideMix	95.6±0.1	94.6±0.1	92.9±0.3	85.6±1.7	95.5±0.1	94.5±0.2
	ELR+	94.9±0.2	93.6±0.1	90.4±0.2	86.1±1.2	94.9±0.1	94.3±0.2
	PES	95.9±0.1	95.1±0.2	93.1±0.2	94.5±0.3	95.9±0.1	95.3±0.1
	PADDLES	**96.1±0.1**	**95.3±0.2**	**93.3±0.1**	**94.6±0.1**	**96.2±0.1**	**95.5±0.2**
CIFAR-100	CE	57.9±0.4	47.3±0.2	22.3±1.2	38.5±0.6	56.8±0.4	48.2±0.5
	MixUp	69.5±0.2	57.1±0.6	34.1±0.6	44.2±0.5	67.1±0.1	55.0±0.1
	DivideMix	75.3±0.1	72.7±0.6	56.4±0.3	48.2±1.0	75.2±0.2	70.9±0.1
	ELR+	75.5±0.2	71.0±0.2	50.4±0.8	65.3±1.3	75.8±0.1	74.3±0.3
	PES	77.4±0.3	74.3±0.6	61.6±0.6	73.6±1.7	77.6±0.3	76.1±0.4
	PADDLES	**77.9±0.1**	**74.8±0.3**	**62.9±0.3**	**74.7±1.5**	**77.7±0.3**	**76.3±0.1**

Table 3: Comparison with different methods of test accuracy on Cloting-1M. All methods use a pretrained ResNet-5 architecture. Results of other methods are taken from the original papers. * indicates that the methods are based on an ensemble model, while other methods are obtained with a single network.

CE	Forward-T	JointOptim	DMI	ELR	CORES^2	SOP
69.21	69.84	72.16	72.46	72.87	73.24	73.50
T-revision	PES	DivideMix*	ELR+*	PES*	PADDLES	PADDLES*
74.18	74.64	74.76	74.81	74.99	**74.90**	**75.07**

Table 4: Comparison with state-of-the-art methods on CIFAR-N. Mean and standard deviation over five runs are reported. The results of the baseline methods are taken from the leaderboard in Wei et al. (2022). We use ResNet-34 as backbone like other methods expect for SOP+, which adopted PreActResNet-18.

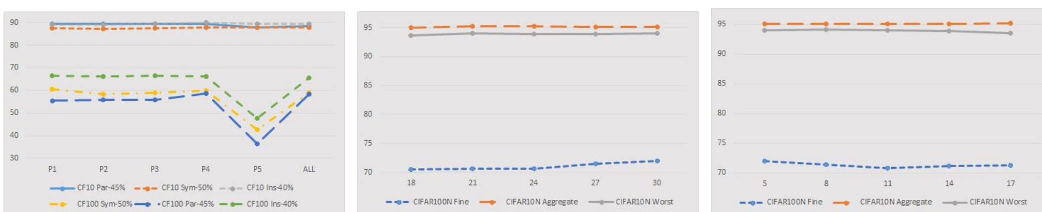
Method	CIFAR-10N					CIFAR-100N
	Random 1	Random 2	Random 3	Aggregate	Worst	Noisy Fine
CE	85.02±0.65	86.46±1.79	85.16±0.61	87.77±0.38	77.69±1.55	55.50±0.66
Forward-T	86.88±0.50	86.14±0.24	87.04±0.35	88.24±0.22	79.79±0.46	57.01±1.03
T-revision	88.33±0.32	87.71±1.02	87.79±0.67	88.52±0.17	80.48±1.20	51.55±0.31
Co-Teaching	90.33±0.13	90.30±0.17	90.15±0.18	91.20±0.13	83.83±0.13	60.37±0.27
ELR+	94.43±0.41	94.20±0.24	94.34±0.22	94.83±0.10	91.09±1.60	66.72±0.07
CORES*	94.45±0.14	94.88±0.31	94.74±0.03	95.25±0.09	91.66±0.09	55.72±0.42
DivideMix	95.16±0.19	95.23±0.07	95.21±0.14	95.01±0.71	92.56±0.42	71.13±0.48
PES	95.06±0.15	95.19±0.23	95.22±0.13	94.66±0.18	92.68±0.22	70.36±0.33
SOP+	95.28±0.13	95.31±0.10	95.39±0.11	**95.61±0.13**	93.24±0.21	67.81±0.23
PADDLES	**95.86±0.12**	**96.03±0.16**	**95.97±0.15**	95.46±0.14	**93.85±0.34**	**71.32±0.36**

3.3 Ablation Studies

We analyze different components of the PADDLES and summarize the results in Table 5. It can be observed that without PES tricks on updating the latter parts of the model, PADDLES_Base achieves a significant improvement over the baseline CE method. Moreover, compared with other state-of-the-art methods, the PADDLES_Base model also obtains comparable performance. For instance, with 45% Pairflip label noise, PADDLES_Base ranks 3rd and 5th among all ten methods on CIFAR-10 and CIFAR-100, as indicated in Table 1. After incorporating PES training on the latter model parts, the PADDLES obtains further improvement and achieves state-of-the-art performance since the proposed training policy is designed from the view of the data frequency domain, which is orthogonal to the PES strategy.

Table 5: Ablation studies about the proposed PADDLES under the supervised setting, experiments on CIFAR-10 are based on a ResNet-18 backbone, and experiments on CIFAR-100 are based on a ResNet-34 backbone. PADDLE_Base denotes the model without using the PES strategy to train the latter parts of the model $\{f_{\Theta_{j+1}}, \ldots, f_{\Theta_T}\}$ in Equation 5.

Dataset	Method	Symmetric	Pairflip	Instance
		50%	45%	40%
CIFAR-10	CE	75.51±1.24	63.34±6.03	77.00±2.17
	PADDLES_Base	83.40±0.78	82.80±2.02	85.20±0.47
	PES	87.45±0.35	88.43±1.08	89.73±0.51
	PADDLES	87.94±0.22	89.32±0.21	89.87±0.51
CIFAR-100	CE	37.69±3.45	34.10±2.04	42.26±1.29
	PADDLES_Base	47.72±3.55	42.17±2.15	54.68±1.36
	PES	58.90±2.72	57.18±1.44	65.68±1.41
	PADDLES	59.78±3.15	58.68±1.28	66.11±1.19



(a) Disentangle Position (b) Stopping Point of AS (c) Stopping Point of PS

Figure 3: Sensitivity analysis for different choices of disentangle positions, early stopping points of AS, and early stopping points of PS.

Another important component of the PADDLES is the frequency disentangle position j, as presented in Algorithm 1. We choose ResNet models as the backbone and disentangle the deep feature at each ResNet block. For example, 'P1' indicates decomposing the feature before block 1, 'P5' is after block 4, and 'ALL' means decomposing the feature at all five positions. As shown in Figure 3(a), we observe that the performance of PADDLES is more stable on CIFAR-10 than on CIFAR-100 at different positions. The best performances are achieved at P3 and P4.

We investigate the hyper-parameter sensitivity of the early stopping points for amplitude spectrum T_A and phase spectrum T_P in Figure 3(b) and Figure 3(c). All experiments are conducted on CIFAR-N datasets with a ResNet-34 backbone. We vary T_A from 18 to 30 with $T_P = 5$ in Figure 3(b) and set T_P from 5 to 17 with $T_A = 30$. We observe that with fixed T_P, the performance will generally increase when T_A is growing for both Fine noises on the CIFAR-100N and Worst noises on the CIFAR-10N. When the T_A is fixed, too large training steps for PS will result in performance degradation, as the model starts to overfit the label noises. Moreover, The performances of Aggregate noises on the CIFAR-10N dataset stay comparatively stable compared with other noises. The model achieves the best performance with $T_A = 30$ and $T_P = 5$.

4 Conclusion

The impact of the noisy labels for the phase spectrum (PS) is less than the amplitude spectrum (AS), resulting in a different fit speed of noisy data. Therefore, we propose a Phase-AmplituDe DisentangLed Early Stopping (PADDLES) method to tackle the learning with noisy labels. During different training steps, we disentangle the AS and PS from the deep image features and separately detach their backpropagation. This way, PADDLES avoids stopping the model training of different frequency components simultaneously and thus achieves better performance. Extensive experiments on different types of data (images and texts) with different network architectures (CNNs and MLP) demonstrate the effectiveness of PADDLES, and PADDLES achieves state-of-the-art performance on five noisy label benchmarks.

ACKNOWLEDGMENTS

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

REFERENCES

Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *ICML*, pp. 312–321. PMLR, 2019.

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *ICML*, pp. 233–242. PMLR, 2017.

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.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *NeurIPS*, 32, 2019.

Peng Bian and Liming Zhang. Biological plausibility of spectral domain approach for spatiotemporal visual saliency. In *International conference on neural information processing*, pp. 251–258. Springer, 2008.

Kenneth R Castleman. *Digital image processing*. Prentice Hall Press, 1996.

Guangyao Chen, Peixi Peng, Li Ma, Jia Li, Lin Du, and Yonghong Tian. Amplitude-phase recombination: Rethinking robustness of convolutional neural networks in frequency domain. In *ICCV*, pp. 458–467, 2021.

Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. *ICLR*, 2021.

Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. Deep learning with label differential privacy. *NeurIPS*, 34:27131–27145, 2021.

Dennis C Ghiglia and Mark D Pritt. Two-dimensional phase unwrapping: theory, algorithms, and software. *A Wiley Interscience Publication*, 1998.

Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. *ICLR*, 2017.

Chenlei Guo, Qi Ma, and Liming Zhang. Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In *CVPR*, pp. 1–8. IEEE, 2008.

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.

Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *ICLR*, 2020.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *NeurIPS*, 32, 2019.

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*, pp. 2304–2313. PMLR, 2018.

Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *ICML*, pp. 143–151, 1997.

Jan Kremer, Fei Sha, and Christian Igel. Robust active label correction. In *AISTATS*, pp. 308–316. PMLR, 2018.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Jia Li, Ling-Yu Duan, Xiaowu Chen, Tiejun Huang, and Yonghong Tian. Finding the secret of image saliency in the frequency domain. *IEEE TPAMI*, 37(12):2428–2440, 2015.

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

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *AISTATS*, pp. 4313–4324. PMLR, 2020b.

Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *CVPR*, pp. 772–781, 2021.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *NeurIPS*, 33, 2020.

Sheng Liu, Kangning Liu, Weicheng Zhu, Yiqiu Shen, and Carlos Fernandez-Granda. Adaptive early-learning correction for segmentation from noisy annotations. *CVPR*, 2022a.

Sheng Liu, Zhihui Zhu, Qing Qu, and Chong You. Robust training under label noise by over-parameterization. In *ICML*, volume 162, pp. 14153–14172. PMLR, 2022b.

Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *ICML*, pp. 6226–6236. PMLR, 2020.

Yueming Lyu and Ivor W Tsang. Curriculum loss: Robust learning and generalization against label corruption. In *ICLR*, 2020.

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

Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. Self: Learning to filter noisy labels with self-ensembling. In *ICLR*, 2020.

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

Alan V Oppenheim and Jae S Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69(5):529–541, 1981.

Alan V Oppenheim, Alan S Willsky, Syed Hamid Nawab, Gloria Mata Hernández, et al. *Signals & systems*. Pearson Educación, 1997.

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*, pp. 1944–1952, 2017.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *NeurIPS*, 34:20596–20607, 2021.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pp. 1532–1543, 2014.

Scott E Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *ICLR (Workshop)*, 2015.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, pp. 4334–4343. PMLR, 2018.

Eero P Simoncelli and Odelia Schwartz. Modeling surround suppression in v1 neurons with a statistically derived normalization model. *NeurIPS*, pp. 153–159, 1999.

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, pp. 369–386. SPIE, 2019.

Zeren Sun, Yazhou Yao, Xiu-Shen Wei, Yongshun Zhang, Fumin Shen, Jianxin Wu, Jian Zhang, and Heng Tao Shen. Webly supervised fine-grained recognition: Benchmark datasets and an approach. In *ICCV*, pp. 10602–10611, 2021.

Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, pp. 5552–5560, 2018.

Kiran K Thekumparampil, Ashish Khetan, Zinan Lin, and Sewoong Oh. Robustness of conditional gans to noisy labels. *NeurIPS*, 31, 2018.

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

Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *IJCV*, 108(1):97–114, 2014.

Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *ICCV*, pp. 8684–8694, 2020.

Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. In *CVPR*, pp. 8688–8696, 2018.

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *ICLR*, 2022. URL `https://openreview.net/forum?id=TBWA6PLJZQm`.

Peter Welinder and Pietro Perona. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In *CVPR Workshops*, pp. 25–32. IEEE, 2010.

Songhua Wu, Mingming Gong, Bo Han, Yang Liu, and Tongliang Liu. Fair classification with instance-dependent label noise. In *Conference on Causal Learning and Reasoning*, pp. 927–943. PMLR, 2022.

Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? *NeurIPS*, 32, 2019.

Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *ICLR*, 2020a.

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, 2020b.

Xiaobo Xia, Shuo Shan, Mingming Gong, Nannan Wang, Fei Gao, Haikun Wei, and Tongliang Liu. Sample-efficient kernel mean estimator with marginalized corrupted data. In *KDD*, pp. 2110–2119, 2022.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015.

Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. *NeurIPS*, 32, 2019.

Erkun Yang, Dongren Yao, Tongliang Liu, and Cheng Deng. Mutual quantization for cross-modal search with noisy labels. In *CVPR*, pp. 7551–7560, 2022.

Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. *NeurIPS*, 33:7260–7271, 2020.

Yu Yao, Tongliang Liu, Mingming Gong, Bo Han, Gang Niu, and Kun Zhang. Instance-dependent label-noise learning under a structural causal model. *NeurIPS*, 34:4409–4420, 2021.

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *ICML*, pp. 7164–7173. PMLR, 2019.

Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. Learning with biased complementary labels. In *ECCV*, pp. 68–83, 2018.

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

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *NeurIPS*, 31, 2018.

Zhaowei Zhu, Tongliang Liu, and Yang Liu. A second-order approach to learning with instance-dependent label noise. In *CVPR*, pp. 10113–10123, 2021.

A Related Work

Learning with noisy labels Current methods (Reed et al. (2015); Goldberger & Ben-Reuven (2017); Malach & Shalev-Shwartz (2017); Patrini et al. (2017); Thekumparampil et al. (2018); Zhang & Sabuncu (2018); Kremer et al. (2018); Han et al. (2018); Ren et al. (2018); Yu et al. (2018); Jiang et al. (2018); Xu et al. (2019); Yu et al. (2019); Liu & Guo (2020); Li et al. (2020b;a); Hu et al. (2020); Lyu & Tsang (2020); Yao et al. (2020); Xia et al. (2020b); Yao et al. (2021); Cheng et al. (2021); Zhu et al. (2021); Ghazi et al. (2021); Paul et al. (2021); Yang et al. (2022); Wu et al. (2022); Liu et al. (2022b); Xia et al. (2022); Wei et al. (2022)) of learning with noisy labels (LNL) can be roughly grouped into two categories: model-based and model-free approaches.

Model-based methods (Patrini et al. (2017); Xia et al. (2020a;b); Yao et al. (2020); Liu et al. (2022b)) propose to directly learn the relations between noisy and clean labels based on the assumption that the noisy label is sampled from a conditional probability distribution on the true labels. Hence, the core idea of these methods is to estimate the underlying noise transition probabilities. For instance, (Goldberger & Ben-Reuven (2017)) used a noise adaptation layer on the top of a classification model to learn the transition probabilities. T-revision (Xia et al. (2019)) added fine-tuned slack variables to estimate the noise transition matrix without anchor points. Moreover, a recent work (Liu et al. (2022b)) proposed to model the label noise via a sparse over-parameterized term and use implicit algorithmic regularizations to recover the underlying mislabels. These methods hold some (somewhat strong) assumptions about the noisy label distribution, which may be inapplicable in some scenarios. Our method does not focus on particular label distribution and therefore does not belong to model-based methods.

Instead of modeling the noisy label directly, model-free methods (Han et al. (2018); Li et al. (2020a); Bai et al. (2021); Xia et al. (2020a)) aim to utilize the memorization effect of deep models to suppress the negative impact of the noisy labels. The memorization effect (Arpit et al. (2017)) indicates that the deep networks tend to fit the clean data first and then memorize the noise ones, which inspired the model-free methods. A representative method is Co-teaching (Han et al. (2018)), which uses two deep networks to train each other with small-loss instances in mini-batches. DivideMix (Li et al. (2020a)) further extended Co-teaching with two Beta Mixture Models. Moreover, DivideMix imported MixMatch (Berthelot et al. (2019)) training to utilize the unlabeled (unconfident) samples to boost the deep models. PES (Bai et al. (2021)) investigated the progressive early stopping of deep networks, which selects different early stopping for different parts of the deep model and achieved significant improvement over previous early stopping methods. Unlike existing model-free methods, our method is the first work designed from the data domain's perspective in frequency representation. Inspired by the biological analysis of the vision system on different spectrums, we find that the Phase spectrum is more resistant to noisy labels than the Amplitude spectrum. Therefore, we propose to disentangle the different components of the frequency domain and choose different early stopping strategies, which further exploit the memorization effect and achieve good performance.

Convolutional neural networks with frequency domain To explain the behavior of Convolutional Neural Networks (CNNs), recent studies provide new insights from the viewpoint of the frequency domain (Ilyas et al. (2019); Wang et al. (2020); Liu et al. (2021); Chen et al. (2021). (Wang et al. (2020)) points out that high-frequency components from the image play significant roles in improving the performance of CNNs. Moreover, (Liu et al. (2021)) investigated the phase spectrum in face forgery detection and inducted that urging CNNs to learn the phase spectrum can boost the detection accuracy. APR (Chen et al. (2021)) presented qualitative and quantitative analyses of amplitude and phase spectrums for CNNs and concluded that a robust deep model should resist amplitude noises and perceive more phase spectrum. Inspired by these breakthroughs, we are the first to investigate the frequency domain in learning with noisy labels and find that the sensitivity of phase and amplitude components are different. Furthermore, we propose to dynamic stop the optimization of CNN on different frequency components in training, which well-address the over-fitting problem of noisy labels.

B 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 Van Rooyen et al. (2015) (Symmetric), pairflip class-dependent label noise Han et al. (2018) (Pairflip), and instance-dependent label noise Xia et al. (2020b) (Instance). We follow the implementation of (Han et al. (2018); Xia et al. (2020b); Bai et al. (2021)) to generate these label noises with different levels, which can be found in PES.

Data preprocessing For learning with confident samples (Table 1), we apply the random crop and random horizontal flip as data augmentations. We further add MixUp Zhang et al. (2018) data augmentation for semi-supervised settings in Table 2. For CIFAR-N dataset (Table 4), we use random crop, random horizontal, and a CIFAR-10 augmentation policy from (Nishi et al. (2021)). The input image size of CIFAR-like datasets is set as 32×32. For the Clothing-1M dataset (Table 3), we first resize input images to the size of 256×256, then randomly crop the image as 224×224, and random horizontal flip the images 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 Bai et al. (2021) 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 1 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 Bai et al. (2021) as 5. We train the semi-supervised models using MixMatch Berthelot et al. (2019) loss with same parameters (λ_u, T, K) in Bai et al. (2021). Moreover, we set T_0 in Algorithm 1 as 0.

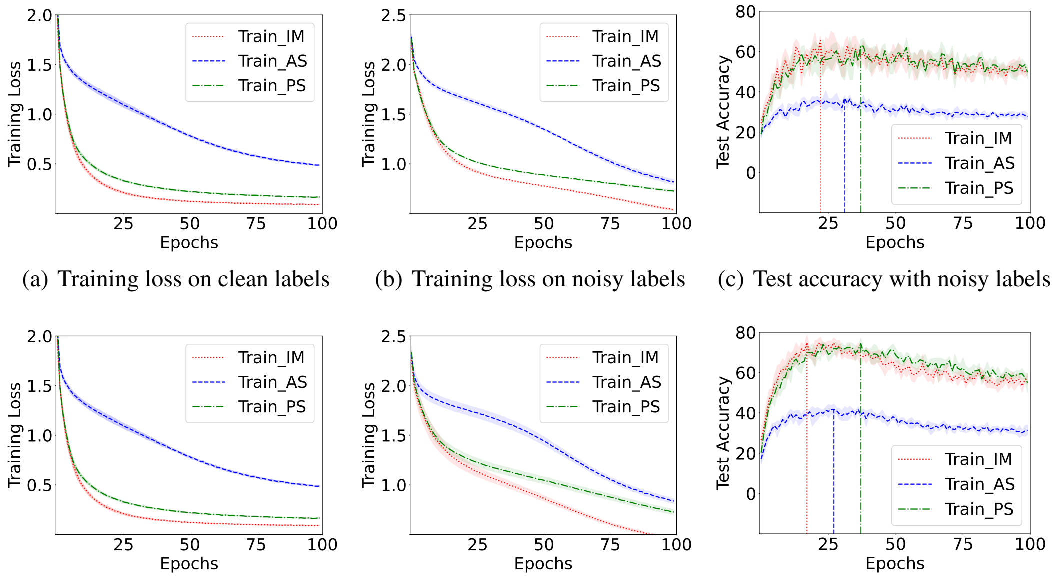
For CIFAR-N datasets, we use the ResNet-34 architecture. We set the learning rate as 0.02, the batch size as 128, the weight decay as 5×10^{-4}, the training epochs as 300, the T_2 in PES as 5. We also employ the MixMatch loss to train the semi-supervised model with MixMatch parameter λ_u as 5 and 75 for CIFAR-10N and CIFAR-100N, respectively. We set T_0 in Algorithm 1 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 Smith & Topin (2019) 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 1 is as 0. More details will be found in our scheduled released codes.

C Additional Experiments

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

We first give more illustration about the impact of different kinds of label noises on deep models in Figure 4. We generate two more kinds of label noises: the Pairflip Han et al. (2018) with a 45% noise rate and the Instance Xia et al. (2020b) with a 40% noise rate. As can be observed that the inflection point of AS's loss decline is earlier than that of PS components, which means the converge speed of CNN on AS is faster than PS. Moreover, the curves of AS and PS get closer as the training epochs increase, indicating that the PS is more robust than AS with different label noises. Another evidence of the difference between AS and PS is that the number of training steps to achieve optimal performance is not the same, and Figures 4(c) and 4(f) show that AS costs less time, achieving the



(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 4: 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 4(a) and Figure 4(b) 4(e)) and testing accuracy with the noisy labels (Figure 4(c) 4(f)) are given. The X-axis illustrates the training epochs. Figure 4(b) 4(c) are based on the 45% Pairflip label noises and Figure 4(e) 4(f) 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.

best performance than PS. Both Figure 1 and Figure 4 inspire us to decompose the AS and PS from the input images and design different stopping points to obtain a more robust deep network over previous ES models.

C.1 Confident Samples Quality

Following (Bai et al. (2021)), we examine the extracted labels' quality in terms of three aspects: test accuracy, label recall, and label precision using CIFAR-10, where label recall indicates the ratio of extracted confident samples with correct labels to the whole correctly labeled samples, and label precision indicates the ratio of extracted confident samples with correct labels to the whole confident samples. Specifically, we train a neural network based on ResNet-18 with various kinds and levels of label noise for total 25 epochs separately. As for our methods, the disentangle point is set between the 3rd and 4th ResNet blocks, while the stopping points of $\mathcal{AS}_\chi$, $\mathcal{PS}_\chi$ are set to 23 and 25, respectively. The results are shown in Table 6.

From the results in Table 6, we can clearly observe that the models generally outperform the corresponding CE and PES methods when using our methods. That is, our methods can help to obtain higher accuracy, recall, and comparable precision in the majority of cases. The collection of more confident samples is essential for learning with confident samples and semi-supervised learning. More importantly, models with high recall values can help to collect more confident samples for the following supervised or semi-supervised training. Consequently, PADDLES can contribute to improving the final classification performance in all cases by improving the performance of the initial model, which is also supported by the experiments in Section 3.

Table 6: 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

C.2 Training Time Comparison

We compare the training time of proposed PADDLES and other baseline methods. For fairness, we follow Bai et al. (2021) to conduct the experiments based on a single Nvidia V100 GPU server. Moreover, we run 200 and 300 training epochs for supervised and semi-supervised settings (noted as PADDLES(Semi)), respectively. The results are presented in Table 7. The proposed PADDLES model costs 1.55h for the supervised training, which is faster than the three methods (CDR, ELR+, and DivideMix) and achieves comparable training speed to Co-teaching. For the semi-supervised setting, due to the import of DFT, iDFT, and MixMatch training, PADDLES is slower than PES but still faster than DivideMix.

Table 7: Training time comparison for different methods on CIFAR-10 with 50% Symmetric label noise. The results of the baseline methods are taken from Bai et al. (2021).

CE	Co-teaching	CDR	T-revision	ELR+	DivideMix	PES	PES(Semi)	Ours	Ours(Semi)
0.9h	1.5h	3.0h	3.5h	2.2h	5.5h	1.0h	3.1h	1.55h	4.8h

C.3 Text Classification

In order to further explore the generalizability of PADDLES, we also evaluate it on the text dataset NEWS. The NEWS dataset, also known as 20 Newsgroups (Joachims (1997)), collected by Ken Lang, is widely used as a benchmark for text classification. The original NEWS dataset contains approximately 20,000 articles among 20 classes. For fairness comparison, we follow Co-teaching+ (Yu et al. (2019)) to re-organize the dataset with 7 classes and set 11,314 samples for training and 7,532 samples for testing. To test the extreme performance of models, we selected two difficult typical noise types with high noise rates: Symmetric 80% and Pariflip 45%.

We adopt the same network architecture of NEWS in (Yu et al. (2019)) as the backbone to build PES-like models and our PADDLES-like models. Specifically, the backbone consists of a pretrained word embedding layer (Pennington et al. (2014)) followed by a 3-layer MLP with Softsign active function. Besides the PES, PADDLES, and their semi-supervised versions, we also extend these two ES strategies into Co-teaching frameworks, denoted as PES_Co-teaching/+ and PADDLES_Co-teaching/+ in Table 8. We empirically choose different parameters to obtain the best performance for each approach. For example, PADDLES_Co-teaching+ adopts the PADDLES training stage to obtain good initial models for Co-teaching training, the disentangle point is set between the 2nd and 3rd layers of the MLP backbone, while the stopping points of $\mathcal{AS}_{\chi}$, $\mathcal{PS}_{\chi}$ are set to 3 and 6, respectively. We train 2 models simultaneously with PADDLES, end the PADDLES training after 6 epochs, and then pass the 2 models into the Co-teaching+ network to continue the training for 20 epochs following the ways in (Yu et al. (2019)). The results are shown in Table 8.

Through the results in Table 8, we observe that the Co-teaching methods achieve superior performances over PES and PADDLES, under heavy noises, which might be caused by the difference be-

Table 8: Test accuracy comparison with state-of-the-art methods on the text dataset NEWS Yu et al. (2019). Mean and standard deviation over five runs are reported.

Method	Symmetric	Pariflip
	80%	45%
CE	19.00±0.41	31.94±0.38
PES	20.69±1.42	31.99±0.41
PADDLES	21.30±1.73	32.45±0.91
PES(Semi)	22.00±2.89	35.45±1.77
PADDLES(Semi)	22.97±4.76	35.51±1.75
Co-teaching	23.26±2.99	35.94±2.68
Co-teaching+	23.52±2.72	34.65±2.25
PES_Co-teching+	24.11±1.29	35.21±2.04
PADDLES_Co-teaching+	**25.66**±**2.63**	**36.04**±**1.89**

tween the text and image data. The proposed PADDLES still outperforms the baseline CE and PES models consistently. More importantly, with PADDLES pretrained base models, PADDLES_Co-teaching+ achieves the state-of-the-art among all methods. As PADDLES is proposed from the data view, it can be combined with different LNL models and help to obtain more confidence samples. Therefore, by training with more confident samples, we can provide a more robust initial model for other subsequent models. Overall, we demonstrate the effectiveness of the proposed PADDLES for different input signals (images and texts) as well as various backbones (CNNs and MLP).