Supplementary Material of Partial Forward Blocking: A Novel Data Pruning Paradigm for Lossless Training Acceleration

Supplementary Material

Our code is available at https://github.com/dywu98/OnlineDataPrune.git.

A. More Experimental Results

In this section, we provide more experimental results to help our readers better understand our proposed method.

A.1. Ablation Study on the Weight in ADE

To enhance the adaptability of PFB to the changing distribution caused by parameter updating, we introduce the w_j^t (Eq. (4) and Eq. (9)) as a dynamic weight to adaptively balance between different centroids. By re-scaling the output of each kernel according to the number of samples that were previously assigned to its cluster (represented by the centroids), those centroids with a great number of n_j^t will contribute more to the probability density, thereby reducing the importance score of samples with similar features. We evaluate its effectiveness by replacing the w_j^t with 1 (denoted as 'w/o weight') and comparing the results with PFB in Tab. S1. Although PFB w/o weight still outperforms DivBS and InfoBatch, the performance drops by a large margin when the prune ratio grows. This phenomenon shows the effectiveness of the balancing weight, which may lead to a better estimation of the probability density function.

Pruning Ratio	30%	50%	70%
Full Data		78.2	
InfoBatch*[7]	$78.2_{\uparrow 0.0}$	$78.1_{\downarrow 0.1}$	$76.5_{\downarrow 1.7}$
DivBS[6]	$78.5_{\uparrow 0.3}$	$78.2_{\uparrow 0.0}$	$77.2_{\downarrow 1.0}$
PFB w/o weight	$78.9_{\uparrow 0.7}$	$78.4_{\uparrow 0.2}$	$77.4_{\downarrow 0.8}$
PFB(ours)	$\mathbf{79.1}_{\uparrow \mathbf{0.9}}$	$\mathbf{78.8}_{\uparrow \mathbf{0.6}}$	$\mathbf{77.9}_{\downarrow \mathbf{0.3}}$

Table S1. Ablation study on the weight in ADE for balancing different centroids. Experiments are conducted on CIFAR-100 using ResNet-18. We use 'w/o weight' to denote our modification that w_j^t is replaced by 1 in Eq. (4) of the main text.

A.2. Ablation Study on the Bandwidth Estimation Methods in ADE

As bandwidth estimation is usually deemed important for KDE methods, we compare the performance of two popular bandwidth estimation rules, namely Scott's rule[8] and Silverman's rule[9]. We also introduce a baseline denoted as 'Identity'. This baseline simply sets the $\mathbf{H}$ as an identity matrix. Results in Tab. S2 indicate that a proper bandwidth is crucial for the performance of our PFB. However, there is no big difference between those two commonly used bandwidth estimation methods.

Methods	**Bandwidth**	30%	50%	70%
Full Data	-		78.2	
InfoBatch*[7]	-	$78.2_{\uparrow 0.0}$	$78.1_{\downarrow 0.1}$	$76.5_{\downarrow 1.7}$
DivBS[6]	-	$78.5_{\uparrow 0.3}$	$78.2_{\uparrow 0.0}$	$77.2_{\downarrow 1.0}$
PFB	Identity	$77.4_{\downarrow 0.8}$	$76.5_{\downarrow 1.7}$	$75.1_{\downarrow 3.1}$
	Scott[8]	$79.0_{\uparrow 0.8}$	$\mathbf{78.8}_{\uparrow \mathbf{0.6}}$	$\mathbf{77.9}_{\downarrow \mathbf{0.3}}$
	Silverman[9]	$\mathbf{79.1}_{\uparrow \mathbf{0.9}}$	$\mathbf{78.8}_{\uparrow \mathbf{0.6}}$	$\mathbf{77.9}_{\downarrow \mathbf{0.3}}$

Table S2. Ablation study on different bandwidth estimation methods. Experiments are conducted on CIFAR-100 using ResNet-18.

A.3. Detailed Explanation of InfoBatch

InfoBatch[7] employs a soft pruning ratio, using the mean loss value of all samples as a threshold to divide the dataset into two subsets. Samples with a loss lower than this threshold form a candidate subset, where each sample has a pruning probability of p, which is reported as the pruning ratio. However, since the candidate subset only contains half of the samples, the actual pruning ratio of InfoBatch is $p/2$. To highlight this distinction, we denote the original version of InfoBatch with '*' in the main text. For a fair comparison, we follow the modification applied by DivBS[6] to InfoBatch, where the threshold is set to the 95% percentile to align with the actual pruning ratio of most pruning methods. We denote this modified version as InfoBatch†. In the main text, Tab. 2-4 present a comparison between our method and InfoBatch† on ImageNet-1k, Cityscapes, and PASCAL VOC 2012. Here, we further provide experimental results on CIFAR-100 in Tab. S3 to supplement the comparison. Aligning the actual pruning ratio reveals a significant performance drop for InfoBatch on CIFAR-100. Beyond the impact of the adjusted pruning ratio, this decline may also stem from the large weights applied to the retained samples within the pruned candidate subset by InfoBatch. (Please refer to [7] for details of this re-scaling operation.) Such a large weight may excessively emphasize the retained samples, potentially hindering the learning of harder examples in the other subset.

A.4. Error Bars

Error statistics for ResNet-18 and Swin-T on different datasets are also included in Tab. S4, exhibiting variations

Pruning Ratio	30%	50%	70%
Actual Ratio	15%	25%	35%
InfoBatch*[7]	78.2$_{\uparrow 0.0}$	78.1$_{\downarrow 0.1}$	76.5$_{\downarrow 1.7}$
Pruning Ratio	30%	50%	70%
InfoBatch†	78.0$_{\downarrow 0.2}$	76.0$_{\downarrow 2.2}$	74.3$_{\downarrow 3.9}$
DivBS[6]	78.5$_{\uparrow 0.3}$	78.2$_{\uparrow 0.0}$	77.2$_{\downarrow 1.0}$
PFB(ours)	**79.1**$_{\uparrow \mathbf{0.9}}$	**78.8**$_{\uparrow \mathbf{0.6}}$	**77.9**$_{\downarrow \mathbf{0.3}}$
Full Data		78.2	

Table S3. Fair comparison on CIFAR-100 with InfoBatch.

within an acceptable range. The results show that PFB can maintain a stable performance with negligible variation.

Dataset/Model	CIFAR-10 / ResNet-18			CIFAR-100 / ResNet-18			ImageNet-1K / Swin-T		
Pruning Ratio	30%	50%	70%	30%	50%	70%	30%	40%	50%
PFB(Ours)	95.9 $\pm$0.1	95.5 $\pm$0.1	95.2 $\pm$0.2	79.1 $\pm$0.2	78.8 $\pm$0.2	77.9 $\pm$0.2	79.6 $\pm$0.1	79.2 $\pm$0.2	78.2 $\pm$0.2
Full Data	95.6 $\pm$0.1			78.2 $\pm$0.1			79.6 $\pm$0.1		

Table S4. Error bars on CIFAR-10/100 and ImageNet-1k.

B. Implementation Details

We further demonstrate the details of experiments on image classification and segmentation datasets here.

B.1. Classification Training Settings

All the classification experiments are conducted on a 4-RTX 4090 GPU server. We follow the training details of InfoBatch[7] on CIFAR-10/100 using ResNet-18 and ImageNet-1k using ResNet-50. For Swin-T, we adopt similar training settings of Dyn-Unc[5]. The AutoAugment[4] is applied to augment training data only for Swin-T, including random path drop and gradient clipping for a fair comparison with Dyn-Unc[5] in Tab. (2) of the main text. All the detailed settings needed for reproduction are listed in the Tab. S5.

B.2. Details of Segmentation Experiments

Our segmentation experiments are based on the implementation of MMSegmentation[2]. On PASCAL VOC 2012[1] and Cityscapes[3], the models are trained for 36,000 iterations. Other training and evaluation details remain the same with MMSegmentation. Please note that the reported mIoU results in Tab. (3) and (4) employ the popular multi-scale evaluation technique. Moreover, as most semantic segmentation methods employ an auxiliary segmentation head at the third stage of the encoder, we extract the features at this stage to utilize the abundant semantic information. Please note that there is a big difference between segmentation and classification networks: aside from the encoder, segmentation networks usually have a computationally expensive decoder. Hence, blocking at the third stage of the encoder can still significantly cut down the training time.

	Parameters	**CIFAR-10**	**CIFAR-100**	**ImageNet-1k**	
	Models	ResNet-18	ResNet-18	ResNet-50	Swin-T
Training	optimizer	SGD	SGD	Lars	AdamW
	weight_decay	0.0005	0.0005	0.00005	0.05
	batch_size	128	128	1024	1024
	epochs	200	200	90	300
	learning_rate	0.10	0.05	6.4	0.001
	label smoothing	0.1	0.1	0.1	0.1
	learning rate scheduler	OncCycle	OncCycle	OneCycle	CosineAnnealing
	learning rate warmup	-	-	5	20
Data Pruning	b	0.01	0.01	0.001	0.001
	N_C	64	64	64	64
	D	128	128	128	128
	PFB Location	stage-1	stage-1	stage-2	stage-1
	epoch start pruning	5	5	5	15
	epoch stop pruning	180	180	80	265

Table S5. Detailed training settings on image classification datasets.

References

[1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 2

[2] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 2

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 2

[4] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018. 2

[5] Muyang He, Shuo Yang, Tiejun Huang, and Bo Zhao. Large-scale dataset pruning with dynamic uncertainty. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7713–7722, 2024. 2

[6] Feng Hong, Yueming Lyu, Jiangchao Yao, Ya Zhang, Ivor Tsang, and Yanfeng Wang. Diversified batch selection for training acceleration. In *Forty-first International Conference on Machine Learning*, 2024. 1, 2

[7] Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Daquan Zhou, Lei Shang, Baigui Sun, Xuansong Xie, Yang You, et al. Infobatch: Lossless training speed up by unbiased dynamic data pruning. In *The Twelfth International Conference on Learning Representations*, 2023. 1, 2

[8] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015. 1

[9] Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018. 1