Graph Foundation Models: State of the Art and Future Directions

Alexander Zhou
The Hong Kong Polytechic University
alexander.zhou@polyu.edu.hk

Haoyang Li
The Hong Kong Polytechic University
haoyang-comp.li@polyu.edu.hk

Lei Chen
HKUST
HKUST (GZ)
leichen@cse.ust.hk

ABSTRACT

Graph Foundation Models (GFMs) have recently emerged as a promising direction for learning reusable knowledge from graph-structured data. Unlike conventional graph neural networks (GNNs), which are usually trained for a specific graph and task, GFMs aim to transfer graph representations across graphs, tasks, and feature modalities. However, the established literature is rapidly fragmenting in different directions. In this tutorial, we divide current GFM research into five predominant families and explore them appropriately. We first introduce the motivation for GFMs and the high-level taxonomy that organizes the area. We then discuss five representative families: LLM-Enhanced GNNs, GNN-Enhanced LLMs, Unified Graph Models, Graph Mixture-of-Experts, and Graph Prompt Learning illustrating their individual strengths which underpin their current design. Finally, we outline future research directions from a data management perspective. This tutorial will help researchers and practitioners from the database and data management community understand the major implementation paths and open problems in Graph Foundation Models.

PVLDB Reference Format:
Alexander Zhou, Haoyang Li, and Lei Chen. Graph Foundation Models: State of the Art and Future Directions. PVLDB, 19(12): 4918 - 4921, 2026.
doi:10.14778/3827998.3828150

1 INTRODUCTION

Graph-structured data is widely encountered across many real-world domains, including social networks, recommendation systems, molecular analysis, and knowledge graphs [17]. Graph Neural Networks (GNNs) have become a standard tool for learning from such data, achieving strong performance on node classification, link prediction, and graph classification [8, 13]. Despite their success, conventional GNNs are usually designed and trained for a specific graph and downstream task. When the graph domain, feature modality, or prediction objective changes, practitioners often need to redesign the model, collect new supervision, and retrain from scratch [19]. Inspired by foundation models and large language models (LLMs) [1], recent research has therefore turned to Graph Foundation Models (GFMs) [31]. Unlike conventional GNNs, which are typically trained for a specific graph task, GFMs are intended to support adaptation across graph types, domains, and downstream tasks. However, this goal is challenging because graph data is highly heterogeneous in structure, feature modality, and task form, making it difficult to learn transferable graph knowledge through a single modeling pipeline.

As research on GFMs continues to grow rapidly, developing a clear understanding of current methods becomes increasingly important. At a high level, existing GFM approaches realize transferable graph modeling through substantially different pipelines, integrating graph encoders, prompting mechanisms, unified architectures, or language models in different ways. This growing methodological diversity makes the field increasingly difficult to organize and compare. As a result, the community still lacks a unified and systematic understanding of the GFM design space, comparative strengths, and applicable conditions, which makes a tutorial both timely and necessary.

In this tutorial, we aim to provide a comprehensive overview of recent efforts on Graph Foundation Models. Over the past two years, research in this area has grown rapidly, with more than fifty papers appearing at venues such as NeurIPS, ICML, ICLR, KDD, WWW, SIGMOD, and VLDB. To organize this rapidly expanding landscape, we propose a comprehensive taxonomy of current GFM implementation paths, viewing existing methods through the lens of how transferable graph modeling is realized. Specifically, we distinguish two major paradigms, namely LLM-Integrated GFMs and GNN-Native GFMs, and further discuss five representative families under these paradigms. For each family, we explain its core pipeline, main mechanism, and adaptation strategy, providing a structured way to understand and compare current GFM methods. Beyond surveying modeling techniques, we highlight open data-centric challenges such as scalable graph construction and cross-domain integration, and discuss how core data management tasks can benefit from transferable graph models, thereby bridging the graph learning and database communities.

2 TUTORIAL OUTLINE

We propose a 1.5-hour tutorial, organized into the following sections and major technical components.

- **Introduction and taxonomy overview (15 mins):** We introduce graphs, GNNs, the motivation for GFMs, and the high-level taxonomy (Table 1) that organizes the tutorial.
- **The Five Families of GFMs**
  - **LLM-Enhanced GNNs (10 mins)**
  - **GNN-Enhanced LLMs (10 mins)**
  - **Unified Graph Models (15 mins)**
  - **Graph Mixture-of-Experts (15 mins)**
  - **Graph Prompt Learning (10 mins)**
- **Future research directions (15 mins)**

We present a general outline of content expected in each section:

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 19, No. 12 ISSN 2150-8097.
doi:10.14778/3827998.3828150

Table 1: Summary of the taxonomy of graph foundation models.

Paradigm	Family	Realization Paths and Representative Models
LLM-Integrated GFMs	**LLM-Enhanced GNNs**	Text Augmentation (TAPE [9], KEA [3], SimTeG [4]). Direct Embedding (ENGINE [38], OFA [14], UniGraph [10]).
	GNN-Enhanced LLMs	Graph-to-Text (NLGraph [26], GraphText [37], InstructGraph [28]). Graph-to-Token Projection (GraphGPT [25], LLaGA [2], Graph2Token [30]).
GNN-Native GFMs	**Unified Graph Models**	Feature-Level Alignment (OpenGraph [35], LEDA [18]). Structure-Level Alignment (RiemannGFM [22], GFT [32], ULTRA [6]). Task-Level Alignment (GIT [33], PRODIGY [12], SCR [29]).
	Graph Mixture-of-Experts	Component-Level Composition (Graph MoE [27], SAMGPT [36]). Model-Level Composition (AnyGraph [34], OMOG [15]).
	Graph Prompt Learning	Appended Node Features (GPF/GPF-Plus [5], GraphPrompt [16]). Inserted Subgraph (All-in-One [24], GPPT [23], PSP [7]).

2.1 LLM-Enhanced GNNs

LLM-Enhanced GNNs keep a GNN as the downstream learner and use LLMs only to improve textual node or edge attributes beforehand. Here the LLM serves as a semantic front-end rather than the final predictor. This design is natural on text-attributed graphs, and existing methods fall into the following categories:

Text Augmentation: Methods that use a LLM to generate additional text and then encode that as graph features. The generated text can plays different roles, including explanations, domain terms or task-relevant descriptions that are absent from the original graph. TAPE [9] uses explanation-style generations, converting LLM-produced explanations into node features before GNN training. KEA [3] augments nodes with technical terms and descriptions and SimTeG [4] further explores this direction by adapting the text encoder on graph corpora and then re-encoding node text with the adapted language model.

Direct Embedding: Methods that use the LLM directly as a semantic encoder, skipping any text generation. ENGINE [38] freezes the LLM and relies on trainable GNN ladders to adapt text embeddings to graph structure. OFA [14] uses shared text encodings for nodes, edges, and task descriptions so that different classification tasks can be handled in one interface. UniGraph [10] applies the same embedding-first pipeline to cross-domain pre-training and transfer on text-attributed graphs. The same pipeline also applies when relation labels or graph descriptions are textualized.

2.2 GNN-Enhanced LLMs

GNN-Enhanced LLMs use graph encoders to make graph structure consumable by an LLM. The main issue is how to map non-sequential graph inputs into the text or token interface expected by the LLM, and existing methods can be categorized as follows:

Graph-to-Text: Methods that serialize the graph into natural language and let the LLM reason over the resulting prompt. NLGraph [26] represents the direct verbalization route by describing graph structure and task context in text. GraphText [37] introduces syntax-aware graph verbalization. InstructGraph [28] standardizes graph descriptions with code-like formats for graph-centric instruction tuning. In practice, this route often requires long or dense graphs to be aggressively summarized before prompting.

Graph-to-Token Projection: Methods that first encode the graph with a GNN and then project graph representations into the token space of the LLM. GraphGPT [25] uses a projector to align GNN-produced graph features with graph tokens. LLaGA [2] first reorganizes the graph into structured node sequences and then projects them into token embeddings. Graph2Token [30] pushes this direction further by aligning graph tokens with the LLM vocabulary without tuning the LLM backbone, and the same idea also extends naturally to multimodal graphs. Compared with graph-to-text prompting, token projection avoids rewriting the whole graph into text but requires a stable alignment layer between graph and language representations.

2.3 Unified Graph Models

Unified Graph Models attempt to reuse one graph backbone across different domains and tasks without explicit expert decomposition. Their central strategy is to align heterogeneous graphs into a shared modeling interface before prediction, and existing methods can be categorized as follows:

Feature-Level Alignment: Methods that map heterogeneous node or edge attributes into a shared latent space and then apply a common graph backbone. OpenGraph [35] combines feature projection with contrastive pre-training so that graphs from different domains can be processed in one representation space. LEDA [18] makes this alignment more explicit by learning a shared latent distribution across domains through variational alignment, rather than only matching projected vectors.

Structure-Level Alignment: Methods that align graphs through shared structural primitives rather than raw attributes. RiemannGFM [22] samples trees and cycles as canonical structures, while GFT [32] extracts computation trees as a discrete structural vocabulary. ULTRA [6] applies the same idea to knowledge graphs by learning relation-level structures that can be reused across different relation spaces. In all three cases, the backbone no longer operates on domain-specific adjacency patterns alone, but on aligned structural units that can transfer across graphs.

Task-Level Alignment: Methods that reformulate node, link and graph-level prediction into a common interface. GIT [33] merges task-relevant computation trees through a virtual node and performs prediction on that shared target. PRODIGY [12] instead constructs prompt graphs whose label nodes are read out after message passing. SCR [29] recasts different tasks as task-specific knowledge-graph structures for a unified reasoning interface. The common goal is to avoid separate task-specific heads by translating different prediction targets into one reusable reasoning pipeline.

2.4 Graph Mixture-of-Experts

Graph Mixture-of-Experts methods maintain multiple specialized modules and use routing to combine them for each graph or task. Instead of forcing one backbone to handle all settings equally well, this family decomposes transfer into expert specialization plus expert selection. The key design choice is whether routing mixes small functional modules or selects full expert backbones, and existing methods can be categorized as follows:

Component-Level Composition: Methods that combine lightweight modules rather than full graph backbones, usually keeping most of the backbone shared and only specialize a small part of the pipeline (such as feature alignment, aggregation, or scaling). Graph MoE [27] routes among lightweight aggregation experts, while SAMGPT [36] composes domain-specific scaling weights through two-stage routing.

Model-Level Composition: Methods that route each input to complete expert models. AnyGraph [34] performs hard routing over multiple experts for each input graph. OMOG [15] assigns one GNN expert to each source graph and combines them through top-k gating. Because each expert keeps its own encoder and inductive bias, this route better preserves graph-specific modeling choices, albeit at higher cost than component-level composition.

2.5 Graph Prompt Learning

Graph Prompt Learning adapts a pre-trained graph backbone with a small prompt module instead of updating all model parameters. The key idea is to reformulate a downstream task so that it better matches the inductive bias of the pre-trained graph model, thereby enabling efficient adaptation without full fine-tuning. Existing methods can be grouped into the two categories:

Appended Node Features: Methods in which the prompt is implemented as additional learnable features attached to existing nodes, so the original graph topology is unchanged. GPF [5] uses a shared prompt vector for all nodes. GPF-Plus [5] introduces multiple prompt vectors and selects them adaptively. GraphPrompt [16] further shows that prompting can also be injected into hidden representations. This design is parameter-efficient and simple to deploy, but its gains are sensitive to prompt design and source-target mismatch.

Inserted Subgraph: Methods that introduce prompt nodes or prompt subgraphs and connect them to the original graph, so the prompt can modify both features and structure. All-in-One [24] learns prompt nodes together with their internal and cross-graph connections, while GPPT [23] uses task or class nodes to recast downstream prediction on the augmented graph. PSP [7] further injects structural cues into the prompted graph to improve adaptation. Inserted-subgraph prompting is more expressive than feature-only prompting, but it also brings extra graph-construction overhead and can become brittle on large or highly mismatched graphs.

2.6 Future Research Directions

Scalable Data Construction for GFMs: A key foundation of successful foundation models is training data scale. For GFMs, however, progress remains constrained by the lack of large-scale, high-quality, and reusable graph data for pre-training. Future research should therefore focus on scalable graph data construction, including richer graph corpora and meaningful synthetic graph generation, so that GFMs can learn from broader semantic and structural coverage.

Domain-Specific GFMs: Current GFMs still suffer from negative transfer under domain mismatch, which suggests that one universal model may not be the most practical path today. A more realistic direction is to develop domain-specific GFMs that combine shared graph backbones with domain-specific experts, adapters, or retrieval modules. Such designs may better balance reusable graph priors and specialized domain knowledge.

Graph Tokenization: Current GFMs still lack a native way to convert graphs into reusable pre-training units. Future research should therefore develop graph tokenization mechanisms that preserve topology, node and edge types, attributes, and recurring higher-order patterns, rather than flattening graphs into text-like sequences. Better graph tokenization can further support graph-native objectives such as masked subgraph completion and joint modeling of topology and attributes.

Robust and Efficient GFMs: Beyond clean benchmarks, practical GFMs must remain reliable under noisy structure, class imbalance, distribution shift, and realistic resource constraints. At the same time, many current methods, especially those integrating LLMs or large expert modules, remain expensive in both time and memory. Future work should therefore pursue uncertainty-aware prediction, corruption-resistant graph modeling, imbalance-aware adaptation, and lightweight yet expressive architectures.

3 ADDITIONAL INFORMATION

Intended Audience The tutorial is intended for researchers and practitioners interested in graph learning, foundation models, large language models for graphs, and transferable learning over graph-structured data. We aim to make the tutorial largely digestible to a general data management audience looking to understand recent advancements in the space of GFMs.

Related tutorials. There are tutorials on large language models for graphs [11], which primarily emphasize the interaction between LLMs and graph learning. Another closely related tutorial is Towards Graph Foundation Models [21]. Compared with these tutorials, our tutorial places stronger emphasis on a family-driven organization of the GFM landscape, covering the major implementation paths of current GFMs, discussing their limitations and future directions from a data management perspective. In VLDB 2024, the presenters hosted a tutorial titled "Efficient Training of Graph Neural Networks on Large Graphs" [20].

4 PRESENTERS

(1) **Alexander Zhou** is an Assistant Professor (Research) in the Department of Computing at The Hong Kong Polytechnic University, having received his PhD from The Hong Kong University of Science and Technology. His research interests include graph structures, algorithms, learning and foundation models with a particular focus on social network analysis. He has published more than 20 publication in top venues.

(2) **Haoyang Li** is an Assistant Professor (Research) in the Department of Computing at The Hong Kong Polytechnic University having received his PhD from The Hong Kong University of Science and Technology. His research interests include graph foundation

models, large language models, and data management for machine learning. He has published more than 30 papers in top venues.

(3) **Lei Chen** is a Chair Professor in the Data Science and Analytics Thrust and Artificial Intelligence Thrust at The Hong Kong University of Science and Technology (Guangzhou), where he also serves as Dean of the Information Hub. He is a Fellow of the ACM and IEEE, and his research interests include data-driven AI, knowledge graphs, and query optimization on large graphs and probabilistic databases. He received the ACM SIGMOD Test-of-Time Award in 2015, the VLDB Best Research Paper Award in 2022 and the VLDB Excellent Demonstration Award in 2014. He was the General Chair of VLDB 2024 in Guangzhou and currently serves as President of the VLDB Endowment for the 2026–2027 term and Editor-in-Chief of IEEE Transactions on Knowledge and Data Engineering.

REFERENCES

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems* 33 (2020), 1877–1901.

[2] Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. 2024. Llaga: Large language and graph assistant. *Proceedings of the International Conference on Machine Learning* (2024), 7809–7823.

[3] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. 2024. Exploring the potential of large language models (llms) in learning on graphs. *ACM SIGKDD Explorations Newsletter* 25, 2 (2024), 42–61.

[4] Keyu Duan, Qian Liu, Tat-Seng Chua, Shuicheng Yan, Wei Tsang Ooi, Qizhe Xie, and Junxian He. 2023. Simteg: A frustratingly simple approach improves textual graph learning. *arXiv preprint arXiv:2308.02565* (2023).

[5] Taoran Fang, Yunchao Zhang, Yang Yang, Chunping Wang, and Lei Chen. 2023. Universal prompt tuning for graph neural networks. *Advances in Neural Information Processing Systems* 36 (2023), 52464–52489.

[6] Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. 2024. Towards Foundation Models for Knowledge Graph Reasoning. *International Conference on Learning Representations* (2024), 31598–31619.

[7] Qingqing Ge, Zeyuan Zhao, Yiding Liu, Anfeng Cheng, Xiang Li, Shuaiqiang Wang, and Dawei Yin. 2024. PSP: Pre-Training and Structure Prompt Tuning for Graph Neural Networks. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (2024), 423–424.

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

[9] Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. 2024. Harnessing Explanations: LLM-to-LM Interpreter for Enhanced Text-Attributed Graph Representation Learning. *International Conference on Learning Representations* (2024).

[10] Yufei He, Yuan Sui, Xiaoxin He, and Bryan Hooi. 2025. Unigraph: Learning a unified cross-domain foundation model for text-attributed graphs. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2025).

[11] Chao Huang, Xubin Ren, Jiabin Tang, Dawei Yin, and Nitesh Chawla. 2024. Large Language Models for Graphs: Progresses and Directions. *Companion Proceedings of the ACM Web Conference* (2024), 1284–1287.

[12] Qian Huang, Hongyu Ren, Peng Chen, Gregor Kržmanc, Daniel Zeng, Percy S Liang, and Jure Leskovec. 2023. Prodigy: Enabling in-context learning over graphs. *Advances in Neural Information Processing Systems* 36 (2023), 16302–16317.

[13] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations* (2016).

[14] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. 2024. One for All: Towards Training One Graph Model for All Classification Tasks. *International Conference on Learning Representations* (2024).

[15] Jingzhe Liu, Haitao Mao, Zhikai Chen, Bingheng Li, Wenqi Fan, Mingxuan Ju, Tong Zhao, Neil Shah, and Jiliang Tang. 2025. One Model for One Graph: A New Perspective for Pretraining with Cross-domain Graphs. *arXiv preprint arXiv:2412.00315* (2025).

[16] Zemin Liu, Xingtong Yu, Yuan Fang, and Xinming Zhang. 2023. GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks. *Proceedings of the ACM Web Conference* (2023), 417–428.

[17] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93–93.

[18] Lianze Shan, Jitao Zhao, Dongxiao He, Siqi Liu, Jiaxu Cui, and Weixiong Zhang. 2026. LEDA: Latent Semantic Distribution Alignment for Multi-domain Graph Pre-training. *Proceedings of the ACM Web Conference* (2026), 614 – 625.

[19] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868* (2018).

[20] Yanyan Shen, Lei Chen, Jingzhi Fang, Xin Zhang, Shihong Gao, and Hongbo Yin. 2024. Efficient training of graph neural networks on large graphs. *Proceedings of the VLDB Endowment* 17, 12 (2024), 4237–4240.

[21] Chuan Shi, Cheng Yang, Yuan Fang, Lichao Sun, and Philip S. Yu. 2024. Lecture-style Tutorial: Towards Graph Foundation Models. *Companion Proceedings of the ACM Web Conference* (2024), 1264–1267.

[22] Li Sun, Zhenhao Huang, Suyang Zhou, Qiqi Wan, Hao Peng, and Philip Yu. 2025. Riemanngfm: Learning a graph foundation model from riemannian geometry. *Proceedings of the ACM on Web Conference* (2025), 1154–1165.

[23] Mingchen Sun, Kaixiong Zhou, Xin He, Ying Wang, and Xin Wang. 2022. Gppt: Graph pre-training and prompt tuning to generalize graph neural networks. *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022), 1717–1727.

[24] Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. 2023. All in One: Multi-Task Prompting for Graph Neural Networks. *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2023), 2120–2131.

[25] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2024. GraphGPT: Graph Instruction Tuning for Large Language Models. *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval* (2024), 491–500.

[26] Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2023. Can language models solve graph problems in natural language? *Advances in Neural Information Processing Systems* 36 (2023), 30840–30861.

[27] Haotao Wang, Ziyu Jiang, Yuning You, Yan Han, Gaowen Liu, Jayanth Srinivasa, Ramana Kompella, Zhangyang Wang, et al. 2023. Graph mixture of experts: Learning on large-scale graphs with explicit diversity modeling. *Advances in neural information processing systems* 36 (2023), 50825–50837.

[28] Jianing Wang, Junda Wu, Yupeng Hou, Yao Liu, Ming Gao, and Julian McAuley. 2024. InstructGraph: Boosting Large Language Models via Graph-centric Instruction Tuning and Preference Alignment. *Findings of the Association for Computational Linguistics* (2024), 13492 – 13510.

[29] Kai Wang, Siqiang Luo, Caihua Shan, and Yifei Shen. 2024. Towards Graph Foundation Models: Training on Knowledge Graphs Enables Transferability to General Graphs. *Advances in Neural Information Processing Systems* 38 (2024).

[30] Runze Wang, Mingqi Yang, and Yanming Shen. 2025. Bridging Molecular Graphs and Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence* (2025), 21234–21242.

[31] Zehong Wang, Zheyuan Liu, Tianyi Ma, Jiazheng Li, Zheyuan Zhang, Xingbo Fu, Yiyang Li, Zhengqing Yuan, Wei Song, Yijun Ma, Qingkai Zeng, Xiusi Chen, Jianan Zhao, Jundong Li, Meng Jiang, Pietro Lio, Nitesh Chawla, Chuxu Zhang, and Yanfang Ye. 2025. Graph Foundation Models: A Comprehensive Survey. *arXiv preprint arXiv:2505.15116* (2025).

[32] Zehong Wang, Zheyuan Zhang, Nitesh Chawla, Chuxu Zhang, and Yanfang Ye. 2024. Gft: Graph foundation model with transferable tree vocabulary. *Advances in neural information processing systems* 37 (2024), 107403–107443.

[33] Zehong Wang, Zheyuan Zhang, Tianyi Ma, Nitesh V Chawla, Chuxu Zhang, and Yanfang Ye. 2025. Towards Graph Foundation Models: Learning Generalities Across Graphs via Task-Trees. *Proceedings of the International Conference on Machine Learning* (2025), 65518–65555.

[34] Lianghao Xia and Chao Huang. 2024. AnyGraph: Graph Foundation Model in the Wild. *Findings of the Association for Computational Linguistics* (2024), 882 – 896.

[35] Lianghao Xia, Ben Kao, and Chao Huang. 2024. Opengraph: Towards open graph foundation models. *Findings of the Association for Computational Linguistics* (2024), 2365–2379.

[36] Xingtong Yu, Zechuan Gong, Chang Zhou, Yuan Fang, and Hui Zhang. 2025. Samgpt: Text-free graph foundation model for multi-domain pre-training and cross-domain adaptation. In *Proceedings of the ACM on Web Conference*. 1142–1153.

[37] Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, Kai Liu, Michael Bronstein, Zhaocheng Zhu, and Jian Tang. 2023. GraphText: Graph Reasoning in Text Space. *arXiv preprint arXiv:2310.01089* (2023).

[38] Yun Zhu, Yaoke Wang, Haizhou Shi, and Siliang Tang. 2024. Efficient Tuning and Inference for Large Language Models on Textual Graphs. *Proceedings of the International Joint Conference on Artificial Intelligence* (2024), 5734 – 5742.