Supplementary Materials of SuperTickets: Drawing Task-Agnostic Lottery Tickets from Supernets via Jointly Architecture Searching and Parameter Pruning

Haoran You[1], Baopu Li[2,3], Zhanyi Sun[1], Xu Ouyang[1], and Yingyan Lin[1]

[1]Rice University [2]Baidu USA [3]Oracle Corporation
{haoran.you,zs19,xo2,yingyan.lin}@rice.edu, baopu.li@oracle.com

1 Visualization of The Adopted Supernet Architecture

We visualize the adopted supernet following [1] in Fig. 1. It begins with two 3×3 convolutions with stride 2, which are followed by five fusion modules and five parallel modules to gradually divide it into four branches of decreasing resolutions, the learned features from all branches are then merged together for classification or dense prediction.

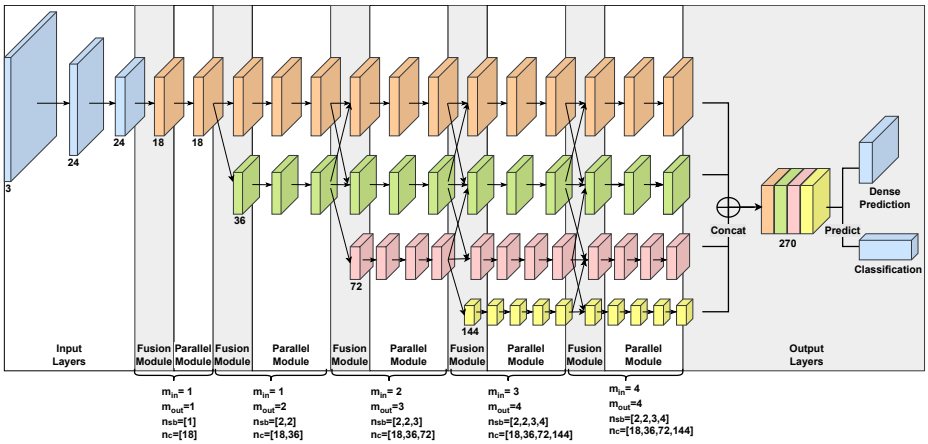

Fig. 1. Visualization of the adopted supernet architecture, where m_{in} and m_{out} denote the number of input and output branches in the fusion module; n_{sb} and n_c represent the number of searching blocks and channels in the parallel module, respectively.

Table 1. Comparing ST with "ST w/ RP" and "ST w/ RR-Init" on both Cityscapes and ADE20K under 80% and 90% sparsity.

Methods	FLOPs	Cityscapes ($p = 80\%$) mIoU	mAcc	aAcc	FLOPs	Cityscapes ($p = 90\%$) mIoU	mAcc	aAcc
S+P w/ RP	405M	1.30%	5.17%	21.96%	203M	1.15%	5.26%	21.9%
ST w/ RP	397M	20.17%	27.57%	68.73%	200M	16.55%	23.83%	65.90%
ST w/ RR-Init	397M	56.88%	67.33%	92.62%	200M	52.96%	62.66%	91.91%
ST	397M	**69.77%**	**79.76%**	**95.12%**	200M	**66.61%**	**76.30%**	**94.63%**
Methods	**FLOPs**	**ADE20K ($p = 80\%$) mIoU**	**mAcc**	**aAcc**	**FLOPs**	**ADE20K ($p = 90\%$) mIoU**	**mAcc**	**aAcc**
S+P w/ RP	308M	0.06%	0.66%	6.54%	154M	0.01%	0.66%	1.72%
ST w/ RP	317M	8.58%	11.93%	60.21%	159M	4.98%	7.19%	55.30%
ST w/ RR-Init	317M	21.24%	30.62%	68.57%	159M	19.49%	28.46%	67.26%
ST	317M	**31.19%**	**43.10%**	**74.82%**	159M	**27.82%**	**39.49%**	**73.37%**

Table 2. ST variants transfer validation tests under 90% sparsity.

Methods	ADE20K → Cityscapes mIoU	mAcc	aAcc	Methods	Cityscapes → ADE20K mIoU	mAcc	aAcc
ST w/ RP	10.51%	14.42%	61.28%	ST w/ RP	6.95%	10.1%	57.7%
ST w/ RR-Init	46.19%	54.92%	90.88%	ST w/ RR-Init	14.82%	21.02%	65.54%
ST	**62.91%**	**73.32%**	**93.82%**	ST	**20.83%**	**29.95%**	**69.00%**

2 SuperTickets (ST) vs. Random Pruning (RP) and Random Re-Initialization (RR-Init).

We compare the proposed SuperTickets (ST) with both the "ST w/ RP" and "ST w/ RR-Init" in Table 1. We consider two datasets under 80% and 90% sparsity: ST consistently outperforms the two baselines, achieving on-average 36.28%/42.03%/21.95% and 11.20%/12.27%/4.34% mIoU/mAcc/aAcc improvements over "ST w/ RP" and "ST w/ RR-Init", respectively, under a comparable number of parameters and FLOPs. These experiments show that SuperTickets performs better than both RP and RR-Init, which is consistent with the LTH finding.

Transferability of ST vs. RP and RR-Init. Similarly, we compare the transferability of the three ST variants when transferring them across different datasets, including (1) ADE20K → Cityscapes or (2) Cityscapes → ADE20K. As shown in Table 2, ST achieves on-average 33.14%/39.38%/21.92% and 11.37%/13.67%/3.20% mIoU/mAcc/aAcc improvements over the "ST w/ RP" and "ST w/ RR-Init" baselines, respectively, indicating that RP and RR-Init are inferior in transferability as compared to the proposed ST.

3 Clarification of the LTH Settings.

There are two confusing settings when talking about LTH: (1) directly test the accuracy of the found structure and the trained weights; and (2) the weights

Table 3. Comparing ST w/ various LTH settings (90% sparsity).

Methods	Cityscapes ($p = 90\%$) mIoU	mAcc	aAcc	Methods	ADE20K ($p = 90\%$) mIoU	mAcc	aAcc
ST w/ RR-Init	52.96%	62.66%	91.91%	ST w/ RR-Init	19.49%	28.46%	67.26%
ST w/ LT-Init	59.63%	70.24%	93.33%	ST w/ LT-Init	25.32%	36.76%	71.33%
ST w/ ELT-Init	65.82%	76.74%	94.54%	ST w/ ELT-Init	25.79%	37.33%	72.10%
ST w/ LLT-Init	67.17%	77.03%	94.73%	ST w/ LLT-Init	28.51%	40.63%	73.49%
ST w/o Retrain	66.61%	77.03%	94.73%	ST w/o Retrain	27.82%	39.49%	73.37%

are restored to their initial value and trained with the obtained mask to obtain test accuracy. We tried both of the aforementioned settings and find the former, i.e., directly testing the accuracy of the found structure and trained weights, has already achieved good results. This is another highlight of our work, as it can help to largely save the retraining time. Furthermore, to address your concern, we re-initialize the SuperTickets to (1) their initial values, following the origin LTH ("ST w/ LT-Init") and (2) *early* or (3) *late* stages following [2] ("ST w/ ELT-Init or LLT-Init"), and compare them with the RR-Init counterparts. From Table 3, we can see that **(1)** ST under all LTH settings achieves better accuracy than RR-Init, indicating the effectiveness of ST; **(2)** vanilla LT-Init underperforms both ELT-Init and LLT-Init under ST settings, consistent with [2]; and **(3)** ST w/ ELT-Init or LLT-Init achieves comparable or slightly better accuracy than ST w/o Retrain at a cost of retraining.

4 Speedups in terms of Inference Time

In addition to the number of parameters and FLOPs, we measure the inference FPS and speedups on both 1080Ti GPUs and a SOTA sparse DNN inference accelerator [4]. As shown in Table 4, ST achieves on par or even higher (i.e., 1.8×∼2.9× speedups) FPS on GPUs and much reduced accelerator time (i.e., 2.9×∼4.1× speedups) on [4] than the baselines, thanks to simultaneous architecture searching and parameter pruning (i.e., 2-in-1) and ST.

Table 4. ST vs. typical baselines on Cityscapes, in terms of inference time measured on both GPUs and sparse accelerators.

Model	Params	FLOPs	mIoU	GPU FPS	Sparse Acc. Time
BiSeNet	5.8M	6.6G	69.00%	105.8	180.8ms
DF1-Seg-d8	-	-	71.40%	136.9	181.7ms
FasterSeg	4.4M	-	71.50%	163.9	142.4ms
SqueezeNAS	0.73M	8.4G	72.40%	117.2	198.5ms
ST ($p = 50\%$)	0.63M	1.0G	**72.68%**	310.7	48.3ms

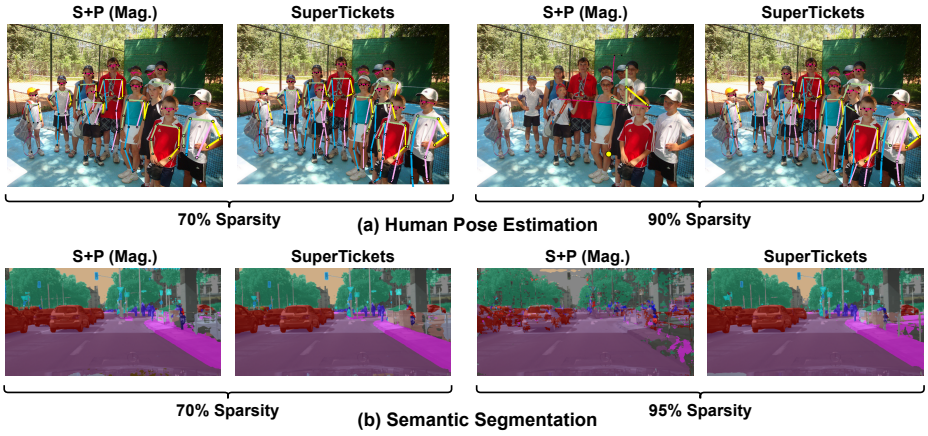

Fig. 2. Visualization of the human pose estimation on COCO keypoint dataset and the streetview/semantic labels on Cityscapes dataset under different pruning ratios.

5 Discussions

Limitations of Transferred SuperTickets. Although identified SuperTickets can transfer with only classifiers as task-specific, there is still a limitation in the transferred SuperTickets. That is, transferred SuperTickets cannot surpass those SuperTickets directly found on the target datasets/tasks. Moreover, when the sparsity is low (e.g., 30%), the transferred SuperTickets will underperform both SuperTickets and S+P. This is counterintuitive and opposite to the observation in compressing pretrained models [3], where low pruning ratios do not hurt the accuracy after transferring while overpruning leads to under-fitting. It implies that the dedicated search is necessary when pruning ratio is relatively low; while for high sparsity, the impacts of neural architectures will be less.

Visualization and Discussion. We visualize the results of SuperTickets and S+P baselines on COCO keypoint and Cityscapes datasets under different pruning ratios, as shown in Fig. 2. We observe that S+P baselines work but miss some keypoints or semantic understandings under medium sparsity (e.g., 70%) while collapse under high pruning ratios (e.g, 90/95%); In contrast, our identified SuperTickets consistently work well among a wide range of pruning ratios, validating the effectiveness of our proposed SuperTickets.

6 More Visualization of Visual Recognition Results

We further visualize the results of SuperTickets and S+P baselines on COCO keypoint and Cityscapes datasets under different pruning ratios, as shown in Fig. 3 and Fig. 4, respectively. We observe that S+P baselines work but miss some keypoints or semantic understandings under medium sparsity (e.g., 70/80%)

while collapse under high pruning ratios (e.g, 90/95%); In contrast, our identified SuperTickets consistently work well among a wide range of pruning ratios, validating the effectiveness of our proposed SuperTickets.

References

1. Ding, M., Lian, X., Yang, L., Wang, P., Jin, X., Lu, Z., Luo, P.: Hr-nas: Searching efficient high-resolution neural architectures with lightweight transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2982–2992 (2021)
2. Frankle, J., Dziugaite, G.K., Roy, D., Carbin, M.: Linear mode connectivity and the lottery ticket hypothesis. In: International Conference on Machine Learning. pp. 3259–3269. PMLR (2020)
3. Gordon, M.A., Duh, K., Andrews, N.: Compressing bert: Studying the effects of weight pruning on transfer learning. arXiv preprint arXiv:2002.08307 (2020)
4. Qin, E., Samajdar, A., Kwon, H., Nadella, V., Srinivasan, S., Das, D., Kaul, B., Krishna, T.: Sigma: A sparse and irregular gemm accelerator with flexible interconnects for dnn training. In: 2020 IEEE International Symposium on High Performance Computer Architecture (HPCA). pp. 58–70. IEEE (2020)

Fig. 3. Visualization of the human pose estimation on COCO keypoint dataset under various pruning ratios.

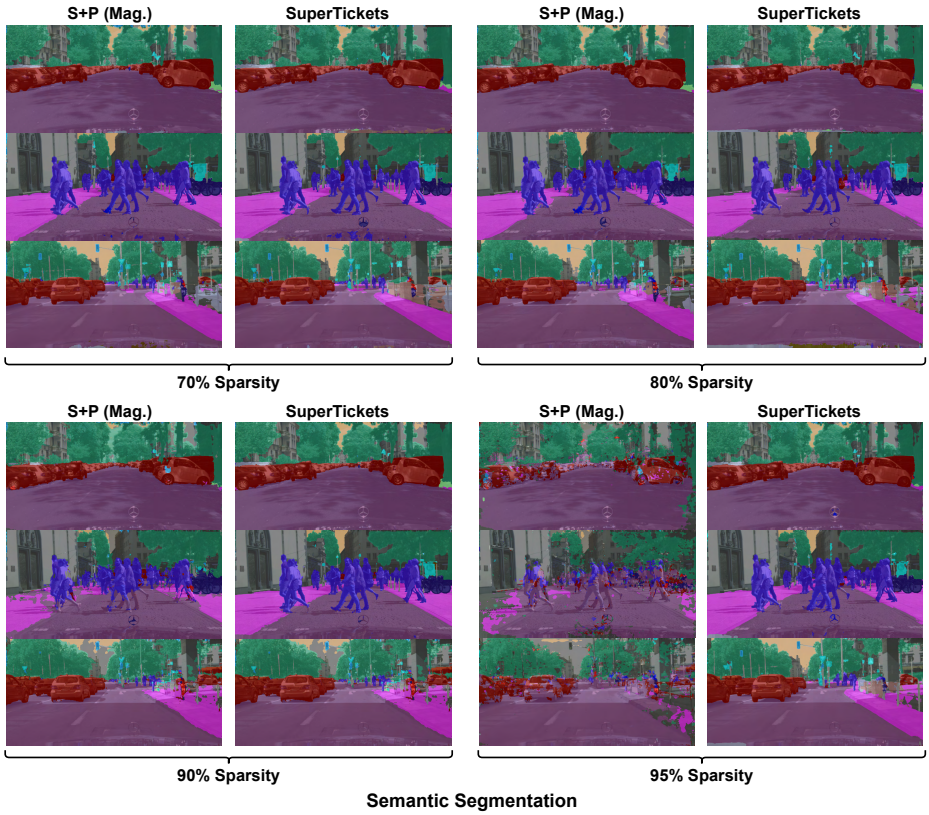

Fig. 4. Visualization of the streetview/semantic labels on Cityscapes dataset under various pruning ratios.