MiniViT: Compressing Vision Transformers with Weight Multiplexing —— Supplementary Material ——

Jinnian Zhang[1,*], Houwen Peng[1,*,†], Kan Wu[1,*], Mengchen Liu[2], Bin Xiao[2], Jianlong Fu[1], Lu Yuan[2]
[1] Microsoft Research, [2] Microsoft Cloud+AI

In this supplementary material, we present more implementation details and experimental comparisons. Besides, we provide the broader impacts. The main content is summarized as follows.

- In Appendix A, we first study the performance of Mini-Swins in the case of large compression ratio, where all layers of each stage in Swin [3] are shared. Then we investigate the performance of Mini-DeiTs with no additional modifications as elaborated in Sec. 4.1. Moreover, we generate a series of Mini-Swins with different compression ratio by applying various sharing strategies in Sec. 4.2, and provide the plot of their top-1 accuracy on ImageNet-1K [1]. Finally, we visualize the raining instability of weight sharing in Sec. 4.4.

- In Appendix B, we discuss the broader impacts.

A. Additional Experiments

Swin Transformers [3]. We consider the extreme case where all layers in each stage of Swin [3] are shared. As shown in Tab. 1, The compression ratio for Swin-T, Swin-S, and Swin-B are 2.3×, 4.2×, and 4.2× times respectively. Weight sharing leads to performance degradation, although it can efficiently reduce the number of model parameters. For instance, Swin-S suffers from 5.6% degradation in accuracy. Our proposed weight multiplexing, consisting of weight transformation and distillation, can largely alleviate the problem, even in this extreme case. After applying both weight sharing and multiplexing, Swin-S and Swin-B can achieve only 1.2% and 0.3% loss in accuracy, respectively. For Swin-T, our Mini-Swin-T can even increase the top-1 accuracy by 0.2%. These results verify the effectiveness of our proposed MiniViT framework.

DeiT [4]. We apply our proposed MiniViT framework to the official DeiT [4] architectures without additional modifications mentioned in Sec. 4.1. In Tab. 2, our Mini-DeiT-Tiny can achieve a 41.2% reduction in the number of model parameters, with only 0.4% accuracy drop. Moreover, both Mini-DeiT-S and Mini-DeiT-B outperforms the original models with only 50% model size. Therefore, our MiniViT framework is still effective in DeiT without modifications. Furthermore, the performance of Mini-DeiTs can be improved after applying the modifications in Sec. 4.1.

Effects of Compression Ratio. We generate a series of MiniViTs by changing the number of sharing blocks to control the compression ratio. In particular, by sharing every two, three, and all layers in each stage of Swin-T [3], Mini-Swin-T can obtain the compression ratio 44%, 51%, and 57%, respectively. For Swin-S and Swin-B, we consider five different sharing strategies, i.e., sharing every two, three, six, nine, and all layers. Fig. 1 shows the effect of compression ratio on the performance of Mini-Swins. It is clear that increasing the compression ratio will cause per-

Model	WS	MUX	#Params (M)	Top-1 Acc(%)	Top-5 Acc(%)
Swin-B (22k) (Teacher)			88	85.2	97.5
Swin-T			28	81.2	95.5
	✓		12(2.3×)	79.0	94.4
	✓	✓	12(2.3×)	81.4	95.8
Swin-S			50	83.2	96.2
	✓		12(4.2×)	77.6	93.8
	✓	✓	12(4.2×)	82.0	96.1
Swin-B			88	83.5	96.5
	✓		21(4.2×)	80.1	94.9
	✓	✓	21(4.2×)	83.2	96.7

Table 1. Comparisons of performance of Swin transformers [3] with WS, and with both WS and MUX, and the original one on ImageNet-1K [1]. We share all layers in each stage. WS: Weight Sharing, MUX: Weight Multiplexing including both weight transformation and distillation.

Model	#Param. (M)	MACs (B)	Top-1 Acc (%)	Top-5 Acc (%)
DeiT-Ti [4]	5	1.3	72.2	91.1
DeiT-S [4]	22	4.6	79.9	95.0
DeiT-B [4]	86	17.6	81.8	95.6
DeiT-B↑384 [4]	87	55.6	82.9	96.2
Mini-DeiT-Ti **(ours)**	3(1.7×)	1.3	71.8(−0.4)	90.6
Mini-DeiT-S **(ours)**	11(2.0×)	4.6	80.0(+0.1)	95.0
Mini-DeiT-B **(ours)**	44(2.0×)	17.7	83.0(+1.2)	96.4
Mini-DeiT-B↑384 **(ours)**	44(2.0×)	56.8	84.5(+1.6)	97.1

Table 2. Comparisons of performance of DeiT [4] and Mini-DeiT. Note that we apply MiniViT to the original DeiTs without additional modifications

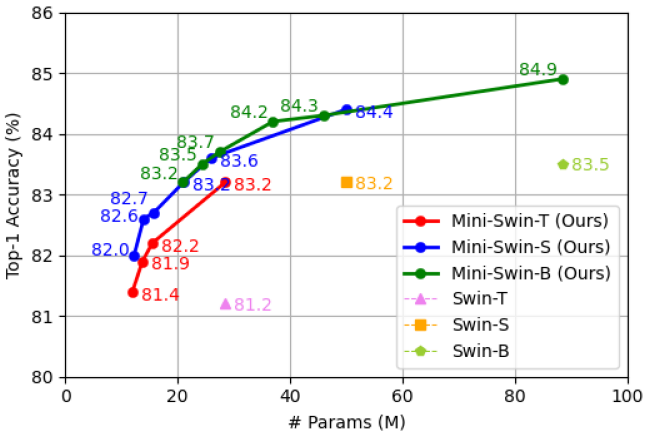

Figure 1. Effects of compression ratio on Mini-Swins. The percentages represent the compression ratio of Mini-Swins compared to the corresponding original Swin transformers. The accuracy of the original Swin-T, Swin-S and Swin-B is also shown on the right side of the points.

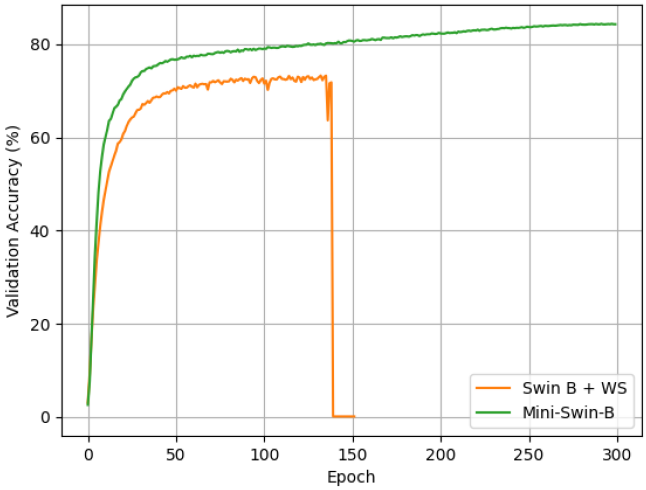

Figure 2. Training collapse of Swin-B with weight sharing.

formance degradation. However, similar to observations in [2], large models are more robust to compression compared to small models. Namely, Mini-Swin-B with compression ratio of 76% outperforms Mini-Swin-T without compression. Therefore, compressing large-scale pre-trained models is promising for the deployment on hardware with limited resources.

Plots for training instability. We visualize the training instability of weight sharing in Fig. 2. Applying weight sharing directly to Swin-B leads to a sharp performance drop in the training process. However, our proposed weight multiplexing method can not only solve the instability problem, but also improve the performance.

B. Broader Impacts

Similar to most previous compression works, our work does not have immediate societal impact, because the algorithm is only designed for image classification. However, it may indirectly impact society. As an example, our work may be applied or inspire the creation of new algorithms to areas with direct societal implications. Moreover, our method requires additional teacher models to guide the training process, which introduces potential privacy leakage. These issues warrant further research and consideration when using or building upon this work.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[2] Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joey Gonzalez. Train big, then compress: Rethinking model size for efficient training and inference of transformers. *ICML*, 2020. 2

[3] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 1

[4] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*. PMLR, 2021. 1