The Supplementary Materials of Early-Bird GCNs: Graph-Network Co-Optimization Towards More Efficient GCN Training and Inference via Drawing Early-Bird Lottery Tickets

Anonymous AAAI submission
Paper ID 6732

Experiment Setting

Models and Datasets. We evaluate the proposed methods over five representative GCN algorithms, i.e., GCN (Kipf and Welling 2017), GAT (Veličković et al. 2018), GIN (Xu et al. 2019), GraphSAGE (Hamilton, Ying, and Leskovec 2017), and 7/14/28-layer deep ResGCNs (Li et al. 2020a), on three citation graph datasets, i.e., Cora, CiteSeer, and Pubmed (Sen et al. 2008), two inductive datasets, i.e., PPI and Reddit (Hamilton, Ying, and Leskovec 2017), and two large-scale graph datasets from *Open Graph Benchmark (OGB)* (Weihua Hu 2020), i.e., Ogbn-ArXiv for node classification and Ogbl-Collab for link prediction. We follow the default setting in (Kipf and Welling 2017) and (Weihua Hu 2020) to split all the datasets. The train/test/validation split ratios for Ogbn-ArXiv and Ogbl-Collab are 54/18/28 and 92/4/4, respectively.

Training Settings. For the three citation graph datasets and two inductive graph datasets, we follow (Kipf and Welling 2017) to train all the chosen two-layer GCN models of 16 hidden units for 100 epochs using an Adam optimizer (Kingma and Ba 2014) with a learning rate of 0.01. For the graphs from OGB, we follow (Li et al. 2020a) to use full-batch training and test for Ogbn-ArXiv while applying mini-batch training for Ogbl-Collab by randomly partitioning the graph into 10 subgraphs and doing full-batch test. For both of these two datasets, the training/retraining takes 500 epochs using an Adam optimizer (Kingma and Ba 2014) with a learning rate of 0.01. A batch normalization and a dropout with a rate of 0.5 are also used for each layer of 128 hidden dimensions.

Baselines and Evaluation Metrics. We evaluate the effectiveness of the proposed GEBT's improved training and inference efficiency in terms of the node classification accuracy (or F1 Score, Hits@50), inference FLOPs, and total training FLOPs, as compared to other graph sparsifiers, i.e., random pruning (Frankle and Carbin 2019) and SGCN (Li et al. 2020b), and the standard SOTA GCN algorithms using unpruned graphs, including the GCN (Kipf and Welling 2017), GraphSAGE (Hamilton, Ying, and Leskovec 2017), GAT (Veličković et al. 2018) GIN (Xu et al. 2019), ClusterGCN (Chiang et al. 2019), FastGCN (Chen, Ma, and Xiao 2018), VRGCN (Chen, Zhu, and Song 2018), L2-GCN (You et al. 2020), GTTF (Markowitz et al. 2021), N-GCN (Abu-El-Haija et al. 2020), and 7/14/28-layer deep ResGCNs (Li et al. 2020a). Note that Hits@50 is only used to evaluate Ogbl-Collab since the corresponding task is to predict the future author collaboration relationships given the past collaborations, we then rank each true collaboration among a set of 100,000 randomly-sampled negative collaborations, and choose the ratio of positive edges that are ranked at K-place or above (Hits@K, K is 50 here) rather than the classification accuracy as the metric for evaluating the performance. **Note that all results are averaged over five runs.**

Table 1: GEBT vs. SOTA ResGCNs on OGB datasets.

ResGCNs # layers	Accuracy/Hits@50 (%)		Inference FLOPs (G)	
	Ogbn-ArXiv	Ogbl-Collab	Ogbn-ArXiv	Ogbl-Collab
7	71.3	53.1	18.56	24.39
14	72.6	52.9	39.03	52.60
28	72.1	53.4	79.96	109.01
GEBT (7)	71.2 (↓0.1)	52.8 (↓0.3)	2.69 (↑6.9×)	5.69 (↑4.3×)
GEBT (14)	72.3 (↓0.3)	52.7 (↓0.2)	9.27 (↑4.2×)	12.13 (↑4.3×)
GEBT (28)	72.7 (↑0.6)	53.5 (↑0.1)	18.92 (↑4.2×)	14.59 (↑7.5×)
Overall Improv.	**↓0.3 ∼ ↑0.6**		**↑4.21× ∼ ↑7.47×**	

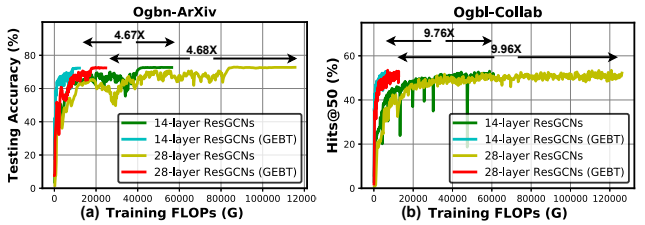

Figure 1: Testing performance's trajectories visualization for 14/28-layer ResGCNs w/ and w/o applying our GEBT versus training FLOPs when evaluated on Ogbn-ArXiv and Obgl-Collab.

Scale Up to Deep GCNs and Larger Graphs

As shown in Tab. 1, GEBT again consistently outperforms the baseline ResGCNs algorithms in terms of the inference efficiency-accuracy trade-offs. Specifically, GEBT achieves 4.21× ∼ 7.47× inference FLOPs reduction, while offering a comparable accuracy (↓0.3%∼↑0.6%), as compared to SOTA ResGCNs of various depths (e.g., 7/14/28-layer ResGCNs). Fig. 1 (a) and (b) further visualize the testing performance's trajectories of graph early-bird tickets training (GEBT) and baseline ResGCNs training versus the training

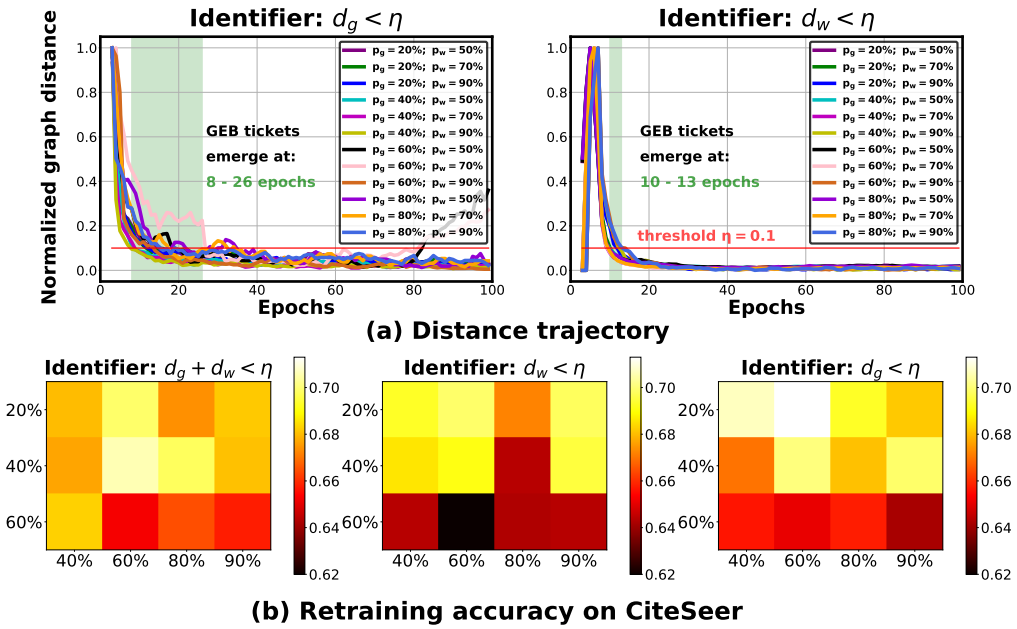

Figure 2: The (a) recorded distance's evolution along GCN training trajetories on the Cora dataset, and (b) retraining accuracies comparisons among different identifiers, where horizontal axis denotes the network sparisty and vertical axis indicates the graph sparsity, respectively.

FLOPs on Ogbn-ArXiv and Ogbl-Collab datasets, respectively. We can see that GEBT achieves a comparable accuracy with fewer training FLOPs, indicating its good generalization capability and the effectiveness of early-bird tickets and the proposed graph-network co-optimization.

Ablation Studies of Joint-EB Detectors

We adopt a mixture of "graph distance" and "network distance" to identify the joint-EB tickets in all the above experiments, and are further curious about whether a detector based on one of the two distances can still lead to similar benefits. To validate this, we first measure and compare the epoch ranges where joint-EB tickets emerge, and when applying another two criteria (1) $d_g < \eta$ and (2) $d_w < \eta$, respectively. We can see that the graph distance criterion is more suitable than the network distance criterion, because the latter undergoes a "warming up" process starting from 0 to 1 and then quickly drops to nearly zero, making joint-EB detection collapse to the initial training stage even if we ignore such "warming up" as shown in Fig. 2 (a). Then, we compare the retraining accuracy of the drawn joint-EB tickets using all the three criteria to see their robustness to different criteria (i.e., drawing epochs). The results in Fig. 2 (b) show that the third identifier $d_g + d_w < \eta$ works best as their overall retraining accuracy is the best across different pruning ratios.

References

Abu-El-Haija, S.; Kapoor, A.; Perozzi, B.; and Lee, J. 2020. N-gcn: Multi-scale graph convolution for semi-supervised node classification. In *uncertainty in artificial intelligence*, 841–851. PMLR.

Chen, J.; Ma, T.; and Xiao, C. 2018. Fastgcn: fast learning with graph convolutional networks via importance sampling. In *International Conference on Learning Representations (ICLR)*.

Chen, J.; Zhu, J.; and Song, L. 2018. Stochastic Training of Graph Convolutional Networks with Variance Reduction. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 942–950. PMLR.

Chiang, W.-L.; Liu, X.; Si, S.; Li, Y.; Bengio, S.; and Hsieh, C.-J. 2019. Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 257–266.

Frankle, J.; and Carbin, M. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations*.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, 1024–1034.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

Li, G.; Xiong, C.; Thabet, A.; and Ghanem, B. 2020a. Deepergcn: All you need to train deeper gcns. *arXiv preprint arXiv:2006.07739*.

Li, J.; Zhang, T.; Tian, H.; Jin, S.; Fardad, M.; and Zafarani, R. 2020b. SGCN: A Graph Sparsifier Based on Graph Convolutional Networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 275–287. Springer.

Markowitz, E.; Balasubramanian, K.; Mirtaheri, M.; Abu-El-Haija, S.; Perozzi, B.; Steeg, G. V.; and Galstyan, A. 2021. Graph Traversal with Tensor Functionals: A Meta-Algorithm for Scalable Learning. In *International Conference on Learning Representations (ICLR)*.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Weihua Hu, M. Z. Y. D. H. R. B. L. M. C. J. L., Matthias Fey. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *arXiv preprint arXiv:2005.00687*.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

You, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. L2-gcn: Layer-wise and learned efficient training of graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2127–2135.