1 Training Overheads Comparison

We train CS-based CNNs to compare with other types of sparse CNNs on CIFAR100 and ImageNet, respectively. Firstly, the accuracy of sparse CNNs on CIFAR100, i.e., Table 3 in the main text, is obtained with the same training hyperparameters and epochs. In particular, we have presented in Table 3 the results of our CS under identical training settings with other sparse patterns, e.g., '**CS-C**' rows. By comparison, '**CS(Ours)**' rows are results using our proposed training scheme.

Secondly, accuracy results on ImageNet of ResNet50 with N:M, OVW, and Shfl_BW patterns in Table 4 are directly cited from related works. Replicating the works of N:M, OVW, and Shfl_BW patterns on ImageNet with multiple sparsity ratios is prohibitively time-consuming. In a sense, it is very hard to ensure an absolutely fair comparison on the ImageNet dataset. Nevertheless, we list the training settings of our work and related works on ImageNet as shown in Table 10 to ensure clarity. Note that although the state-

Hyperparameters	Shfl_BW $_{\times 2}$	OVW$_{\times 2}$	N:M	Ours$_{\times 2}$
Scheme	Pretrain+finetune	Pretrain+finetune	Training from scratch	Pretrain+finetune
Batch_size	N/A	N/A	256	32
Epochs	N/A	100	120	90
Initial lr	N/A	0.1/0.0008	0.1	0.1
Momentum	N/A	N/A	N/A	0.9
Extra method	**Heuristic searching**	**Row clustering**	**Sparse-refined regularization**	**Gradual pruning**

Table 10: Training settings on ImageNet

of-the-art N:M work only employs 120 epochs in their training scheme, the time of one epoch in their scheme is much longer than that in other works' training schemes. It is because their sparse-refined regularization incurs extra computations in each iteration of training.

Thirdly, the proposed training scheme theoretically brings down the total training float-point operations (FLOPs) as we do not constantly maintain a dense network during pretraining. In practice, we found the reduction in FLOPs does not significantly shorten the total training time as the weight re-selection and scheduling sparsity of sparse CNNs in our scheme also incur overheads. This type of overheads is generally not suitable to be measured with the FLOPs metric. On the whole, trainings with and without our scheme under the same hyperparameters, epochs, and hardware take closely similar times. For example, training an N:M-based CNN on ImageNet takes **65.37** hours, while for a CS-based CNN, the time is **65.23** hours.

2 Inference Performance of Block Sparsity

As for block sparsity-based CNNs, there are two versions of GPU implementation, original and Cutlass-based versions (Markidis et al., 2018). After replication as shown in Table 11, we found that both versions' speedup performances are far inferior to ours, not to mention that block sparsity actually incurs accuracy collapse when sparsity ratios larger than 75%.

Sparsity(%)	Cutlass-based version		Original version	
	Time(ms)	Speedup	Time(ms)	Speedup
0	2.21	1	3.76	1
50	1.91	1.16	2.79	1.35
75	1.98	1.12	2.07	1.82
87.5	1.91	1.16	1.53	2.46
93.75	1.9	1.16	0.98	3.84

Table 11: The speedups of block sparsity-based ResNet50

References

Stefano Markidis, Steven Wei Der Chien, Erwin Laure, Ivy Bo Peng, and Jeffrey S Vetter. Nvidia tensor core programmability, performance & precision. In *2018 IEEE international parallel and distributed processing symposium workshops (IPDPSW)*, pp. 522–531. IEEE, 2018.