A APPENDIX

A.1 MARGIN DISTRIBUTION ANALYSIS

We analyze how the margin distribution evolves during training to better understand the impact of the length of *exploration* phase on the generalization performance. The decision boundary between two classes i and j is defined as follows.

$$d_{f,x,(i,j)} := \min_{\delta} \|\delta\|_2 \quad s.t. \quad f_i(x+\delta) = f_j(x+\delta) \tag{11}$$

Because the above 'exact' distance for a deep neural network is intractable, we employ an approximation technique proposed in (Jiang et al. (2018)).

$$d_{f,x,(i,j)} = \frac{f_i(x) - f_j(x)}{\|\nabla f_i(x) - \nabla f_j(x)\|_2}, \tag{12}$$

where $f_i(x)$ is the output of the network logit i given the model parameters x, and $\nabla f_i(x)$ is the gradients with respect to the model parameters of logit i. We use the normalized margin $\hat{d}_{f,x,(i,j)}$ that is $d_{f,x,(i,j)}$ divided by the square root of total variation of the input x. Similarly to the reference work (Jiang et al. (2018)), we ignore the margins smaller than the first quartile and larger than the third quartile. We collect the margins across all training samples and compare the accumulated margin to the training loss curve. The margin curves are shown in Figure 1.

A.2 DATA AUGMENTATIONS AND DETAILED SETTINGS

To enable reproduction, we summarize the detailed settings including data preprocessing and augmentation settings.

CIFAR-10 / CIFAR-100 – For each training image, all individual pixels are subtracted by the mean values and divided by the standard deviation values. We mostly follow the hyper-parameter settings used in (Lin et al. (2018; 2020)). The CIFAR-10 training is performed for 300 epochs. The CIFAR-100 training is performed for 250 epochs.

SVHN – SVHN consists of $73K$ training samples, $26K$ validation samples, and $531K$ extra training samples. We use both sets of training samples when training the model and use the validation samples for evaluating the trained model. Likely to CIFAR datasets, each training image is normalized using the pixel-wise statistics. The training is performed for 160 epochs following the settings in (Zagoruyko & Komodakis (2016)). The drop out factor is set to 0.5.

ImageNet – We adopt several data preprocessing methods used in (He et al. (2019)). First, each training sample is normalized in a channel-wise way. Each image is resized to 256×384 pixels keeping the original aspect ratio, and 224×224 image is randomly cropped from the resized image. The cropped image is horizontally flipped with a probability of 0.5. The brightness is randomly adjusted using a maximum delta value of $32/255$. Then, the image color is randomly adjusted in HSV color space using the maximum delta value of 0.1. The saturation is also augmented between 0.6 and 1.4. Finally, the contrast is also augmented between 0.6 and 1.4. We slightly modify ResNet50 model such that the last batch normalization *gamma* parameters are initialized to 0 in all the residual blocks. The bias parameters at the last fully-connected layer are regularized like all the other weight parameters.

The learning rate decay schedule for *small-batch*, *large-batch*, and *LARS* follows the hand-tuned setting used in Goyal et al. (2017). Although many previous works train ResNet50 for 90 epochs only, we observed that the accuracy was still improved after 90 epochs. So, we perform the training for 100 epochs in total.

A.3 LEARNING CURVES

Here, we show the full training loss curves and the validation curves. To clearly show the performance of *two-phase*, we show the learning curves without using LARS.

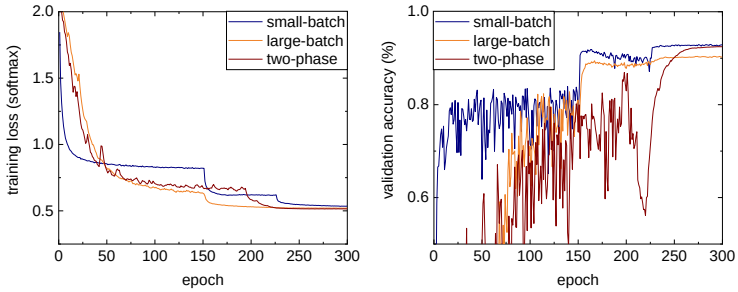

Figure 4: CIFAR-10 (ResNet20) learning curves comparison. The hyper-parameters are shown in Table 2.

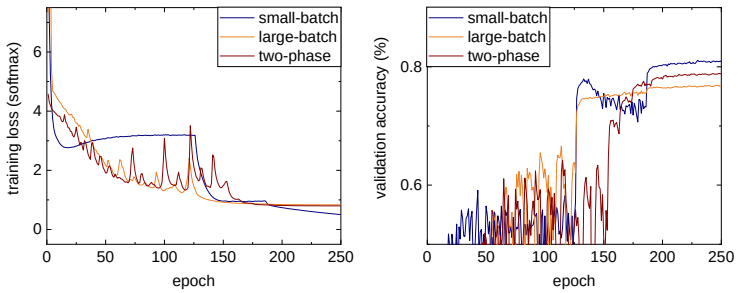

Figure 5: CIFAR-100 (WideResNet28-10) learning curves comparison. The hyper-parameters are shown in Table 3.

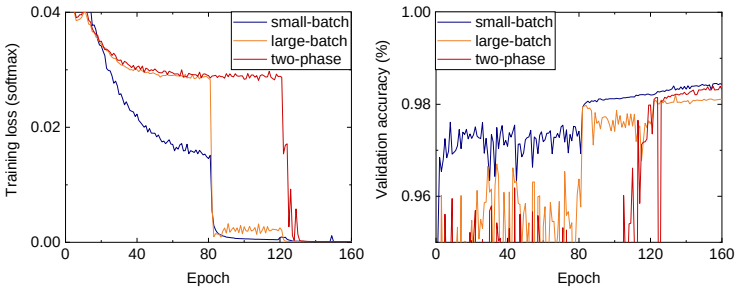

Figure 6: SVHN (WideResNet16-8) learning curves comparison. The hyper-parameters are shown in Table 4.

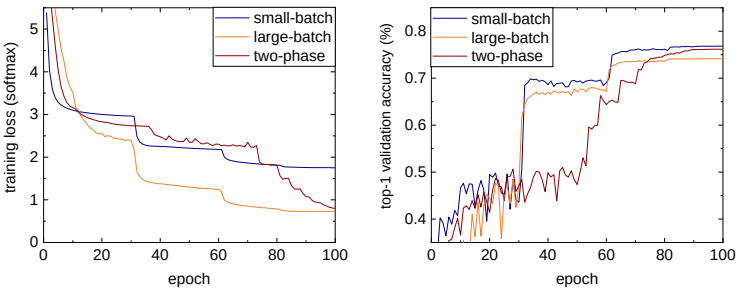

Figure 7: ImageNet (ResNet50) learning curves comparison. The hyper-parameters are shown in Table 5.