TABLE VIII
RUNTIME STATISTICS ON LARGE DATASET OGBN-ARXIV. THE "PEAK GPU" MEANS THE MAX GPU USAGE DURING THE TRAINING PROCESS. "PE" REFERS TO PROXY EVALUATION.

Model	Model Selection		Training				Total	
	Time (s)	Peak GPU	Time		Peak GPU		Time (s)	Peak GPU
			Search	Train	Search	Train		
AutoHEnsGNN$_{\text{Adaptive}}$	12410	10.2G	511	8989	2.8G	2.6G	21910	10.2G
AutoHEnsGNN$_{\text{Gradient}}$	12410	10.2G	696	8121	6.9G	2.5G	21227	10.2G
D-Ensemble, L-Ensemble	12410	10.2G	-	10116	-	2.6G	22526	10.2G
Goyal *et al.* [45]	12410	10.2G	-	10116	-	2.6G	22526	10.2G
Ensemble + PE	12410	10.2G	-	3293	-	2.6G	14266	10.2G
Ensemble	52730	19.4G	-	-	-	-	52730	19.4G

D. Runtime Statistics

The experiments on Section IV-D show that a single model suffers from high variance and undesirable performance, which is discussed by Shchur et al. [17] and is named pitfalls of GNN evaluation. These pitfalls hinder practical usage in real-life scenarios. To obtain robust and accurate predictions, AutoHEnsGNN leverages hierarchical ensemble on a pool of effective models to reduce variance and improve accuracy. Compared with other ensemble methods, AutoHEnsGNN can achieve superior performance. In Table VIII, we further present the runtime statistics comparison between AutoHEnsGNN and other ensemble methods. "Ensemble" means the naive ensemble of all possible candidate models (20 models including spectral-based [11], [42], [57], [58] and spatial-based [12], [13], [34], [37], [39], [59]–[63] aggregators), attention aggregator [64], [65], skip connection [44], [46], [66], gate updater [43] and dynamic updater [67]). "Ensemble+PE" means the ensemble of the models in the pool selected by proxy evaluation. Other methods adopt the proxy evaluation and consider the variance of initialization in the ensemble. For example, in the training phase for L-ensemble, each kind of model (N models in the pool) is trained with K different initialization. Then L-ensemble learns the ensemble weights of the N$\times$K models (N=3, K=3 for ogbn-arxiv).

From Table VIII, 1) by comparing the "Ensemble" to other methods, we can find that proxy evaluation can greatly improve the training efficiency in terms of time and GPU memory; 2) "Ensemble+PE" obtains the lowest time cost since it does not run the models multiple times with different initialization. However, as discussed in Section IV-D and previous work [17], it usually cannot achieve good performance. For example, in Table V, "Ensemble+PE+GSE" can improve the score from 87.3 to 88.6 and reduce the variance from 0.8 to 0.3; 3) Other methods consume similar time and GPU memory, where AutoHEnsGNN$_{\text{Gradient}}$ uses slightly less time; 4) For AutoHEnsGNN$_{\text{Gradient}}$ and AutoHEnsGNN$_{\text{Adaptive}}$, although AutoHEnsGNN$_{\text{Gradient}}$ can leverage proxy model for memory reduction at the search stage, it still requires more GPU memory at the training stage. In all, the "Peak GPU" of AutoHEnsGNN$_{\text{Adaptive}}$ is the lower bound of that of AutoHEnsGNN$_{\text{Gradient}}$. For example, if we only have one model as the candidate, the total "Peak GPU" depends on that at the search stage. In this case, AutoHEnsGNN$_{\text{Gradient}}$ consumes more GPU memory than than AutoHEnsGNN$_{\text{Adaptive}}$. For scenarios with limited GPU memory, AutoHEnsGNN$_{\text{Adaptive}}$ may be a better choice. Otherwise, AutoHEnsGNN$_{\text{Gradient}}$ can be used to achieve better performance.

E. Model Comparison

We further add a kernel-based ensemble baseline MixCobra [81], a mixed strategy of Mojirsheibani *et al*. [82] and Biau *et al*. [83]. As shown in Table IX, AutoHEnsGNN still outperform the state-of-the-art single models and ensemble baselines by a large margin.

TABLE IX
RESULTS ON THE CORA, CITECEER AND PUBMED.

Method	Cora	Citeseer	Pubmed
GCN [11]	81.5	70.3	79.0
GAT [10]	83.0$\pm$0.7	72.5$\pm$0.7	79.0$\pm$0.3
APPNP [61]	83.8$\pm$0.3	71.6$\pm$0.5	79.7$\pm$0.3
Graph U-Net [71]	84.4$\pm$0.6	73.2$\pm$0.5	79.6$\pm$0.2
SGC [37]	82.0$\pm$0.0	71.9$\pm$0.1	78.9$\pm$0.0
MixHop [72]	81.9$\pm$0.4	71.4$\pm$0.8	80.8$\pm$0.6
GraphSAGE [12]	78.9$\pm$0.8	67.4$\pm$0.7	77.8$\pm$0.6
GraphMix [69]	83.9$\pm$0.6	74.5$\pm$0.6	81.0$\pm$0.6
GRAND [62]	85.4$\pm$0.4	75.5$\pm$0.4	82.7$\pm$0.6
GCNII [46]	85.5$\pm$0.5	73.4$\pm$0.6	80.2$\pm$0.4
D-ensemble	85.6$\pm$0.3	75.7$\pm$0.2	82.7$\pm$0.4
L-ensemble	85.9$\pm$0.2	76.0$\pm$0.2	82.9$\pm$0.1
Goyal *et al.* [45]	85.9$\pm$0.3	75.7$\pm$0.2	82.8$\pm$0.2
MixCobra [81]	85.3$\pm$0.7	75.5$\pm$0.7	82.3$\pm$0.4
AutoHEnsGNN$_{\text{Adaptive}}$	86.1$\pm$0.2	76.3$\pm$0.1	83.5$\pm$0.2
AutoHEnsGNN$_{\text{Gradient}}$	**86.5$\pm$0.2**	**76.9$\pm$0.2**	**84.0$\pm$0.1**

REFERENCES

[1] T. Derr, C. Wang, S. Wang, and J. Tang, "Relevance measurements in online signed social networks," in *Proceedings of the 14th international workshop on mining and learning with graphs (MLG)*, 2018.

[2] L. Liu, F. Zhu, L. Zhang, and S. Yang, "A probabilistic graphical model for topic and preference discovery on social media," *Neurocomputing*, vol. 95, pp. 78–88, 2012.

[3] C. C. Aggarwal, H. Wang *et al.*, *Managing and mining graph data*. Springer, vol. 40.

[4] J. Xu, J. Zhou, Y. Jia, J. Li, and X. Hui, "An adaptive master-slave regularized model for unexpected revenue prediction enhanced with alternative data," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 2020, pp. 601–612.

[5] Y. Li, U. Islambekov, C. Akcora, E. Smirnova, Y. R. Gel, and M. Kantarcioglu, "Dissecting ethereum blockchain analytics: What we learn from topology and geometry of the ethereum graph?" in *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 2020, pp. 523–531.

[6] N. C. Abay, C. G. Akcora, Y. R. Gel, M. Kantarcioglu, U. D. Islambekov, Y. Tian, and B. M. Thuraisingham, "Chainnet: Learning on blockchain graphs with topological features," in *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*, J. Wang, K. Shim, and X. Wu, Eds. IEEE, 2019, pp. 946–951. [Online]. Available: https://doi.org/10.1109/ICDM.2019.00105

[7] Z. Yang, W. Cohen, and R. Salakhudinov, "Revisiting semi-supervised learning with graph embeddings," in *International conference on machine learning*. PMLR, 2016, pp. 40–48.

[8] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," in *Advances in neural information processing systems*, 2020.

[9] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu, "Traffic flow prediction via spatial temporal graph neural network," in *Proceedings of The Web Conference 2020*, 2020, pp. 1082–1092.

[10] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

[11] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[12] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in neural information processing systems*, 2017, pp. 1024–1034.

[13] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *International Conference on Learning Representations*, 2018.

[14] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 974–983.

[15] T. Derr, Y. Ma, and J. Tang, "Signed graph convolutional networks," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 929–934.

[16] M. Zitnik, M. Agrawal, and J. Leskovec, "Modeling polypharmacy side effects with graph convolutional networks," *Bioinformatics*, vol. 34, no. 13, pp. i457–i466, 2018.

[17] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, "Pitfalls of graph neural network evaluation," *arXiv preprint arXiv:1811.05868*, 2018.

[18] Y. Gao, H. Yang, P. Zhang, C. Zhou, and Y. Hu, "Graph neural architecture search," in *IJCAI*, vol. 20, 2020, pp. 1403–1409.

[19] K. Zhou, Q. Song, X. Huang, and X. Hu, "Auto-gnn: Neural architecture search of graph neural networks," *arXiv preprint arXiv:1909.03184*, 2019.

[20] M. Nunes and G. L. Pappa, "Neural architecture search in graph neural networks," in *Brazilian Conference on Intelligent Systems*. Springer, 2020, pp. 302–317.

[21] H. Zhao, L. Wei, and Q. Yao, "Simplifying architecture search for graph neural network," *arXiv preprint arXiv:2008.11652*, 2020.

[22] Anonymous, "Efficient graph neural architecture search," in *Submitted to International Conference on Learning Representations*, 2021, under review. [Online]. Available: https://openreview.net/forum?id=IjIzIOkK2D6

[23] A. Pourchot, A. Ducarouge, and O. Sigaud, "To share or not to share: A comprehensive appraisal of weight-sharing," *arXiv preprint arXiv:2002.04289*, 2020.

[24] Y. Shu, W. Wang, and S. Cai, "Understanding architectures learnt by cell-based neural architecture search," in *International Conference on Learning Representations*, 2019.

[25] L. Xie, X. Chen, K. Bi, L. Wei, Y. Xu, Z. Chen, L. Wang, A. Xiao, J. Chang, X. Zhang *et al.*, "Weight-sharing neural architecture search:a battle to shrink the optimization gap," *arXiv preprint arXiv:2008.01475*, 2020.

[26] K. Yu, R. Ranftl, and M. Salzmann, "How to train your super-net: An analysis of training heuristics in weight-sharing nas," *arXiv preprint arXiv:2003.04276*, 2020.

[27] X. Wang, D. Kondratyuk, K. M. Kitani, Y. Movshovitz-Attias, and E. Eban, "Multiple networks are more efficient than one: Fast and accurate models via ensembles and cascades," *arXiv preprint arXiv:2012.01988*, 2020.

[28] H. Wang, W. Fan, P. S. Yu, and J. Han, "Mining concept-drifting data streams using ensemble classifiers," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 226–235.

[29] S. Puuronen, A. Tsymbal, and I. Skrypnyk, "Correlation-based and contextual merit-based ensemble feature selection," in *International Symposium on Intelligent Data Analysis*. Springer, 2001, pp. 135–144.

[30] S. Puuronen, I. Skrypnyk, and A. Tsymbal, "Ensemble feature selection based on the contextual merit," in *International Conference on Data Warehousing and Knowledge Discovery*. Springer, 2001, pp. 111–120.

[31] G. Anandalingam and T. L. Friesz, "Hierarchical optimization: An introduction," *Annals of Operations Research*, vol. 34, no. 1, pp. 1–11, 1992.

[32] B. Colson, P. Marcotte, and G. Savard, "An overview of bilevel optimization," *Annals of operations research*, vol. 153, no. 1, pp. 235–256, 2007.

[33] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," in *International Conference on Learning Representations*, 2018.

[34] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe, "Weisfeiler and leman go neural: Higher-order graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 4602–4609.

[35] Y. Liu, Y. Gu, Z. Ding, J. Gao, Z. Guo, Y. Bao, and W. Yan, "Decoupled graph convolution network for inferring substitutable and complementary items," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2621–2628.

[36] Z. Wang, W. Wei, G. Cong, X.-L. Li, X.-L. Mao, and M. Qiu, "Global context enhanced graph neural networks for session-based recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 169–178.

[37] F. Wu, A. H. Souza Jr, T. Zhang, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," in *ICML*, 2019.

[38] L. Zeng, J. Xu, Z. Yao, Y. Zhu, and J. Li, "Graph symbiosis learning," *arXiv preprint arXiv:2106.05455*, 2021.

[39] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "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*, 2019, pp. 257–266.

[40] V. P. Dwivedi, C. K. Joshi, T. Laurent, Y. Bengio, and X. Bresson, "Benchmarking graph neural networks," *arXiv preprint arXiv:2003.00982*, 2020.

[41] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *ICML*, 2017.

[42] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, pp. 3844–3852, 2016.

[43] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *arXiv preprint arXiv:1511.05493*, 2015.

[44] M. Fey, "Just jump: Dynamic neighborhood aggregation in graph neural networks," *arXiv preprint arXiv:1904.04849*, 2019.

[45] P. Goyal, D. Huang, S. R. Chhetri, A. Canedo, J. Shree, and E. Patterson, "Graph representation ensemble learning," *arXiv preprint arXiv:1909.02811*, 2019.

[46] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1725–1735.

[47] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowledge-Based Systems*, vol. 151, pp. 78–94, 2018.

[48] Y. Zhang, Z. Lin, J. Jiang, Q. Zhang, Y. Wang, H. Xue, C. Zhang, and Y. Yang, "Deeper insights into weight sharing in neural architecture search," *arXiv preprint arXiv:2001.01431*, 2020.

[49] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and systems magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[50] C. Zhao, Y. Qiu, S. Zhou, S. Liu, W. Zhang, and Y. Niu, "Graph embedding ensemble methods based on the heterogeneous network for lncrna-mirna interaction prediction," *BMC genomics*, vol. 21, no. 13, pp. 1–12, 2020.

[51] H. Ren, G. F. Kokai, W. J. Turner, and T.-S. Ku, "Paragraph: Layout parasitics and device parameter prediction using graph neural networks," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2020, pp. 1–6.

[52] Y. Kong and T. Yu, "forgenet: A graph deep neural network model using tree-based ensemble classifiers for feature extraction," *arXiv preprint arXiv:1905.09889*, 2019.

[53] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *ICML*, 2018.

[54] M. Liu, H. Gao, and S. Ji, "Towards deeper graph neural networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 338–348.

[55] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[56] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[57] J. Du, S. Zhang, G. Wu, J. M. Moura, and S. Kar, "Topology adaptive graph convolutional networks," *arXiv preprint arXiv:1710.10370*, 2017.

[58] F. M. Bianchi, D. Grattarola, L. Livi, and C. Alippi, "Graph neural networks with convolutional arma filters," *arXiv preprint arXiv:1901.01343*, 2019.

[59] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," *Advances in neural information processing systems*, vol. 28, pp. 2224–2232, 2015.

[60] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5115–5124.

[61] J. Klicpera, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," *arXiv preprint arXiv:1810.05997*, 2018.

[62] W. Feng, J. Zhang, Y. Dong, Y. Han, H. Luan, Q. Xu, Q. Yang, E. Kharlamov, and J. Tang, "Graph random neural networks for semi-supervised learning on graphs," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[63] T. Xie and J. C. Grossman, "Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties," *Physical review letters*, vol. 120, no. 14, p. 145301, 2018.

[64] K. K. Thekumparampil, C. Wang, S. Oh, and L.-J. Li, "Attention-based graph neural network for semi-supervised learning," *arXiv preprint arXiv:1803.03735*, 2018.

[65] E. Ranjan, S. Sanyal, and P. P. Talukdar, "Asap: Adaptive structure aware pooling for learning hierarchical graph representations." in *AAAI*, 2020, pp. 5470–5477.

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

[67] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[68] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[69] V. Verma, M. Qu, A. Lamb, Y. Bengio, J. Kannala, and J. Tang, "Graphmix: Regularized training of graph neural networks for semi-supervised learning," *arXiv preprint arXiv:1909.11715*, 2019.

[70] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[71] H. Gao and S. Ji, "Graph u-nets," in *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[72] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. Ver Steeg, and A. Galstyan, "Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *International Conference on Machine Learning*, 2019, pp. 21–29.

[73] Q. Huang, H. He, A. Singh, S.-N. Lim, and A. Benson, "Combining label propagation and simple models out-performs graph neural networks," in *International Conference on Learning Representations*, 2020.

[74] Y. Wang, J. Jin, W. Zhang, Y. Yu, Z. Zhang, and D. Wipf, "Bag of tricks for node classification with graph neural networks," *arXiv preprint arXiv:2103.13355*, 2021.

[75] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[76] F. Frasca, E. Rossi, D. Eynard, B. Chamberlain, M. Bronstein, and F. Monti, "Sign: Scalable inception graph neural networks," in *ICML 2020 Workshop on Graph Representation Learning and Beyond*, 2020.

[77] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," *arXiv preprint arXiv:1803.07294*, 2018.

[78] G. Li, M. Müller, B. Ghanem, and V. Koltun, "Training graph neural networks with 1000 layers," *arXiv preprint arXiv:2106.07476*, 2021.

[79] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.

[80] F. Errica, M. Podda, D. Bacciu, and A. Micheli, "A fair comparison of graph neural networks for graph classification," in *International Conference on Learning Representations*, 2019.

[81] A. Fischer and M. Mougeot, "Aggregation using input–output trade-off," *Journal of Statistical Planning and Inference*, vol. 200, pp. 1–19, 2019.

[82] M. Mojirsheibani, "Combining classifiers via discretization," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 600–609, 1999.

[83] G. Biau, A. Fischer, B. Guedj, and J. D. Malley, "Cobra: A combined regression strategy," *Journal of Multivariate Analysis*, vol. 146, pp. 18–28, 2016.