Benchmarking Graph Foundation Models

Jinyu Yang
Beijing University of Posts and Telecommunications
Beijing, China
jinyu.yang@bupt.edu.cn

Liangwei Yang
University of Illinois Chicago
Chicago, United States
lyang84@uic.edu

Zeyuan Guo
Beijing University of Posts and Telecommunications
Beijing, China
1154459434@bupt.edu.cn

Jiayi Gao
Beijing University of Posts and Telecommunications
Beijing, China
jiayigao@bupt.edu.cn

Jing Wu
Beijing University of Posts and Telecommunications
Beijing, China
luoxc007@bupt.edu.cn

Tianhao Chai
Beijing University of Posts and Telecommunications
Beijing, China
chai1098518666@bupt.edu.cn

Hai Huang
Beijing University of Posts and Telecommunications
Beijing, China
hhuang@bupt.edu.cn

Cheng Yang
Beijing University of Posts and Telecommunications
Beijing, China
yangcheng@bupt.edu.cn

Chuan Shi*
Beijing University of Posts and Telecommunications
Beijing, China
shichuan@bupt.edu.cn

Abstract

In real-world applications, graph data has garnered significant attention for its representation and analysis using Graph Neural Networks. Recent advancements have led to the development of Graph Foundation Models (GFMs), which aim to enhance cross-domain and cross-task generalization ability. Despite promising results from GFMs, a lack of standardized evaluation processes hinders comparative analysis and cross-domain applicability. To address this gap, we propose GFMBench, an open-source pipeline that standardizes the training, evaluation, and deployment of GFMs across diverse real-world graph applications. GFMBench integrates state-of-the-art GFMs and datasets, providing a modular design for comprehensive support across data preprocessing, model training, and evaluation. The pipeline includes a robust evaluation framework for benchmarking GFM generalization ability, encompassing supervised learning, cross-domain zero-shot and few-shot learning, and in-context learning. To validate the usability of GFMs, we deploy them on the Open Academic Graph, enabling applications such as topic search and author recommendation. This work provides a unified benchmark for GFMs, enabling deeper insights into their generalization ability across various graph tasks and domains. We further open-source GFMBench [1] and related documents [2].

CCS Concepts

• **Computing methodologies** → **Neural networks**.

Keywords

Graph Neural Networks, Graph Foundation Models, Academic Social Networks, Frameworks

ACM Reference Format:
Jinyu Yang, Liangwei Yang, Zeyuan Guo, Jiayi Gao, Jing Wu, Tianhao Chai, Hai Huang, Cheng Yang, and Chuan Shi. 2025. Benchmarking Graph Foundation Models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3711896.3737410

1 Introduction

Many real-world data are naturally presented in a graph structure, comprising a set of nodes and edges, such as social networks [10, 34], citation networks [15], knowledge graphs [2], recommender systems [7, 38, 39], and biological networks [21]. Graph Neural Networks (GNNs) [10, 15, 32] are an effective graph representation learning method for modeling graph-structured data. However, the end-to-end training paradigm limits the model's generalization ability across various datasets from domains and different graph tasks [5, 18]. With the success of Large Language Models (LLMs) [1, 28], Graph Foundation Models (GFMs) have been proposed to enable cross-domain and cross-task graph learning. A graph foundation model [4, 19] is expected to benefit from the pre-training of broad graph data and can be adapted to a wide range of downstream graph tasks. Although GFMs have shown significant power under various experimental settings on real-world datasets [18, 25, 26], a comprehensive evaluation pipeline is lacking to benchmark their generalization ability. With the recent efforts of researchers, many graph foundation models have been developed [3, 18, 24]. However, adopting data processing techniques

[*] Corresponding author
[1] https://github.com/BUPT-GAMMA/ggfm
[2] https://ggfm.readthedocs.io/en/latest/

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '25, Toronto, ON, Canada.

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737410

and evaluation settings across different GFMs impedes a comprehensive understanding of their relative capabilities [17]. To achieve generalization across domains and graph tasks, some GFM frameworks include LLMs and require customized input/output for LLMs' training [25]. For example, HiGPT [26] designs tailored heterogeneous instructions for pre-training the LLM, thereby facilitating the model to achieve heterogeneous relation awareness. Moreover, different GFMs frequently employ distinct experimental settings and evaluation metrics. For instance, although both LLaGA [3] and OFA [18] evaluate models on the link prediction task of the PubMed [36] dataset, OFA utilizes ROC AUC as the evaluation metric, while LLaGA relies on accuracy. This discrepancy complicates quantitative comparisons among researchers. Consequently, we are supposed to standardize the training and evaluation processes of graph foundation models.

To address these challenges, we design and construct the Graph Foundation Model Benchmark (GFMBench), an open-source pipeline for building and evaluating GFMs in real-world graph applications. GFMBench standardizes the training and evaluation processes of GFMs, integrating 10 state-of-the-art models and 10 datasets from diverse domains within a modularized design. The pipeline provides comprehensive support across various stages, including data preprocessing, model training, and evaluation. For example, the data module within GFMBench offers interfaces for tasks such as instruction generation and corpus construction, which are essential for training large language models (LLMs). Its modularized design and detailed documentation enable users to easily customize sub-modules and integrate new components into graph foundation models. To benchmark the generalization ability of GFMs, we design a variety of evaluation tasks under different settings, such as supervised learning, cross-domain zero-shot and few-shot learning, and in-context learning. These settings apply to node classification and link prediction tasks across different graphs. This diverse evaluation framework allows for a more comprehensive understanding of GFM performance, helping researchers assess the models' generalization ability across different domains and tasks.

To validate the usability of GFMs, we further implement a Scientific Literature Analysis system for real-world GFM deployment based on Open Academic Graph (OAG) [27, 41]. It contains a rich set of node and edge types, making it an ideal dataset for testing GFMs' performance. We implement various relational applications [16] such as topic search, similar author recommendation, and research interest prediction. Based on graph foundation models, we can leverage the abundant structural and attribute information within the graph while enabling a single model to support multiple downstream applications. Furthermore, we provide visualizations of the inference results for different GFMs across the same applications, which can help users better understand each model's performance and strengths in different graph tasks. Our contributions can be summarized as follows:

- We create GFMBench, an open-source pipeline that standardizes the training and evaluation of GFMs, integrating state-of-the-art models and diverse datasets.
- Evaluation Framework: We design a comprehensive evaluation framework in GFMBench to test GFM's generalization ability, covering various settings like supervised learning, cross-domain zero-shot, and few-shot learning.
- Real-World GFM Deployment: We deploy GFMs on the Open Academic Graph (OAG) dataset for applications like topic search and author recommendation, providing interactive visualizations of model performance.

2 Related Work

Graph Neural Networks (GNNs). GNNs have emerged as a powerful paradigm for learning from graph-structured data, enabling effective representation learning through message passing. GCN [15] introduced spectral-based convolutional operations to aggregate neighborhood information, significantly improving node classification tasks. GAT [29] further enhanced representation learning by incorporating attention mechanisms to assign different importance weights to neighbors. Meanwhile, GIN [33] was designed to be as expressive as the Weisfeiler-Lehman graph isomorphism test, improving graph-level representation capabilities. While these models excel in semi-supervised learning, self-supervised GNN pretraining has emerged as a promising direction to improve generalization [35]. Notable methods include GraphCL [40], which employs contrastive learning to obtain transferable representations, and GCC [23], which pre-trains GNNs on large-scale graphs to capture universal structural patterns. These advancements have laid the groundwork for GFMs, empowering pre-trained GNNs to generalize across diverse datasets and downstream tasks.

Graph Foundation Models (GFMs). The emergence of Large Language Models (LLM) has sparked researchers' interest in GFMs [19]. Currently, the GFMs can be categorized into three types. Graph neural network (GNN)-based GFMs [13, 23, 40] pre-train GNNs on a variety of graphs and tasks to achieve generalization ability. For example, PT-HGNN [14] designs pre-training tasks at both the node and schema levels to contrastively retain diverse semantic and structural properties, enabling transferable knowledge for a variety of downstream tasks. The second category is LLM-based methods. These methods address the challenge of aligning graph data with natural language through two primary approaches. The graph-to-token approach, pioneered by GIMLET [43], treats node representations as unique tokens for transformer-based models or open-source LLMs [3]. The graph-to-text approach [24] describes graph structures using natural language, for instance, WalkLM [24] integrates language models and random walks to generate unsupervised graph representations, achieving superior performance in diverse downstream tasks. Hybrid methods [18, 42, 44] combine GNNs and LLMs in different ways. Notable examples include GraphGPT [25] and HiGPT [26], which enhance LLM capabilities with graph structural knowledge.

Besides the research trend in GFMs, we provide the first comprehensive benchmark for GFMs to evaluate current methods on the same page. GFMBench provides carefully designed datasets, integrated pre-training/fine-tuning scripts as well as comprehensive evaluate settings, making it the most suitable testbed for GFMs.

3 Graph Foundation Models Benchmark (GFMBench)

This section introduces GFMBench. The overall architecture is illustrated in Figure 1. GFMBench is built on PyTorch [22] and is

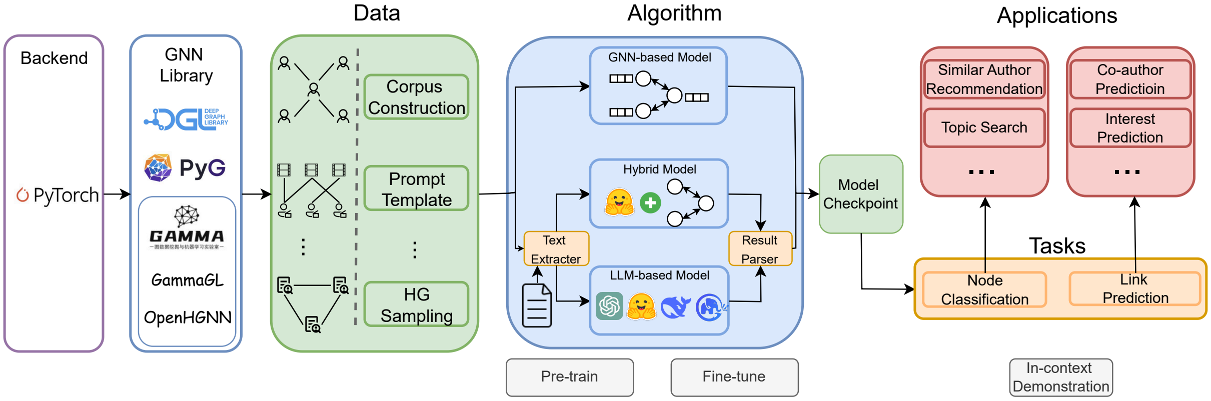


Figure 1: Framework of GFMBench. Our benchmark mainly consists of Data, Algorithm, Tasks and Applications modules. Within the algorithm module, we can pre-train and fine-tune GFMs. We can also provide in-context demonstration within the tasks module during inference. GFMBench's backend is PyTorch, supported by DGL and PyG graph learning libraries. Data is stored in MySQL, and we also build websites as the application layer for academic graph analysis supporting applications as Topic Search, Similar Author Recommendation, etc.

compatible with several mainstream graph learning libraries, including DGL [31], PyG [8], OpenHGNN [11], and GammaGL [20]. The pipeline mainly consists of three core modules: Data, Algorithm, and Tasks. In addition, we have developed an interactive online application platform for comparing GFMs in practice, as introduced in Section 5.

3.1 Data

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Each node $v \in \mathcal{V}$ and edge $e \in \mathcal{E}$ may have an associated attribute. The attribute of node v is denoted as $\mathbf{a}_v$, which can be either a feature vector $\mathbf{a}_v \in \mathbb{R}^d$ or a textual description $\mathbf{a}_v \in \mathcal{T}$, where $\mathcal{T}$ represents the set of all possible texts. Similarly, the attribute of edge e is denoted as $\mathbf{a}_e$, which can also be either a feature vector $\mathbf{a}_e \in \mathbb{R}^d$ or a textual description $\mathbf{a}_e \in \mathcal{T}$. Nodes and edges can have different types. We denote the type of node v as $\tau(v) \in \mathcal{A}$, where $\mathcal{A}$ is the set of possible node types in the graph. Similarly, the type of edge $e \in \mathcal{E}$ is denoted as $\tau(e) \in \mathcal{R}$, where $\mathcal{R}$ is the set of possible edge types in the graph.

In this work, we have collected a diverse and large-scale set of graphs to facilitate the pre-training and fine-tuning of GFMs. A summary of the dataset is shown in Table 1. The datasets vary significantly in both size and domain, ranging from homogeneous to heterogeneous graphs, with some datasets containing over a billion nodes, such as MAG and AMiner. This diversity ensures that GFMs can learn generalizable patterns applicable to different real-world contexts. To ensure high-quality and standardized preprocessing, we implement various data processing techniques. For large-scale datasets, we provide HGSampling [13], an efficient sampling method for handling massive graphs. Additionally, for models involving LLMs, we introduce a unified prompt design along with corpus construction methods, ensuring consistency in how textual information is leveraged. For a fair and standardized evaluation, we carefully partition each dataset into fixed training, validation, and test splits with predefined node and edge types. This setup ensures consistency across different GFMs, allowing direct performance comparisons under the same evaluation conditions. Furthermore, users can customize datasets according to their needs, with an example provided in our documentation website.

3.2 Models

GFMs are pre-trained on a set of graphs $\mathcal{G}_{\text{train}} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_N\}$, where each graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ is drawn from a potentially diverse set of domains. The goal of pre-training is to learn generalized graph representations by capturing structural and relational patterns that can be transferred across different graphs. During pre-training, the model leverages node and edge features, along with node and edge types, to learn a shared embedding space that can generalize to various graph-related tasks. The pre-training process typically utilizes large-scale graph datasets, enabling the model to learn high-level abstractions that are independent of specific graph instances.

Once pre-training is completed, the GFM $\mathcal{M}_{\text{pre-train}}$ can be applied to a target graph $\mathcal{G}_{\text{test}} = (\mathcal{V}_{\text{test}}, \mathcal{E}_{\text{test}})$, which may come from a different domain than the pre-training graphs. The model can either be directly used for downstream tasks, such as node classification, link prediction, or community detection, or undergo a fine-tuning stage to adapt the learned representations to the specifics of the target graph. fine-tuning involves adjusting the model's parameters based on the test graph's data, often with supervision or other task-specific signals, to enhance its performance for the given graph task. The key advantage of GFMs is their ability to generalize across different graphs, tasks, and domains, allowing a single model to be reused for a wide variety of graph-based applications. As shown in Figure 1, current GFMs can be classified into three categories based on their backbone structure:

- GNN-based Models (PT-HGNN [14], GPT-GNN [13]): These models pre-train a shared GNN across multiple graphs, aiming to achieve generalization through message passing mechanism across diverse graph structures.

Graph	#Nodes	#Edges	#N Types	#E Types	Fine-tuning Nodes				Fine-tuning Edges			
					Type	#Train	#Valid	#Test	Type	#Train	#Valid	#Test
OAG [41]	1,119,637	14,322,988	5	16	Paper	437,363	54,671	54,670	Paper-Field	3,755,582	469,448	469,448
MAG [30]	244,160,499	1,728,364,232	3	3	Paper	9,7401,333	12,175,167	12,175,167	Paper-Author	617,636,352	77,204,544	77,204,544
DBLP [9]	26,128	479,132	4	12	Author	3,245	406	406	Author-Paper	31,432	3,929	3,929
Cora [15]	2,708	10,498	1	2	Paper	2,166	271	271	Paper-Paper	8,686	1,085	1,087
Citeseer [15]	3,312	6,596	1	2	Paper	2,649	331	332	Paper-Paper	7,571	946	947
AMiner [6]	9,043,633	70,898,698	5	12	Paper	1,908,737	238,592	238,593	Paper-Author	8,308,796	1,038,599	1,038,601
IMDB [32]	11,616	34,212	3	4	Movie	3,422	428	428	Actor-Movie	20,524	2,565	2,567
LastFM [12]	31,860	378,664	3	6	Artist	14,417	1,802	1,803	Artist-User	148,534	18,567	18.567
Yelp [36]	82,465	32,548,358	4	8	Business	5,979	747	748	Business-Location	11,958	1,494	1,496
PubMed [24]	63,109	472,916	4	20	Disease	16,130	2,017	2,016	Disease-Species	8,392	1,049	1,049

Table 1: Benchmark data summary. GFMBench supports 10 graphs of varied sizes and domains. For each graph, we fix one node/edge type for fine-tuning and build a train/valid/test set for the selected type to unify GFM's performance. Datasets are from different domains: Academic , Entertainment , Recommender System , Medicine .

- LLM-based Models (WalkLM [24], LLaGA [3]): These models focus on leveraging textual information associated with graphs to facilitate generalization, emphasizing the role of graph-related text.
- Hybrid Models (OFA [18], GraphGPT [25], HiGPT [26], Graph-Translator [42], LMCH [37]): Drawing inspiration from multimodal paradigms, hybrid models treat text and graph as distinct modalities, aligning them to enhance performance.

3.3 Tasks

GFMBench incorporates two widely used tasks for evaluating GFMs: *Node Classification* and *Link Prediction*. Based on our Online Application Platform for academic network analysis introduced in Section 5, we prioritize NC and LP, which are essential for author profiling and research collaboration discovery. In future versions of GFMBench, we plan to extend the benchmark to encompass a more comprehensive range of tasks, such as graph-level tasks.

3.3.1 *Node Classification.*

Node classification aims to predict the label of a node based on its features and its relationships with other nodes in the graph. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ denotes the set of edges, the task is to predict the class label $y_v \in \mathcal{Y}$ for a node $v \in \mathcal{V}$. The node's label is typically determined by the graph structure and the features associated with the node. The goal is to learn a model $f : \mathcal{V} \rightarrow \mathcal{Y}$ that minimizes the classification error on a given set of labeled nodes.

$$y_v = f(\mathbf{a}_v, \mathcal{G}),$$

where $\mathbf{a}_v$ is the feature vector (or text description) of node v, and $\mathcal{G}$ is the graph structure to incorporate structure information.

3.3.2 *Link Prediction.*

Link prediction seeks to predict the likelihood of the existence of an edge between two nodes, which is essential for tasks such as recommender systems and social network analysis. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the task is to predict whether an edge exists between two nodes u and v, where $u, v \in \mathcal{V}$, $u \neq v$. Formally, the link prediction task can be defined as predicting the existence of an edge between nodes u and v, denoted as $e_{uv} \in \mathcal{E}$. The model outputs a score $\hat{y}_{uv}$ representing the probability that an edge exists between nodes u and v:

$$\hat{y}_{uv} = g(\mathbf{a}_u, \mathbf{a}_v, \mathcal{G}),$$

where g is a function that takes as input the feature vectors $\mathbf{a}_u$ and $\mathbf{a}_v$ of nodes u and v, and the graph structure $\mathcal{G}$. The output $\hat{y}_{uv}$ represents the probability of an edge existing between u and v.

Both of these tasks are fundamental in evaluating the generalization ability of GFMs. In our benchmark, we apply these tasks to all datasets for pre-training or evaluation, enabling the comparison of model performance across various graph structures and tasks. By leveraging both node classification and link prediction, we provide a comprehensive evaluation framework that tests the adaptability and robustness of GFMs on a wide range of graph-based tasks.

3.4 Evaluation Setting

To thoroughly evaluate the generalization ability of GFMs, we define several evaluation settings with varying difficulty levels. These settings assess the model's performance in different scenarios and measure its adaptability to new graphs and tasks. Specifically, we consider four evaluation settings: *Supervised Learning*, *Cross-domain Zero-shot Learning*, *Few-shot Learning*, and *In-context Learning*. Each setting tests distinct capabilities of GFMs, as described below. Further implementation details are available in our code repository, where we will also release additional experiments in future updates, including efficiency analyses.

3.4.1 *Supervised Learning.*

In the supervised learning setting, the model is pre-trained on a single graph and subsequently evaluated across different tasks on the same graph, thereby measuring its cross-task generalization capability. Formally, let $\mathcal{G}_{\text{train}}$ be the graph used for pre-training. The model $\mathcal{M}_{\text{pre-train}}$ is pre-trained to minimize the error on the graph $\mathcal{G}_{\text{train}}$:

$$\mathcal{L}_{\text{pre-train}} = \mathcal{L}(\mathcal{M}_{\text{pre-train}}(\mathcal{G}_{\text{train}}), \mathcal{Y}_{\text{train}}),$$

where $\mathcal{Y}_{\text{train}}$ is the true label associated with graph $\mathcal{G}_{\text{train}}$. For evaluation, the model's performance is inferred on the same set of pre-training graphs with different tasks:

$$\mathcal{Y}_{\text{pred}} = \mathcal{M}_{\text{pre-train}}(\mathcal{G}_{\text{train}}),$$

where $\mathcal{Y}_{\text{pred}}$ is the predicted label associated with the graph $\mathcal{G}_{\text{train}}$. This setting evaluates the model's ability to learn from labeled data and generalize across different tasks.

3.4.2 Cross-domain Zero-shot Learning. In this setting, a model is pre-trained on one set of graphs $\mathcal{G}_{\text{train}}$ and then tested on a different graph without fine-tuning. This setting tests the GFM's ability to generalize across different domains and apply learned knowledge to unseen graphs. Let $\mathcal{G}_{\text{train}}$ be a set of graphs from one domain used for pre-training, and $\mathcal{G}_{\text{test}}$ be a graph from a different domain for evaluation. The model $\mathcal{M}_{\text{pre-train}}$ is pre-trained on $\mathcal{G}_{\text{train}}$ and then directly evaluated on $\mathcal{G}_{\text{test}}$ without further fine-tuning:

$$\mathcal{Y}_{\text{pred}} = \mathcal{M}_{\text{pre-train}}(\mathcal{G}_{\text{test}}),$$

where $\mathcal{Y}_{\text{pred}}$ is the predicted label for the test graph $\mathcal{G}_{\text{test}}$. This setting evaluates the model's ability to transfer knowledge from one domain to another without requiring task-specific fine-tuning, demonstrating the model's zero-shot generalization ability.

3.4.3 Few-shot Learning. In the few-shot learning setting, the model is fine-tuned on a small number of randomly selected labeled samples from the target graph $\mathcal{G}_{\text{test}}$ after being pre-trained on a source graph set $\mathcal{G}_{\text{train}}$. This scenario tests the model's ability to adapt quickly to new tasks or domains with limited supervision. Formally, let $\mathcal{G}_{\text{train}}$ be the pre-trained graph set, and $\mathcal{G}_{\text{test}}$ be a small set of labeled samples from a new target graph for fine-tuning. The model $\mathcal{M}_{\text{pre-train}}$ is fine-tuned on this few-shot dataset:

$$\mathcal{L}_{\text{fine-tune}} = \sum_{i=1}^{k} \mathcal{L}(\mathcal{M}_{\text{pre-train}}(\mathcal{G}_i), \mathcal{Y}_i),$$

where k is the number of labeled samples in the few-shot set $\mathcal{G}_{\text{test}}$, and $\mathcal{Y}_i$ are the true labels associated with the few-shot samples. After fine-tuning, we obtain the fine-tuned GFM $\mathcal{M}_{\text{fine-tune}}$. It is evaluated on the same few-shot dataset or other target domain graphs.

$$\mathcal{Y}_{\text{pred}} = \mathcal{M}_{\text{fine-tune}}(\mathcal{G}_{\text{test}}),$$

where $\mathcal{Y}_{\text{pred}}$ is the predicted label for the few-shot samples in $\mathcal{G}_{\text{test}}$. This setting evaluates the model's ability to quickly adapt to new, unseen graphs with minimal labeled data, showcasing its flexibility in few-shot learning scenarios.

3.4.4 In-context Learning. In the in-context learning setting, the model is provided with samples from the target graph as part of an in-context demonstration, and it performs inference directly on new instances from the target graph without any additional fine-tuning. This scenario evaluates the model's ability to utilize the provided context at inference time to make predictions. Formally, let $\mathcal{G}_{\text{test}}$ represent the target graph. The model is provided with a subset of samples from $\mathcal{G}_{\text{test}}$ as an in-context demonstration, and uses this context to make predictions for new instances from the same target graph, without further fine-tuning:

$$\mathcal{Y}_{\text{pred}} = \mathcal{M}_{\text{pre-train}}(\mathcal{S}_{\text{context}}, \mathcal{G}_{\text{test}}),$$

where $\mathcal{S}_{\text{context}} \subset \mathcal{G}_{\text{test}}$ denotes the subset of samples from the target graph $\mathcal{G}_{\text{test}}$ with provided labels used as the in-context demonstration. $\mathcal{S}_{\text{context}}$ provides the necessary contextual information for the inference process. This setting evaluates the model's ability to leverage context-dependent demonstrations from $\mathcal{G}_{\text{test}}$ to make predictions without fine-tuning, emphasizing its in-context reasoning and generalization ability.

3.4.5 Summary of Evaluation Settings. Each of these evaluation settings assesses different aspects of GFM's generalization ability:

- *Supervised Learning*: Tests the model's ability to generalize across different graph tasks.
- *Cross-domain Zero-shot Learning*: Evaluates the model's ability to transfer knowledge across different graphs without additional fine-tuning.
- *Few-shot Learning*: Assesses the model's ability to adapt to new graphs with minimal labelled data.
- *In-context Learning*: Tests the model's ability to use contextual information at inference time.

These diverse settings provide a comprehensive evaluation framework for measuring the generalization and adaptability of GFMs across various graph tasks and domains.

4 GFMBench Empirical Analysis

4.1 Supervised Learning

In this experiment setting, we conduct a mixed pre-training of various Graph Foundation Models (GFMs) on three benchmark datasets: OAG-CS, IMDB, and DBLP. The goal is to evaluate the performance of these models across different tasks. We focus on two common graph-related tasks: Node Classification (NC) and Link Prediction (LP). The models undergo pre-training on the respective datasets, and we report their performance based on these tasks.

Experiment results are shown in Table 2. We can have the following observations. **(1) GFMs exhibit reasonable performance across different tasks**: As shown in the table, most models exhibit robust performance across multiple tasks on all three datasets. Models like GPT-GNN, PT-HGNN, and SGFormer demonstrate consistent results across different tasks. This suggests that GFMs possess the ability to unify learning from multiple graph datasets, highlighting their cross-domain applicability and generalization capacity. This supports the potential of GFMs in handling various graph data types. **(2) No single model stands out among the categories (GNN, LLM, Hybrid)**: The results reveal significant variability in the performance of GNN, LLM, and Hybrid models across different datasets and tasks. For instance, PT-HGNN performs well on OAG-CS and IMDB, but not as well on DBLP. Similarly, HiGPT excels on IMDB but shows less competitive results on OAG-CS and DBLP. This indicates that no single model type consistently outperforms the others across all tasks and datasets. The performance differences suggest that the choice of model may depend heavily on the specific characteristics of the dataset and task at hand. **(3) The research on GFMs is still in its early stages, with vast room for improvement**: Although some models show competitive performance, the results highlight that there is still significant room for improvement in GFM research. No model has yet consistently outperformed others across all tasks, which indicates the need for further exploration in this field. Future work could focus on optimizing model architectures, integrating more sophisticated graph structures, or leveraging external domain knowledge to improve the generalization ability and performance across tasks. This underscores the potential for further innovation and refinement.

Target	**OAG-CS**				**IMDB**				**DBLP**			
Task	NC		LP		NC		LP		NC		LP	
Metric	Micro-F1	Macro-F1	NDCG	MRR	Micro-F1	Macro-F1	NDCG	MRR	Micro-F1	Macro-F1	NDCG	MRR
SGFormer	0.4872	0.2394	0.2231	0.2573	0.3985	0.2144	0.5082	0.4226	0.4918	0.2946	0.6902	0.5009
GPT-GNN	0.3906	0.2809	0.3181	0.2924	0.4180	0.2580	0.8729	0.7524	0.7109	0.6824	**0.8520**	**0.7342**
PT-HGNN	0.5234	0.2650	0.3265	0.3430	0.5249	0.3094	0.7126	0.6405	0.6158	0.4235	0.8062	0.6740
LLaGA	0.0748	0.0809	0.1569	0.0276	0.2383	0.2604	0.3782	0.3409	0.3107	0.2981	0.5487	0.4124
WalkLM	0.4654	0.3176	**0.4813**	**0.5874**	0.5234	0.5229	0.6360	0.4910	0.3300	0.3032	0.8154	0.5556
OFA	0.4025	0.2642	0.3065	0.3114	0.4035	0.2487	0.5526	0.4630	0.5172	0.3365	0.7283	0.5367
GraphGPT	0.5136	0.2678	-	-	0.5023	0.4309	-	-	0.3961	0.3846	-	-
HiGPT	**0.5689**	0.2950	-	-	0.6509	0.4811	-	-	0.5232	0.4742	-	-
LMCH	0.4723	**0.3208**	0.3120	0.3404	**0.6822**	**0.6829**	**0.9345**	**0.9724**	**0.8547**	**0.8438**	0.8362	0.5210

Table 2: Performance comparison on node classification (NC) and link prediction (LP) tasks in supervised learning setting. However, since the absence of specific instructions for link prediction tasks on target HGs, we focus solely on the node classification task for GraphGPT and HiGPT. The best performer is marked in bold, and the runner-up is underlined. Models are grouped by backbone structures: GNN-based , LLM-based , Hybrid .

4.2 Cross-domain Zero-shot Learning

The goal of this experiment is to evaluate the zero-shot inference capability of the models in a cross-domain setting. To achieve this, we first conduct large-scale pre-training on the OAG-CS graph. The models are then directly applied to perform inference on other graphs from different domains, specifically testing their performance on the IMDB graph (a different domain) and the DBLP graph (a same-domain comparison). This setting allows us to test how well the models can generalize across unseen domains and graph structures without further domain-specific fine-tuning.

Experiment results are shown in Table 3, we can have the following observations. **(1) Reasonable Performance Across Datasets**: In the cross-domain zero-shot setting, the models exhibit fairly reasonable results on both IMDB and DBLP. This indicates that the GFMs demonstrate strong applicability in zero-shot scenarios. Despite being pre-trained on the OAG-CS graph, the models are able to generalize well to completely different domains, showcasing their capacity to learn transferable representations. This supports the idea that GFMs possess the ability to capture graph structural features that can be applied across different domains without the need for additional domain-specific pre-training. **(2) Generalization Ability Compared to Supervised Learning**: When comparing the performance of the models in the zero-shot setting to their counterparts in the supervised learning setup in Table 2, it is evident that the drop in performance is not drastic. For instance, models like PT-HGNN and WalkLM, although showing some decline in the zero-shot setting, still maintain relatively competitive performance across all metrics. This suggests that the GFM framework possesses a strong generalization capability, allowing the models to perform reasonably well in unseen domains, without requiring additional fine-tuning on the target graph. **(3) Importance of Fine-tuning for Specific Graphs**: However, despite the promising results, there is a noticeable performance gap when compared to their supervised learning counterparts. This highlights an important aspect: although GFMs can generalize across domains, the models still experience a decline in performance when applied to a specific target graph. This suggests that for the best results on a particular graph, fine-tuning may still be necessary. The ability to adapt to specific graphs is a crucial aspect of GFM, which motivates us on the few-shot learning investigation.

4.3 Few-shot Learning

The goal of this experiment is to evaluate the adaptability of GFMs to new graph datasets with limited supervision. In this setting, we first pre-train a model on the large-scale OAG-CS graph and then fine-tune it on the IMDB and DBLP datasets using a few labeled samples. Specifically, we conduct few-shot fine-tuning with 1-shot, 3-shot, and 5-shot settings to assess the models' ability to generalize with minimal labeled data. We evaluate the models on Node Classification (NC) and Link Prediction (LP) tasks. These tasks allow us to analyze how well GFMs can leverage limited labeled samples to improve their performance on new graphs.

We demonstrate few-shot experiments in Figure 2. In the few-shot fine-tuning setting, the models generally perform better with more fine-tuning samples. Specifically, performance improves as the number of labeled samples increases, with 5-shot fine-tuning yielding the best results. For example, in the IMDB Node Classification task, HiGPT shows a significant improvement when moving from 1-shot to 5-shot, highlighting that GFM models benefit from more labeled data for downstream tasks. However, even with just 5-shot fine-tuning, models demonstrate strong adaptation capabilities, performing well with limited labeled data. For instance, in the DBLP Node Classification task, PT-HGNN achieves solid performance even with only 5-shot fine-tuning, suggesting that GFM models can generalize well with fewer examples. We also observe the variability in the models' ability to adapt. On node classification task of DBLP dataset, the improvement of LLaGA is very significant when growing from 3-shot to 5-shot. However, GPT-GNN nearly remains the same across 1-5 shots. This highlights the importance of model architecture in determining adaptation success.

4.4 In-context Learning

In this experiment, we focus on models that have the ability to perform in-context learning, which is currently supported only by

Target	**IMDB**				**DBLP**			
Task	NC		LP		NC		LP	
Metric	Micro-F1	Macro-F1	NDCG	MRR	Micro-F1	Macro-F1	NDCG	MRR
SGFormer	0.2154	0.1570	0.4162	0.2376	0.2831	0.1345	0.6187	0.3579
GPT-GNN	0.2913	0.2085	0.8295	0.6974	0.2773	0.1891	**0.8115**	0.6833
PT-HGNN	0.3216	0.2443	0.6287	0.6107	0.2938	0.1942	0.7938	**0.7165**
LLaGA	0.2042	0.1899	0.3192	0.2897	0.1906	0.1722	0.4882	0.3723
WalkLM	0.3248	0.2368	0.6013	0.4250	0.2685	0.2317	0.7612	0.5167
OFA	0.2463	0.1956	0.4718	0.3005	0.2853	0.1668	0.7012	0.4086
GraphGPT	0.3671	0.2782	-	-	**0.2951**	**0.2437**	-	-
HiGPT	**0.4901**	**0.3136**	-	-	0.2831	0.2395	-	-
LMCH	0.2874	0.1488	**0.8548**	**0.8874**	0.2586	0.1214	0.7960	0.4333

Table 3: Performance comparison on node classification (NC) and link prediction (LP) tasks in cross-domain zero-shot setting. The best performer is marked in bold, and the runner-up is underlined. Models are grouped by backbone structures: GNN-based , LLM-based , Hybrid .

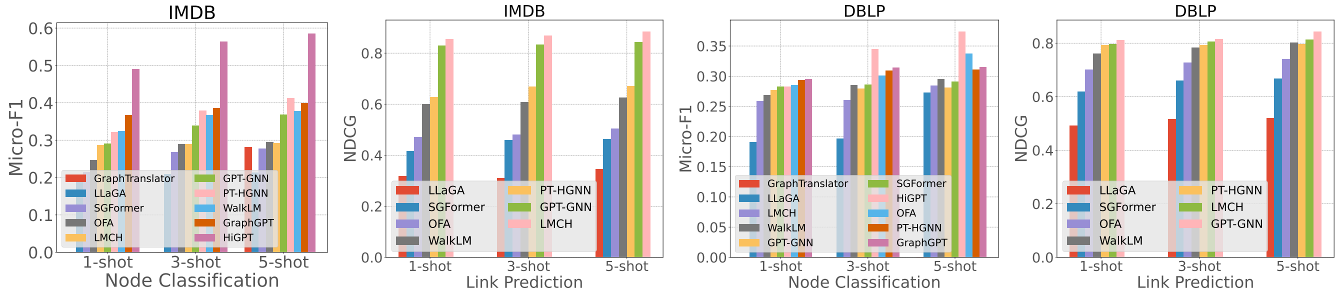


Figure 2: Few-shot learning experiment results.

LLM-based and certain hybrid models. Specifically, we conduct experiments using GraphGPT [25], HiGPT [26], and GraphTranslator [42], all of which are capable of leveraging in-context learning. These models are first pre-trained on the OAG-CS graph and then evaluated on the IMDB and DBLP datasets in a zero-shot setting for the Node Classification task. To evaluate the models' ability to use context effectively, we provide varying numbers of context examples (ranging from 1 to 5) from the target graph during inference. This allows us to observe how the models' performance changes with more context, testing their ability to generalize to new graphs without further fine-tuning.

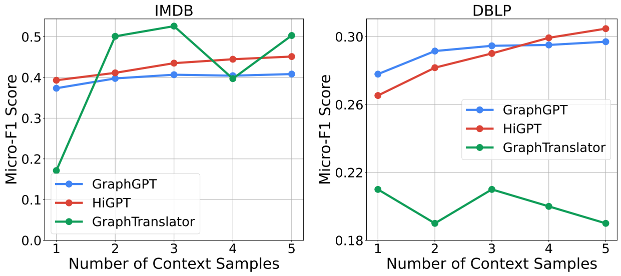


Figure 3: In-context learning experiment results.

Experiment results are shown in Figure 3. All three models—GraphGPT, HiGPT, and GraphTranslator—demonstrate the ability to perform in-context learning. Both GraphGPT and HiGPT show relatively stable performance across different context sample sizes. As the number of context samples increases from 1 to 5, these models exhibit consistent improvement, as seen in the Micro-F1 scores for both DBLP and IMDB datasets. This indicates that these models can leverage additional context to enhance their performance in the Node Classification task. Although the models can improve with the increasing number of context samples, the overall performance gain is not very significant. For instance, in both datasets, the improvement from 1 context sample to 5 context samples is noticeable, but the magnitude of the change is limited. This suggests that the in-context learning capability of GFM models still requires further development. Despite the addition of context samples, the performance improvements are relatively modest. This could be because the selected context samples may not provide highly relevant or effective information for the inference task. It suggests that simply increasing the number of context samples might not be sufficient for substantial performance gains. A key area for future research could involve developing strategies for context selection, ensuring that the provided context contains the most relevant and useful information for the graph-based inference task.

5 Online Application Platform

We further developed an online application platform for GFMBench [3]. It serves as both a demonstration of GFM capabilities and a practical tool for the research community. The platform encompasses two primary categories of applications: basic information services and relationship-based analysis. One pre-trained GFM supports all the introduced applications and users can select different GFMs to test their generalization ability in an intervative manner.

5.1 Basic Information Services

The platform provides several fundamental services for academic literature access and analysis:

5.1.1 Homepage Recommendations. The system implements a dynamic recommendation engine that showcases the latest research papers on the homepage. This feature is automatically updated daily, ensuring users have access to current research developments. The recommendation algorithm considers both publication recency and relevance to user interests.

5.1.2 Literature and Author Search. Users can perform detailed searches using various parameters including paper titles, author names, and keywords. The search functionality leverages GFM's understanding of academic text to return highly relevant results.

5.1.3 Author Profiles. Each author in the system has a comprehensive profile page that consolidates their basic biographical information, educational background, professional experience, academic achievements, and complete publication history with citation metrics into a single, accessible interface.

5.1.4 Research Trend Visualization. The platform generates visual representations of research trends by combining interactive word clouds of research interests, temporal analysis of topic evolution, and comprehensive metrics tracking publication frequency.

5.2 Relationship-Based Analysis

Beyond basic information services, the platform leverages GFM's capabilities to provide sophisticated relationship-based analyses:

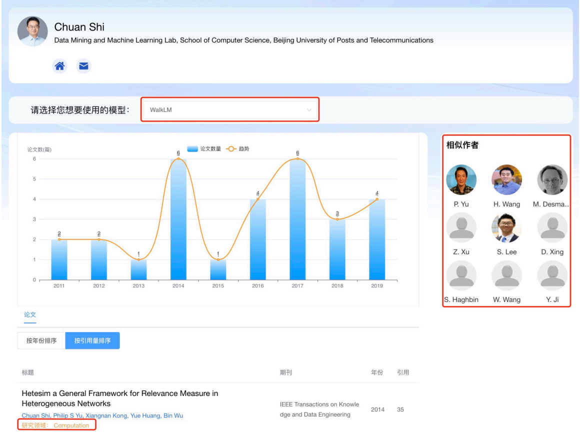
5.2.1 Topic-Based Exploration. The system provides a multi-faceted interface that synthesizes relevant papers, key contributing authors, related conferences, and temporal topic evolution to offer comprehensive insights into academic topics.

5.2.2 Research Interest Prediction. The system implements dual predictive analyses, combining paper-level prediction of future research directions based on content and citations with author-level prediction of future research interests derived from publication history and collaboration patterns.

5.2.3 Similar Author Recommendation. The platform provides intelligent author recommendations by analyzing research topic overlap, publication venues, citation networks, and collaboration history to identify researchers with similar academic profiles and facilitate potential collaborations.

5.3 Implementation Details

The platform is implemented as a web-based application with a responsive interface, ensuring accessibility across different devices with free access to all users. A screenshot of our system is shown in Figure 4. Our platform provides an intuitive interface for users to evaluate and compare different GFMs' performance on various applications. When a specific GFM is selected (e.g., WalkLM in our illustrative example), the system automatically generates comprehensive results for multiple downstream applications. For example, the system provides similar author recommendations, displayed in an easily accessible panel that shows the most relevant researchers based on the selected model's embeddings and classification results. Simultaneously, the GFM analyzes and presents research topic recommendations for each publication, demonstrating the model's capability to capture semantic relationships within the academic network. This interactive interface enables users to directly observe and assess the practical performance of different GFMs in real-world scenarios. The system's visualization components, including publication history trends and collaboration networks, provide additional context for understanding the model's predictions and recommendations.



Chuan Shi
Data Mining and Machine Learning Lab, School of Computer Science, Beijing University of Posts and Telecommunications

请选择您想要使用的模型： WalkLM

标题	期刊	年份	引用
Hetesim a General Framework for Relevance Measure in Heterogeneous Networks Chuan Shi, Philip S Yu, Xiangnan Kong, Yue Huang, Bin Wu 研究领域： Computation	IEEE Transactions on Knowledge and Data Engineering	2014	35

Figure 4: Screenshot of our online application.

6 Conclusion

In this paper, we presented GFMBench, a comprehensive benchmark pipeline for Graph Foundation Models (GFMs) that addresses critical challenges in standardizing evaluation and demonstrating real-world applicability. Through its integration of state-of-the-art models and diverse datasets, GFMBench establishes a unified framework for assessing model performance across various learning scenarios, including supervised, zero-shot, few-shot and in-context learning. Our implementation of practical applications on the Open Academic Graph, such as topic search and author recommendation, demonstrates the pipeline's capability to support multiple downstream tasks while effectively utilizing both structural and attribute information. With its modular design and extensive documentation, GFMBench serves as a valuable resource for the research community, advancing the development and practical deployment of graph foundation models.

[3] http://39.106.34.142:8080/recommend

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. 62192784, U22B2038, 62472329).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Mengru Chen, Chao Huang, Lianghao Xia, Wei Wei, Yong Xu, and Ronghua Luo. 2023. Heterogeneous graph contrastive learning for recommendation. In *Proceedings of the sixteenth ACM international conference on web search and data mining*. 544–552.

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

[4] Zhikai Chen, Haitao Mao, Jingzhe Liu, Yu Song, Bingheng Li, Wei Jin, Bahare Fatemi, Anton Tsitsulin, Bryan Perozzi, Hui Liu, et al. 2024. Text-space graph foundation models: Comprehensive benchmarks and new insights. *arXiv preprint arXiv:2406.10727* (2024).

[5] Pengfei Ding, Yan Wang, and Guanfeng Liu. 2023. Cross-heterogeneity graph few-shot learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 420–429.

[6] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 135–144.

[7] Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. 2019. Metapath-guided heterogeneous graph neural network for intent recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2478–2486.

[8] Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428* (2019).

[9] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. 2020. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *Proceedings of the web conference 2020*. 2331–2341.

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

[11] Hui Han, Tianyu Zhao, Cheng Yang, Hongyi Zhang, Yaoqi Liu, Xiao Wang, and Chuan Shi. 2022. Openhgnn: An open source toolkit for heterogeneous graph neural network. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3993–3997.

[12] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*. Ieee, 263–272.

[13] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. 2020. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 1857–1867.

[14] Xunqiang Jiang, Tianrui Jia, Yuan Fang, Chuan Shi, Zhe Lin, and Hui Wang. 2021. Pre-training on large-scale heterogeneous graph. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 756–766.

[15] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[16] Xiangjie Kong, Yajie Shi, Shuo Yu, Jiaying Liu, and Feng Xia. 2019. Academic social networks: Modeling, analysis, mining and applications. *Journal of Network and Computer Applications* 132 (2019), 86–103.

[17] Tianqianjin Lin, Pengwei Yan, Kaisong Song, Zhuoren Jiang, Yangyang Kang, Jun Lin, Weikang Yuan, Junjie Cao, Changlong Sun, and Xiaozhong Liu. 2024. LangGFM: A Large Language Model Alone Can be a Powerful Graph Foundation Model. *arXiv preprint arXiv:2410.14961* (2024).

[18] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. 2023. One for all: Towards training one graph model for all classification tasks. *arXiv preprint arXiv:2310.00149* (2023).

[19] Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S Yu, et al. 2023. Towards graph foundation models: A survey and beyond. *arXiv preprint arXiv:2310.11829* (2023).

[20] Yaoqi Liu, Cheng Yang, Tianyu Zhao, Hui Han, Siyuan Zhang, Jing Wu, Guangyu Zhou, Hai Huang, Hui Wang, and Chuan Shi. 2023. Gammagl: A multi-backend library for graph neural networks. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2861–2870.

[21] Anjun Ma, Xiaoying Wang, Jingxian Li, Cankun Wang, Tong Xiao, Yuntao Liu, Hao Cheng, Juexin Wang, Yang Li, Yuzhou Chang, et al. 2023. Single-cell biological network inference using a heterogeneous graph transformer. *Nature Communications* 14, 1 (2023), 964.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[23] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 1150–1160.

[24] Yanchao Tan, Zihao Zhou, Hang Lv, Weiming Liu, and Carl Yang. 2024. Walklm: A uniform language model fine-tuning framework for attributed graph embedding. *Advances in Neural Information Processing Systems* 36 (2024).

[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. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 491–500.

[26] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. 2024. Higpt: Heterogeneous graph language model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2842–2853.

[27] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 990–998.

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

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

[30] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. 2020. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies* 1, 1 (2020), 396–413.

[31] Minjie Yu Wang. 2019. Deep graph library: Towards efficient and scalable deep learning on graphs. In *ICLR workshop on representation learning on graphs and manifolds*.

[32] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*. 2022–2032.

[33] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=ryGs6iA5Km

[34] Bo Yan, Yang Cao, Haoyu Wang, Wenchuan Yang, Junping Du, and Chuan Shi. 2024. Federated heterogeneous graph neural network for privacy-preserving recommendation. In *Proceedings of the ACM Web Conference 2024*. 3919–3929.

[35] Bo Yan, Cheng Yang, Chuan Shi, Jiawei Liu, and Xiaochen Wang. 2023. Abnormal event detection via hypergraph contrastive learning. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM, 712–720.

[36] Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. 2020. Heterogeneous network representation learning: A unified framework with survey and benchmark. *IEEE Transactions on Knowledge and Data Engineering* 34, 10 (2020), 4854–4873.

[37] Jinyu Yang, Ruijia Wang, Cheng Yang, Bo Yan, Qimin Zhou, Yang Juan, and Chuan Shi. 2025. Harnessing Language Model for Cross-Heterogeneity Graph Knowledge Transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 13026–13034.

[38] Liangwei Yang, Zhiwei Liu, Yu Wang, Chen Wang, Ziwei Fan, and Philip S Yu. 2022. Large-scale personalized video game recommendation via social-aware contextualized graph neural network. In *Proceedings of the ACM Web Conference 2022*. 3376–3386.

[39] Liangwei Yang, Shengjie Wang, Yunzhe Tao, Jiankai Sun, Xiaolong Liu, Philip S Yu, and Taiqing Wang. 2023. Dgrec: Graph neural network for recommendation with diversified embedding generation. In *Proceedings of the sixteenth ACM international conference on web search and data mining*. 661–669.

[40] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems* 33 (2020), 5812–5823.

[41] Fanjin Zhang, Xiao Liu, Jie Tang, Yuxiao Dong, Peiran Yao, Jie Zhang, Xiaotao Gu, Yan Wang, Bin Shao, Rui Li, et al. 2019. OAG: Toward linking large-scale heterogeneous entity graphs. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2585–2595.

[42] Mengmei Zhang, Mingwei Sun, Peng Wang, Shen Fan, Yanhu Mo, Xiaoxiao Xu, Hong Liu, Cheng Yang, and Chuan Shi. 2024. Graphtranslator: Aligning graph

model to large language model for open-ended tasks. In *Proceedings of the ACM Web Conference 2024*. 1003–1014.

[43] Haiteng Zhao, Shengchao Liu, Ma Chang, Hannan Xu, Jie Fu, Zhihong Deng, Lingpeng Kong, and Qi Liu. 2023. Gimlet: A unified graph-text model for instruction-based molecule zero-shot learning. *Advances in Neural Information Processing Systems* 36 (2023).

[44] Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. 2023. Learning on Large-scale Text-attributed Graphs via Variational Inference. In *Proc. of ICLR*.