Learning from Graphs Beyond Message Passing Neural Networks

Tong Zhao, Neil Shah, Elham Ghazizadeh
Snap Inc., USA
{tong,nshah,eghazizadeh}@snap.com

Abstract

Graph, as a potent data structure, models complex relational data that are ubiquitous in real-world applications like social networks and recommendation systems. In the past few years, message passing-based Graph Neural Networks (GNNs) have emerged as standard tools for direct learning from graph data. However, such direct integration during training also introduces challenges, including scalability issues with large-scale graphs and oversmoothing problems with increased model parameter sizes via additional layers. This study offers a bird's-eye view of high-level paradigms for learning from graph data, categorizing techniques into three distinct classes: (1) using graph structure during preprocessing, (2) using graph structure during training, and (3) using graph structure at test-time inference. Through this overview, we aim to illuminate diverse approaches and advantages inherent in learning from graph data across these fundamental paradigms.

1 Introduction and Background

Graph machine learning (GML) methods mainly refer to data-driven techniques that learn from the omnipresent graph-structured data. From shallow graph embedding methods (Perozzi et al., 2014; Grover & Leskovec, 2016; Wang et al., 2016; Cai et al., 2018), to message passing-based GNNs (Kipf & Welling, 2016; Hamilton et al., 2017; Veličković et al., 2017; Wu et al., 2020), and more recently, graph transformers (Min et al., 2022; Kong et al., 2023; Dwivedi et al., 2023), these methods essentially learn representations from graph data for downstream tasks.

With their capability to directly model graph structure via neighborhood aggregation during training, message passing-based GNNs and graph transformers achieve state-of-the-art performances across many graph-related tasks. However, such design on explicitly using graph structure to define model architecture not only brings benefits on performances but also introduces unique challenges. Firstly, using graph structure during both forward and backward passes in training is known to be expensive compared with non-graph models (Han et al., 2022). Furthermore, when the graph size exceeds the capacity of a single GPU's memory, mini-batching becomes necessary to make training feasible. Although neighbor sampling (Hamilton et al., 2017) can effectively alleviate the neighbor explosion issue, this graph-based sampling differs from existing batching solutions developed for tabular machine learning in other domains, such as CV and NLP. Consequently, special frameworks involving significant engineering efforts, such as GraphStorm (Zhang et al., 2023) and TF-GNN (Ferludin et al., 2022), have been developed for industrial-scale usage of GNNs.

Additionally, in other machine learning domains, following the neural scaling law (Kaplan et al., 2020), increasing the number of parameters by stacking more layers and increasing the hidden dimension size generally yields better performance (Cheng et al., 2016; OpenAI, 2023; Touvron et al., 2023). However, with the neighborhood aggregation design, it is not intuitive to scale up GNNs and graph transformers in terms of parameter sizes by adding more layers. When GNNs are deep, oversmoothing becomes an inevitable issue (Zhao & Akoglu, 2019; Zhao et al., 2021), which is essentially caused by the heavy overlap of k-hop computational subgraphs for different nodes when k becomes large, making node representations less separable in the latent space. It's also worth mentioning that even for Graph Transformers, the optimal numbers of Transformer layers and attention heads are usually relatively small numbers (Chen et al., 2022; Kong et al., 2023).

Despite the aforementioned drawbacks, message passing-based GML methods continue to receive significant attention from the community and achieve state-of-the-art performance across graph tasks and applications. Nevertheless, not all GML techniques follow the same learning paradigm, and the graph structure's usage in other steps of the learning process is gradually gaining more attention, especially with the recent success of large language models (LLMs). Despite there are already several existing surveys on GML, in this short article, we aim to provide a brief overview and highlight the less-explored GML paradigms with their advantages and draw the community's attention to them.

2 Three Graph Machine Learning Paradigms

We begin by defining the most common learning paradigm that involves utilizing graph structures during training. Specifically, any method that necessitates the use of graph structures during back-propagation is categorized as **using graphs during training**, which inherently encompasses all GNNs and the majority of graph transformers (e.g., GOAT (Kong et al., 2023)). Note that this also includes methods that does not have learnable parameters in the message-passing components, such as APPNP (Gasteiger et al., 2018) and LightGCN (He et al., 2020). Despite not having learnable parameters in message passing, these methods still require the graph structure for the model training.

With the above definition of using graphs during training, the definitions of the other two paradigms naturally emerge as using graphs during preprocessing (prior to model training) and using graphs at test-time inference (after model training), respectively. We elaborate on these below.

Using graphs during preprocessing. The concept of using graphs during preprocessing can be traced back to the beginning of shallow graph embedding, exemplified by DeepWalk (Perozzi et al., 2014), which essentially transforms the graph structure into random walk sequences and then learns node embeddings via Word2Vec (Mikolov et al., 2013). Methods falling under this paradigm convert the graph structure into other tabular formats that can be consumed by models (e.g., MLP, Language Models) that were not explicitly designed for GML (Wu et al., 2019; Dwivedi et al., 2023). For instance, SIGN (Frasca et al., 2020) uses local graph operators to preprocess the graph data and a MLP as the model, achieving significant scalability improvements over GNNs without sacrificing performances. Additionally, LLaGA (Chen et al., 2024) uses structure-aware graph translation and alignment tuning to encode graph data into tokens that LLMs can directly consume, hence benefiting LLMs' strong generalization and zero-shot capabilities. In short, methods in this category typically turn the graph data into tabular-formatted data, hence the graph structure is no-longer necessary for training and inference, as the transformed data already contains all the needed information. Therefore, this category of methods allows flexibility in using other ML models that are not specifically designed for graph data, and benefiting from their advantages such as scalability and zero-shot capability.

Using graphs in test-time inferencing. The strategy of using graph structure only during test-time inferencing has historical roots, with the classic Label Propagation graph algorithm (Zhu & Ghahramani, 2002). Label propagation assumes that any connected node pairs are more likely to have the same label and hence iteratively propagates node labels along the edges. More recently, researchers found this philosophy can be generalized to modern GML techniques via applying neighborhood aggregation only at test-time (Bojchevski et al., 2020). For instance, Yang et al. (2022) proposed PMLP, which is trained as an MLP but used as a GNN at test-time only. Similarly, Ju et al. (2024) proposed TAG-CF that applies a single layer of parameter-less neighborhood aggregation at test-time on top of traditional two-tower collaborative filtering models, resulting with significant performance improvements. By training none message passing models and only apply message passing once at test time, these methods essentially only add minimal computational overhead that's equivalent to one forward pass of a message passing-based GNN. Therefore, methods belonging to this category usually can achieve substantial efficiency improvements over GNN-based methods, while also achieving non-marginal performance improvements over the non-graph methods.

3 Conclusion

This paper provides bird's-eye view of GML paradigms in term of where to make use of the graph structure information. By categorizing methods in to these three distinct classes, we highlight the challenges of the most commonly used message passing-based methods, and most importantly the advantages of the less-explored two: using graphs during preprocessing offers versatility in integrating advanced ML techniques, while using graphs solely at test-time inference demonstrates efficiency gains over training-centric methods. This thousand-feet view aims to draw more attention from researchers and practitioners in this evolving landscape of GML paradigms, fostering a deeper understanding and encouraging further exploration in this dynamic field.

URM Statement

The authors acknowledge that all authors of this paper meet the URM criteria[1] of ICLR 2024 Tiny Papers Track.

References

Aleksandar Bojchevski, Johannes Gasteiger, Bryan Perozzi, Amol Kapoor, Martin Blais, Benedek Rózemberczki, Michal Lukasik, and Stephan Günnemann. Scaling graph neural networks with approximate pagerank. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2464–2473, 2020.

Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE transactions on knowledge and data engineering*, 30(9):1616–1637, 2018.

Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. Nagphormer: A tokenized graph transformer for node classification in large graphs. In *The Eleventh International Conference on Learning Representations*, 2022.

Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. Llaga: Large language and graph assistant. *arXiv preprint arXiv:2402.08170*, 2024.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pp. 7–10, 2016.

Vijay Prakash Dwivedi, Yozen Liu, Anh Tuan Luu, Xavier Bresson, Neil Shah, and Tong Zhao. Graph transformers for large graphs. *arXiv preprint arXiv:2312.11109*, 2023.

Oleksandr Ferludin, Arno Eigenwillig, Martin Blais, Dustin Zelle, Jan Pfeifer, Alvaro Sanchez-Gonzalez, Wai Lok Sibon Li, Sami Abu-El-Haija, Peter Battaglia, Neslihan Bulut, et al. Tf-gnn: Graph neural networks in tensorflow. *arXiv preprint arXiv:2207.03522*, 2022.

Fabrizio Frasca, Emanuele Rossi, Davide Eynard, Ben Chamberlain, Michael Bronstein, and Federico Monti. Sign: Scalable inception graph neural networks. *arXiv preprint arXiv:2004.11198*, 2020.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Xiaotian Han, Tong Zhao, Yozen Liu, Xia Hu, and Neil Shah. Mlpinit: Embarrassingly simple gnn training acceleration with mlp initialization. In *The Eleventh International Conference on Learning Representations*, 2022.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 639–648, 2020.

Mingxuan Ju, William Shiao, Zhichun Guo, Yanfang Ye, Yozen Liu, Neil Shah, and Tong Zhao. How does message passing improve collaborative filtering? *arXiv preprint arXiv:2404.08660*, 2024.

[1] https://iclr.cc/Conferences/2023/CallForTinyPapers

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Kezhi Kong, Jiuhai Chen, John Kirchenbauer, Renkun Ni, C Bayan Bruss, and Tom Goldstein. Goat: A global transformer on large-scale graphs. In *International Conference on Machine Learning*, pp. 17375–17390. PMLR, 2023.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Erxue Min, Runfa Chen, Yatao Bian, Tingyang Xu, Kangfei Zhao, Wenbing Huang, Peilin Zhao, Junzhou Huang, Sophia Ananiadou, and Yu Rong. Transformer for graphs: An overview from architecture perspective. *arXiv preprint arXiv:2202.08455*, 2022.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1225–1234, 2016.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pp. 6861–6871. PMLR, 2019.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Chenxiao Yang, Qitian Wu, Jiahua Wang, and Junchi Yan. Graph neural networks are inherently good generalizers: Insights by bridging gnns and mlps. *arXiv preprint arXiv:2212.09034*, 2022.

Jian Zhang, Da Zheng, Xiang Song, Theodore Vasiloudis, Israt Nisa, and Jim Lu. Graphstorm an easy-to-use and scalable graph neural network framework: From beginners to heroes. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5790–5791, 2023.

Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns. *arXiv preprint arXiv:1909.12223*, 2019.

Tong Zhao, Yozen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. Data augmentation for graph neural networks. In *Proceedings of the aaai conference on artificial intelligence*, volume 35, pp. 11015–11023, 2021.

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. *ProQuest Number: INFORMATION TO ALL USERS*, 2002.