Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)
Survey Track

A Survey of Graph Meets Large Language Model: Progress and Future Directions

Yuhan Li[1], **Zhixun Li**[2], **Peisong Wang**[3], **Jia Li**[1], **Xiangguo Sun**[2], **Hong Cheng**[2] and **Jeffrey Xu Yu**[2]

[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]SHIAE and SEEM, The Chinese University of Hong Kong
[3]Tsinghua University

Abstract

Graph plays a significant role in representing and analyzing complex relationships in real-world applications such as citation networks, social networks, and biological data. Recently, Large Language Models (LLMs), which have achieved tremendous success in various domains, have also been leveraged in graph-related tasks to surpass traditional Graph Neural Networks (GNNs) based methods and yield state-of-the-art performance. In this survey, we first present a comprehensive review and analysis of existing methods that integrate LLMs with graphs. First of all, we propose a new taxonomy, which organizes existing methods into three categories based on the role (i.e., enhancer, predictor, and alignment component) played by LLMs in graph-related tasks. Then we systematically survey the representative methods along the three categories of the taxonomy. Finally, we discuss the remaining limitations of existing studies and highlight promising avenues for future research. The relevant papers are summarized and will be consistently updated at: https://github.com/yhLeeee/Awesome-LLMs-in-Graph-tasks.

1 Introduction

Graph, or graph theory, serves as a fundamental part of numerous areas in the modern world, particularly in technology, science, and logistics [Ji *et al.*, 2021]. Graph data represents the structural characteristics between nodes, thus illuminating relationships within the graph's components. Many real-world datasets, such as citation networks [Sen *et al.*, 2008], social networks [Hamilton *et al.*, 2017], and molecular [Wu *et al.*, 2018], are intrinsically represented as graphs. To tackle graph-related tasks, Graph Neural Networks (GNNs) [Kipf and Welling, 2016; Velickovic *et al.*, 2018] have emerged as one of the most popular choices for processing and analyzing graph data. The main objective of GNNs is to acquire expressive representations at the node, edge, or graph level for different kinds of downstream tasks through recursive message passing and aggregation mechanisms among nodes.

In recent years, significant advancements have been made in Large Language Models (LLMs) like Transformers [Vaswani *et al.*, 2017], BERT [Kenton and others, 2019], GPT [Brown *et al.*, 2020], and their variants. These LLMs can be easily applied to various downstream tasks with little adaptation, demonstrating remarkable performance across various natural language processing tasks, such as sentiment analysis, machine translation, and text classification [Zhao *et al.*, 2023c]. While their primary focus has been on text sequences, there is a growing interest in enhancing the multi-modal capabilities of LLMs to enable them to handle diverse data types, including graphs, images, and videos.

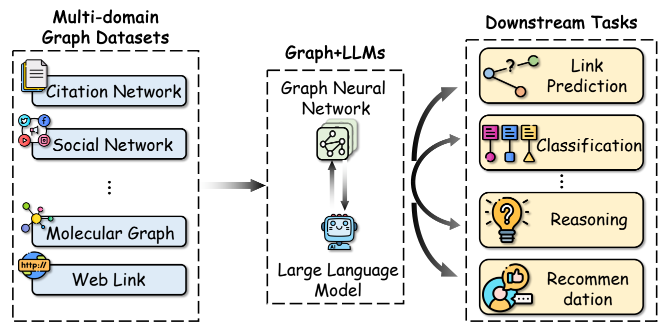


Figure 1: Across a myriad of graph domains, the integration of graphs and LLMs demonstrates success in various downstream tasks.

LLMs help graph-related tasks. With the help of LLMs, there has been a notable shift in the way we interact with graphs, particularly those containing nodes associated with text attributes. As shown in Figure 1, the integration of graphs and LLMs demonstrates success in various downstream tasks across a myriad of graph domains. Integrating LLMs with traditional GNNs can be mutually beneficial and enhance graph learning. While GNNs are proficient at capturing structural information, they primarily rely on semantically constrained embeddings as node features, limiting their ability to express the full complexities of the nodes. Incorporating LLMs, GNNs can be enhanced with stronger node features that effectively capture both structural and contextual aspects. On the other hand, LLMs excel at encoding text but often struggle to capture structural information present in graph data. Combining GNNs with LLMs can leverage the robust textual understanding of LLMs while harnessing GNNs' ability to capture structural relationships, leading to more powerful graph learning. To achieve a better systematic overview, as shown in Figure 2, we

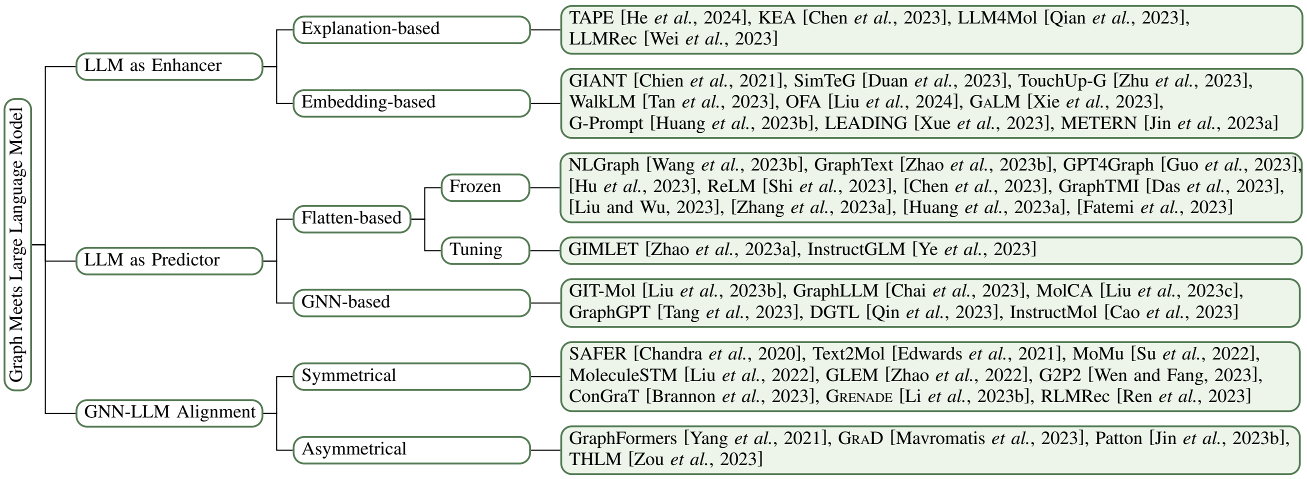


Figure 2: A taxonomy of models for solving graph tasks with the help of large language models (LLMs) with representative examples.

follow [Chen *et al.*, 2023] to organize our first-level taxonomy, categorizing based on the role (i.e., enhancer, predictor, and alignment component) played by LLMs throughout the entire model pipeline. We further refine our taxonomy and introduce more granularity to the initial categories.

Motivations. Although LLMs have been increasingly applied in graph-related tasks, this rapidly expanding field still lacks a systematic review. [Zhang *et al.*, 2023b] conducts a forward-looking survey, presenting a perspective paper that discusses the challenges and opportunities associated with the integration of graphs and LLMs. [Liu *et al.*, 2023a] provide another related survey that summarizes existing graph foundation models and offers an overview of pre-training and adaptation strategies. However, both of them have limitations in terms of comprehensive coverage and the absence of a taxonomy specifically focused on how LLMs enhance graphs. In contrast, we concentrate on scenarios where both graph and text modalities coexist and propose a more fine-grained taxonomy to systematically review and summarize the current status of LLMs techniques for graph-related tasks.

Contributions. The contributions of this work can be summarized from the following three aspects. **(1)** *A structured taxonomy*. A broad overview of the field is presented with a structured taxonomy that categorizes existing works into four categories (Figure 2). **(2)** *A comprehensive review*. Based on the proposed taxonomy, the current research progress of LLMs for graph-related tasks is systematically delineated. **(3)** *Some future directions*. We discuss the remaining limitations of existing works and point out possible future directions.

2 Preliminary

2.1 Graph Neural Networks

Definitions. Most existing GNNs follow the message-passing paradigm which contains message aggregation and feature update, such as GCN [Kipf and Welling, 2016] and GAT [Velickovic *et al.*, 2018]. They generate node representations by iteratively aggregating information of neighbors and updating them with non-linear functions. The forward process can be defined as:

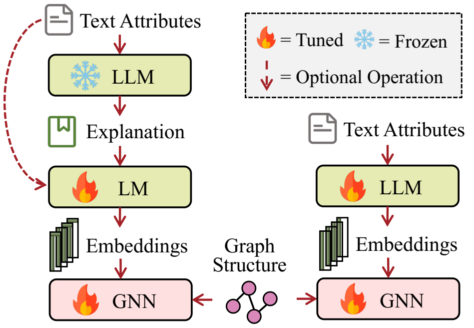
$$h_i^{(l)} = \mathbf{U}\Big(h_i^{(l-1)}, \mathbf{M}(\{h_i^{(l-1)}, h_j^{(l-1)} | v_j \in \mathcal{N}_i\})\Big)$$

where $h_i^{(l)}$ is the feature vector of node i in the l-th layer, and $\mathcal{N}_i$ is a set of neighbor nodes of node i. $\mathbf{M}$ denotes the message passing function of aggregating neighbor information, $\mathbf{U}$ denotes the update function with central node feature and neighbor node features as input. By stacking multiple layers, GNNs can aggregate messages from higher-order neighbors.

2.2 Large Language Models

Definitions. While there is currently no clear definition for LLMs [Shayegani *et al.*, 2023], here we provide a specific definition for LLMs mentioned in this survey. Two influential surveys on LLMs [Zhao *et al.*, 2023c; Yang *et al.*, 2023] distinguish between LLMs and pre-trained language models (PLMs) from the perspectives of model size and training approach. To be specific, LLMs are those huge language models (i.e., billion-level) that undergo pre-training on a significant amount of data, whereas PLMs refer to those early pre-trained models with moderate parameter sizes (i.e., million-level), which can be easily further fine-tuned on task-specific data to achieve better results to downstream tasks. Due to the relatively smaller parameter size of GNNs, incorporating GNNs and LLMs often does not require LLMs with large parameters. Hence, we follow [Liu *et al.*, 2023a] to extend the definition of LLMs in this survey to encompass both LLMs and PLMs as defined in previous surveys.

Evolution. LLMs can be divided into two categories based on non-autoregressive and autoregressive language modeling. Non-autoregressive LLMs typically concentrate on natural language understanding and employ a "masked language modeling" pre-training task, while autoregressive LLMs focus more on natural language generation, frequently leveraging the "next token prediction" objective as their foundational task. Classic encoder-only models such as BERT and RoBERTa fall under the category of non-autoregressive LLMs. Recently, autoregressive LLMs have witnessed continuous development.



(a) Explanation-based (b) Embedding-based

Figure 3: The illustration of LLM-as-enhancer approaches.

Examples include Flan-T5 and ChatGLM, which are built upon the encoder-decoder structure, as well as GPT-3 and LLaMA, which are based on decoder-only architectures. Significantly, advancements in architectures and training methodologies of LLMs have given rise to emergent capabilities [Wei *et al.*, 2022a], which is the ability to handle complex tasks in few-shot or zero-shot scenarios via some techniques such as in-context learning [Radford *et al.*, 2021] and chain-of-thought [Wei *et al.*, 2022b].

2.3 Proposed Taxonomy

We propose a taxonomy (as illustrated in Figure 2) that organizes representative techniques involving both graph and text modalities into three main categories: **(1)** LLM as Enhancer, where LLMs are used to enhance the classification performance of GNNs. **(2)** LLM as Predictor, where LLMs utilize the input graph structure information to make predictions. **(3)** GNN-LLM Alignment, where LLMs semantically enhance GNNs through alignment techniques. We note that in some models, due to the rarity of LLMs' involvement, it becomes difficult to categorize them into these three main classes. Therefore, we separately organize them into the "Others" category with specific roles. For example, LLM-GNN [Chen *et al.*, 2024] actively selects nodes for ChatGPT to annotate, thereby augmenting the GNN training by utilizing the LLM as an *annotator*. GPT4GNAS [Wang *et al.*, 2023a] considers the LLM as an experienced *controller* in the task of graph neural architecture search. and utilizes GPT-4 [OpenAI, 2023] to explore the search space. Furthermore, ENG [Yu *et al.*, 2023] empowers the LLM as a *sample generator* to generate additional training samples with labels to provide sufficient supervision signals for GNNs.

In the following sections, we present a comprehensive survey along the three main categories of our taxonomy for incorporating LLMs into graph-related tasks, respectively.

3 LLM as Enhancer

GNNs have become powerful tools to analyze graph-structure data. However, the most mainstream benchmark datasets (e.g., Cora and Ogbn-Arxiv) adopt naive methods to encode text information in text-attributed graphs (TAGs) using shallow embeddings, such as bag-of-words, skip-gram [Mikolov *et al.*, 2013], or TF-IDF [Salton and Buckley, 1988]. This inevitably constrains the performance of GNNs on TAGs. LLM-as-enhancer approaches correspond to enhancing the quality of node embeddings with the help of powerful LLMs. The derived embeddings are attached to the graph structure to be utilized by any GNNs or directly inputted into downstream classifiers for various tasks. We naturally categorize these approaches into two branches: explanation-based and embedding-based, depending on whether they use LLMs to produce additional textual information.

3.1 Explanation-based Enhancement

To enrich the textual attributes, explanation-based enhancement approaches focus on utilizing the strong zero-shot capability of LLMs to capture higher-level information. As shown in Figure 3(a), generally they prompt LLMs to generate semantically enriched additional information, such as explanations, knowledge entities, and pseudo labels.

For instance, TAPE [He *et al.*, 2024] is a pioneer work of explanation-based enhancement, which prompts LLMs to generate explanations and pseudo labels to augment textual attributes. After that, relatively small language models are fine-tuned on both original text data and explanations to encode text semantic information as initial node embeddings. [Chen *et al.*, 2023] explore the potential competence of LLMs in graph learning. They first compare embedding-visible LLMs with shallow embedding methods and then propose KEA to enrich the text attributes. KEA prompts LLMs to generate a list of knowledge entities along with text descriptions and encodes them by fine-tuned PLMs and deep sentence embedding models. LLM4Mol [Qian *et al.*, 2023] attempts to employ LLMs to assist in molecular property prediction. Specifically, it uses LLMs to generate semantically enriched explanations for the original SMILES and then fine-tunes a small-scale language model to conduct downstream tasks. LLMRec [Wei *et al.*, 2023] aims to utilize LLMs to figure out data sparsity and data quality issues in the graph recommendation system. It reinforces user-item interaction edges and generates user/item side information by LLMs. Lastly, it employs a lightweight GNN to encode the augmented recommendation network.

3.2 Embedding-based Enhancement

Refer to Figure 3(b), embedding-based enhancement approaches directly utilize LLMs to output text embeddings as initial node embeddings for GNN training. This kind of approach requires the use of embedding-visible or open-source LLMs because it needs to access text embeddings straightaway or fine-tune LLMs with structural information. Many of the current advanced LLMs (e.g., GPT4 [OpenAI, 2023] and PaLM [Chowdhery *et al.*, 2022]) are closed-source and only provide online services. Strict restrictions prevent researchers from accessing their parameters and output embeddings. Embedding-based approaches mostly adopt a cascading form and utilizes structure information to assist the language model in pre-training or fine-tuning.

Typically, GaLM [Xie *et al.*, 2023] pre-trains PLMs and GNN aggregator on a given large graph corpus to capture the information that can maximize utility towards massive applications and then fine-tunes the framework on a specific downstream application to further improve the performance. Several works also aim to generate node embeddings by incorporating structural information into the fine-tuning phase of LLMs. GIANT [Chien *et al.*, 2021], SimTeG [Duan *et al.*, 2023] and TouchUp-G [Zhu *et al.*, 2023] follow a similar way, they both fine-tune PLMs through link-prediction-like methods to help them perceive structural information. The subtle difference between them is that GIANT employs XR-Transformer to solve extreme multi-label classification over link prediction, TouchUp-G uses negative sampling during link prediction, while SimTeG employs parameter-efficient fine-tuning to accelerate the fine-tuning process. G-Prompt [Huang *et al.*, 2023b] introduces a graph adapter at the end of PLMs to help extract graph-aware node features. Once trained, task-specific prompts are incorporated to produce interpretable node representations for various downstream tasks. WalkLM [Tan *et al.*, 2023] is an unsupervised generic graph representation learning method. The first step of it is to generate attributed random walks on the graph and compose roughly meaningful textual sequences by automated textualization program. The second step is to fine-tune an LLM using textual sequences and extract representations from LLM. METERN [Jin *et al.*, 2023a] introduces relation prior tokens to capture the relation-specific signals and uses one language encoder to model the shared knowledge across relations. LEADING [Xue *et al.*, 2023] effectively finetunes LLMs and transfers risk knowledge in LLM to downstream GNN model with less computation cost and memory overhead.

A recent work, OFA [Liu *et al.*, 2024], attempts to propose a general graph learning framework, which can utilize a single graph model to conduct adaptive downstream prediction. It describes all nodes and edges using human-readable texts and encodes them from different domains into the same space by LLMs. Subsequently, the framework is adaptive to perform different tasks by inserting task-specific prompting substructures into the input graph.

3.3 Discussions

LLM-as-enhancer approaches have demonstrated superior performance on TAG, being able to effectively capture both textual and structural information. Moreover, they also exhibit strong flexibility, as GNNs and LLMs are plug-and-play, allowing them to leverage the latest techniques to address the encountered issues. However, despite some papers claiming strong scalability, in fact, LLM-as-enhancer approaches entail significant overhead when dealing with large-scale datasets. Taking explanation-based approaches as an example, they need to query LLMs' APIs for N times for a graph with N nodes, which is indeed a substantial cost.

4 LLM as Predictor

The core idea behind this category is to utilize LLMs to make predictions for a wide range of graph-related tasks, such as classifications and reasonings, within a unified generative paradigm. However, applying LLMs to graph modalities presents unique challenges, primarily because graph data often lacks straightforward transformation into sequential text, as different graphs define structures and features in different ways. We classify the models broadly into flatten-based and GNN-based predictions, depending on whether they employ GNNs to extract structural features for LLMs.

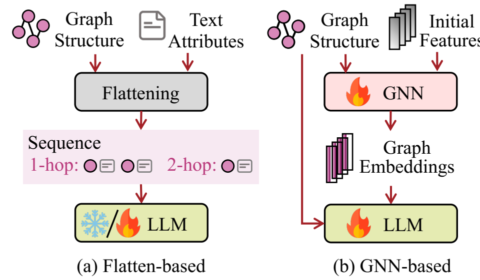


Figure 4: The illustration of LLM-as-predictor approaches.

4.1 Flatten-based Prediction

The majority of the existing attempts that utilize LLMs as predictors employ the strategy of flattening the graph into textual descriptions, which facilitates direct processing of graph data by LLMs through text sequences. As shown in Figure 4(a), flatten-based prediction typically involves two steps: **(1)** utilizing a flatten function to transform a graph structure into a sequence of nodes or tokens, and **(2)** a parsing function is then applied to retrieve the predicted label from the output generated by LLMs. As the core of flatten-based prediction, a variety of flatten functions has been leveraged.

GPT4Graph [Guo *et al.*, 2023] utilizes graph description languages such as GML and GraphML to represent graphs. These languages provide standardized syntax and semantics for representing the nodes and edges within a graph. Inspired by linguistic syntax trees, GraphText [Zhao *et al.*, 2023b] leverages graph-syntax trees to convert a graph structure to a sequence of nodes, which is then fed to LLMs for training-free graph reasoning. Furthermore, ReLM [Shi *et al.*, 2023] uses SMILES strings to provide one-dimensional linearizations of molecular graph structures. GIMLET [Zhao *et al.*, 2023a] adopts distance-based position embedding to extend the capability of LLMs to perceive graph structures. Graph data can be also represented through methods like adjacency matrices and adjacency lists. Several methods [Wang *et al.*, 2023b; Fatemi *et al.*, 2023; Liu and Wu, 2023; Zhang *et al.*, 2023a] directly employ numerically organized node and edge lists to depict the graph data in plain text. GraphTMI [Das *et al.*, 2023] further explores different modalities such as motif and image to integrate graph data with LLMs.

The use of natural narration to express graph structures is also making steady progress. [Chen *et al.*, 2023] and [Hu *et al.*, 2023] both integrate the structural information of citation networks into the prompts, which is achieved by explicitly representing the edge relationship through the word "cite" and representing the nodes using paper indexes or titles. [Huang *et al.*, 2023a], on the other hand, does not use the word "cite" to represent edges but instead describes the relationships via enumerating randomly selected k-hop neighbors of the current node. Similarly, InstructGLM [Ye *et al.*, 2023] designs a series of scalable prompts based on the maximum hop level. These prompts allow a central paper node to establish direct associa-

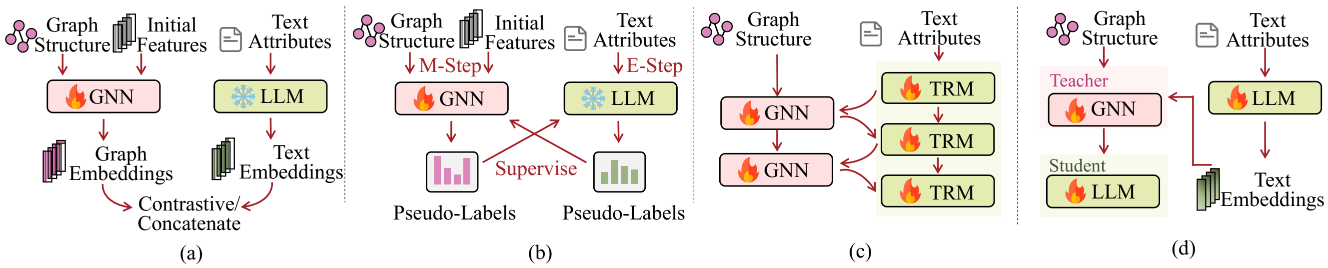


Figure 5: The illustration of GNN-LLM-Alignment approaches.

tions with its neighbors up to any desired hop level by utilizing the described connectivity relationships expressed in natural language. In addition, GPT4Graph [Guo *et al.*, 2023] and [Chen *et al.*, 2023] imitate the aggregation behavior of GNNs and summarize the current neighbor's attributes as additional inputs, aiming to provide more structural information.

4.2 GNN-based Prediction

GNNs have demonstrated impressive capabilities in understanding graph structures through recursive information exchange and aggregation among nodes. As illustrated in Figure 4(b), in contrast to flatten-based prediction, which converts graph data into textual descriptions as inputs to LLMs, GNN-based prediction leverages the advantages of GNNs to incorporate inherent structural characteristics and dependencies present in graph data with LLMs, allowing LLMs to be structure-aware. GNN-based prediction also relies on a parser to extract the output from LLMs. Integrating GNN representations into LLMs often requires tuning, making it easier to standardize the prediction format of LLMs by providing desirable outputs during training.

Various strategies have been proposed to fuse the structural patterns learned by GNNs and the contextual information captured by LLMs. For instance, GIT-Mol [Liu *et al.*, 2023b] and MolCA [Liu *et al.*, 2023c] both implement BLIP-2's Q-Former [Li *et al.*, 2023a] as the cross-modal projector to map the graph encoder's output to the LLM's input text space. Multiple objectives with different attention masking strategies are employed for effective graph-text interactions. GraphLLM [Chai *et al.*, 2023] derives the graph-enhanced prefix by applying a linear projection to the graph representation during prefix tuning, allowing the LLM to synergize with the graph transformer to incorporate structural information crucial to graph reasoning. Additionally, both GraphGPT [Tang *et al.*, 2023] and InstructMol [Cao *et al.*, 2023] employ a simple linear layer as the lightweight alignment projector to map the encoded graph representation to some graph tokens, while the LLM excels at aligning these tokens with diverse text information. DGTL [Qin *et al.*, 2023] injects the disentangled graph embeddings directly into each layer of the LLM, highlighting different aspects of the graph's topology and semantics.

4.3 Discussions

Utilizing LLMs directly as predictors shows superiority in processing textual attributes of graphs, especially achieving remarkable zero-shot performance compared with traditional GNNs. The ultimate goal is to develop and refine methods for encoding graph-structured information into a format that LLMs can comprehend and manipulate effectively and efficiently. Flatten-based prediction may have an advantage in terms of effectiveness, while GNN-based prediction tends to be more efficient. In flatten-based prediction, the input length limitation of LLMs restricts each node's access to only its neighbors within a few hops, making it challenging to capture long-range dependencies. Additionally, without the involvement of GNNs, inherent issues of GNNs such as heterophily cannot be addressed. On the other hand, for GNN-based prediction, training an additional GNN module and inserting it into LLMs for joint training is challenging due to the problem of vanishing gradients in the early layers of deep transformers.

5 GNN-LLM Alignment

The alignment of GNNs and LLMs offers an efficient method for integrating graph and text data. This alignment retains the distinct capabilities of each encoder by synchronizing their embedding spaces at a certain point. We categorize alignment techniques into symmetric and asymmetric: symmetric alignment treats GNNs and LLMs equally, whereas asymmetric alignment prioritizes one over the other.

5.1 Symmetric

Symmetric alignment refers to the equal treatment of the graph and text modalities during alignments. These approaches ensure that the encoders of both modalities achieve comparable performance in their respective applications.

A typical symmetric alignment architecture shown in Figure 5(a) adopts a two-tower style, employing separate encoders to individually encode the graph and text. Interaction between these modalities occurs only once during alignment. Approaches like SAFER [Chandra *et al.*, 2020] typically use basic concatenation for combining these embeddings.

Recent developments in two-tower models increasingly leverage contrastive learning, similar to the CLIP [Radford *et al.*, 2021], for more effective alignment of different modalities. The methods generally involve a two-step process: initially extracting graph and text representations, followed by applying contrastive learning with a modified InfoNCE loss [Oord *et al.*, 2018]. Text2Mol [Edwards *et al.*, 2021] introduces a cross-modal attention mechanism for early fusion of graph and text embeddings. It employs a transformer decoder, using LLM output as the source sequence and GNN output as the target

	Model	GNN	LLM	Predictor	Fine-tuning	Prompting	Domain	Task	Code
LLM as Enhancer	GIANT [Chien *et al.*, 2021]	SAGE, RevGAT, etc.	BERT	GNN	✗	✗	Citation, Co-purchase	Node	Link
	GALM [Xie *et al.*, 2023]	RGCN, RGAT	BERT	GNN	✓	✗	E-Commerce, Recommendation	Node, Link	-
	TAPE [He *et al.*, 2024]	RevGAT	ChatGPT	GNN	✗	✓	Citation	Node	Link
	Chen et al. [Chen *et al.*, 2023]	GCN, GAT	ChatGPT	GNN	✗	✓	Citation, Co-purchase	Node	-
	LLM4Mol [Qian *et al.*, 2023]	-	ChatGPT	LM	✗	✗	Molecular	Graph	Link
	SimTeG [Duan *et al.*, 2023]	SAGE, RevGAT, SEAL	allMiniLM-L6-v2, etc.	GNN	✓$^{\heartsuit}$	✗	Citation, Co-purchase	Node, Link	Link
	G-Prompt [Huang *et al.*, 2023b]	SAGE, RevGAT	RoBERTa-Large	GNN	✓	✓	Citation, Social	Node	-
	TouchUp-G [Zhu *et al.*, 2023]	SAGE, MB-GCN, etc.	BERT	GNN	✓	✗	Citation, Co-purchase, Recommendation	Node, Link	-
	OFA [Liu *et al.*, 2024]	R-GCN	Sentence-BERT	GNN	✗	✓	Citation, Web link, Knowledge, Molecular	Node, Link, Graph	Link
	LLMRec [Wei *et al.*, 2023]	LightGCN	ChatGPT	GNN	✗	✓	Recommendation	Recommendation	Link
	WalkLM [Tan *et al.*, 2023]	-	DistilRoBERTa	MLP	✓	✗	Knowledge	Node, Link	Link
	METERN [Jin *et al.*, 2023a]	-	BERT	LM	✓	✗	Citation, E-Commerce	Node	-
	LEADING [Xue *et al.*, 2023]	GCN, GAT	BERT	GNN	✓	✗	Citation	Node	-
LLM as Predictor	NLGraph [Wang *et al.*, 2023b]	-	Text-davinci-003	LLM	✗	✓	-	Reasoning	Link
	GPT4Graph [Guo *et al.*, 2023]	-	Text-davinci-003	LLM	✗	✓	-	Reasoning, Node, Graph	Link
	GIMLET [Zhao *et al.*, 2023a]	-	T5	LLM	✓/✗	✓	Molecular	Graph	Link
	Chen et al. [Chen *et al.*, 2023]	-	ChatGPT	LLM	✗	✓	Citation	Node	Link
	GIT-Mol [Liu *et al.*, 2023b]	GIN	MolT5	LLM	✓$^{\heartsuit}$	✓	Molecular	Graph, Captioning	-
	InstructGLM [Ye *et al.*, 2023]	-	FLAN-T5/LLaMA-v1	LLM	✓$^{\heartsuit}$	✓	Citation	Node	Link
	Liu et al. [Liu and Wu, 2023]	-	GPT-4, etc.	LLM	✗	✓	-	Reasoning	Link
	Huang et al. [Huang *et al.*, 2023a]	-	ChatGPT	LLM	✗	✓	Citation, Co-purchase	Node	Link
	GraphText [Zhao *et al.*, 2023b]	-	ChatGPT/GPT-4	LLM	✗	✓	Citation, Web link	Node	-
	Fatemi et al. [Fatemi *et al.*, 2023]	-	PaLM/PaLM 2	LLM	✗	✓	-	Reasoning	-
	GraphLLM [Chai *et al.*, 2023]	Graph Transformer	LLaMA-v2	LLM	✓$^{\heartsuit}$	✓	-	Reasoning	Link
	Hu et al. [Hu *et al.*, 2023]	-	ChatGPT/GPT-4	LLM	✗	✓	Citation, Knowledge, Social	Node, Link, Graph	-
	MolCA [Liu *et al.*, 2023c]	GINE	Galactica/MolT5	LLM	✓$^{\heartsuit}$	✓	Molecular	Graph, Retrieval, Captioning	Link
	GraphGPT [Tang *et al.*, 2023]	Graph Transformer	Vicuna	LLM	✓$^{\heartsuit}$	✓	Citation	Node	Link
	ReLM [Shi *et al.*, 2023]	TAG, GCN	Vicuna/ChatGPT	LLM	✗	✓	Molecular	Reaction Prediction	Link
	LLM4DyG [Zhang *et al.*, 2023a]	-	Vicuna/LLaMA-v2/ChatGPT	LLM	✗	✓	-	Reasoning	-
	DGTL [Qin *et al.*, 2023]	Disentangled GNN	LLaMA-v2	LLM	✓	✓	Citation, E-Commerce	Node	-
	GraphTMI [Das *et al.*, 2023]	-	GPT-4/GPT-4V	LLM	✗	✓	Citation	Node	-
	InstructMol [Cao *et al.*, 2023]	GIN	Vicuna	LLM	✓$^{\heartsuit}$	✓	Molecular	Graph, Captioning	Link
GNN-LLM Alignment	SAFER[Chandra *et al.*, 2020]	GCN, GAT, etc.	RoBERTa	Linear	✓	✗	News	Node	Link
	GraphFormers [Yang *et al.*, 2021]	Graph Transformer	UniLM	LLM	✓	✗	Citation, E-Commerce, Knowledge	Link	Link
	Text2Mol[Edwards *et al.*, 2021]	GCN	SciBERT	GNN/LLM	✓	✗	Molecular	Retrieval	Link
	MoMu [Su *et al.*, 2022]	GIN	BERT	GNN/LLM	✓	✗	Molecular	Graph, Retrieval	Link
	MoleculeSTM [Liu *et al.*, 2022]	GIN	BERT	GNN/LLM	✓	✗	Molecular	Graph, Retrieval	Link
	GLEM [Zhao *et al.*, 2022]	SAGE, RevGAT, etc.	DeBERTa	GNN/LLM	✓	✗	Citation, Co-purchase	Node	Link
	GRAD [Mavromatis *et al.*, 2023]	SAGE	SciBERT/DistilBERT	LLM	✓	✗	Citation, Co-purchase	Node	Link
	G2P2 [Wen and Fang, 2023]	GCN	Transformer	GNN/LLM	✓	✓	Citation, Recommendation	Node	Link
	Patton [Jin *et al.*, 2023b]	Graph Transformer	BERT/SciBERT	Linear/LLM	✓	✗	Citation, E-Commerce	Node, Link, Retrieval, Reranking	Link
	ConGraT [Brannon *et al.*, 2023]	GAT	all-mpnet-base-v2/DistilGPT2	GNN/LLM	✓	✗	Citation, Knowledge, Social	Node, Link	Link
	THLM [Zou *et al.*, 2023]	R-HGNN	BERT	LLM	✓	✗	Academic, Recommendation, Patent	Node, Link	Link
	GRENADE [Li *et al.*, 2023b]	SAGE, RevGAT-KD, etc.	BERT	GNN/MLP	✓	✗	Citation, Co-purchase	Node, Link	Link
	RLMRec [Ren *et al.*, 2023]	GCCF, LightGCN, etc.	ChatGPT, text-embedding-ada-002	GNN/LLM	✓	✗	Recommendation	Node	Link
Others	LLM-GNN [Chen *et al.*, 2024]	GCN, SAGE	ChatGPT	GNN	✗	✓	Citation, Co-purchase	Node	Link
	GPT4GNAS [Wang *et al.*, 2023a]	GCN, GIN, etc.	GPT-4	GNN	✗	✓	Citation	Node	-
	ENG [Yu *et al.*, 2023]	GCN, GAT	ChatGPT	GNN	✗	✓	Citation	Node	-

Table 1: A summary of models that leverage LLMs to assist graph-related tasks in literature, ordered by their release time. **Fine-tuning** denotes whether it is necessary to fine-tune the parameters of LLMs, and ♡ indicates that models employ parameter-efficient fine-tuning (PEFT) strategies, such as LoRA and prefix tuning. **Prompting** indicates the use of text-formatted prompts in LLMs, done manually or automatically. Acronyms in **Task**: Node refers to node-level tasks; Link refers to link-level tasks; Graph refers to graph-level tasks; Reasoning refers to Graph Reasoning; Retrieval refers to Graph-Text Retrieval; Captioning refers to Graph Captioning.

sequence. The decoder's output is used for contrastive learning, paired with GNN-processed outputs. MoMu [Su *et al.*, 2022], MoleculeSTM [Liu *et al.*, 2022], ConGraT [Brannon *et al.*, 2023], and RLMRec [Ren *et al.*, 2023] share a similar framework, which adopts paired graph embeddings and text embeddings to implement contrastive learning. MoMu and MoleculeSTM both source molecules from PubChem. MoMu pairs these with texts from scientific papers, whereas MoleculeSTM uses molecules' descriptions. ConGraT extends this graph-text pairing approach to social, knowledge, and citation networks. RLMRec aligns LLMs' semantic space with user-item interaction signals in recommendation systems. G2P2 [Wen and Fang, 2023] and GRENADE [Li *et al.*, 2023b] have further advanced the use of contrastive learning. G2P2 enhances the granularity by employing contrastive learning at three levels during the pre-training stage: node-text, text-text summary, and node-node summary. GRENADE is optimized by integrating graph-centric contrastive learning with dual-level graph-centric knowledge alignment, which includes both node-level and neighborhood-level alignment.

The iterative alignment, depicted in Figure 5(b), distinguishes itself by allowing iterative interaction between the modalities. For example, GLEM [Zhao *et al.*, 2022] employs the Expectation-Maximization (EM) framework, where one encoder iteratively generates pseudo-labels for the other, allowing the alignment between two representation spaces.

5.2 Asymmetric

Symmetric alignment balances both modalities equally, whereas asymmetric alignment prioritizes one, often using GNNs' structural processing to bolster LLMs. Current approaches mainly include graph-nested transformers and graph-aware distillation.

The graph-nested transformer, as exemplified by Graphformer [Yang *et al.*, 2021] in Figure 5(c), demonstrates asymmetric alignment through the integration of GNNs into each transformer layer. Within each layer of the LLM, the node embedding is obtained from the first token-level embedding, which corresponds to the `[CLS]` token. The process involves gathering embeddings from all relevant nodes and applying them to a graph transformer. The output is then concatenated with the input embeddings and passed on to the next layer of the LLM. Patton [Jin *et al.*, 2023b] extends GraphFormer by proposing network-contextualized masked language modeling and masked node prediction and shows strong performance in various downstream tasks, including classification, retrieval, reranking, and link prediction.

GRAD [Mavromatis *et al.*, 2023] employs graph-aware distillation for aligning two modalities, depicted in Figure 5(d). It utilizes a GNN as a teacher model to generate soft labels for an LLM. Since the LLMs share parameters, the GNN can also benefit from improved textual encodings after the updates to the LLMs' parameters. Through iterative updates, a graph-aware LLM is developed, resulting in enhanced scalability in inference due to the absence of the GNN. Similar to GRAD, THLM [Zou *et al.*, 2023] employs a heterogeneous GNN to enhance LLMs with multi-order topology learning capabilities. It involves pretraining a LLM alongside an auxiliary GNN through Context Graph Prediction and Masked Language Modeling tasks. After the pretraining process, the auxiliary GNN is discarded and the LLM is fine-tuned for downstream tasks.

5.3 Discussions

To align GNNs and LLMs, symmetric alignments treat each modality equally to both enhance GNNs and LLMs. This enables encoders to efficiently manage multimodal tasks, utilizing their unique strengths to enhance modality-specific representations. Asymmetric methods enhance LLMs by inserting graph encoders into transformers or using GNNs as teachers. However, alignment techniques face severe data scarcity, since only a few graph datasets contain native graph-text pairs, limiting the applicability of these methods.

6 Future Directions

Table 1 summarizes the models that leverage LLMs to assist graph-related tasks according to the proposed taxonomy. Based on the above review and analysis, we believe that there is still much space for further enhancement in this field.

Dealing with non-TAG. Text-attributed graphs (TAGs) have shown improved learning with LLM assistance, yet many real-world graphs lack textual information. For instance, traffic networks (e.g., PeMS03 [Song *et al.*, 2020]) use nodes for sensors, and superpixel graphs (e.g., PascalVOC-SP [Dwivedi *et al.*, 2022]) for superpixels, without text attributes for nodes, making semantic descriptions challenging. While OFA [Liu *et al.*, 2024] proposes using texts to describe nodes and edges, embedding them with LLMs isn't always effective across all domains. Research into utilizing LLMs for graph foundation models without rich text data remains promising.

Dealing with data leakage. Data leakage in LLMs has become a focal point of discussion. Given that LLMs undergo pre-training on extensive text corpora, it's likely that LLMs may have seen and memorized at least part of the test data of the common benchmark datasets, especially for citation networks. [Chen *et al.*, 2023] proves that specific prompts could potentially enhance the "activation" of LLMs' corresponding memory, thereby influencing the evaluation. Hence, it's crucial to reconsider the methods employed to accurately evaluate the performance of LLMs on graph-related tasks. A fair, systematic, and comprehensive benchmark is also needed.

Improving transferability. Transferability in graphs is tough because each graph has its unique features and structure. Differences in size, nodes, edges, and topology make it hard to apply what's learned from one graph to another. While LLMs have demonstrated promising zero/few-shot abilities in language tasks due to their extensive pre-training on vast amounts of corpora, using their knowledge for graph tasks isn't well-explored. OFA [Liu *et al.*, 2024] tries to make this easier by describing graphs in a unified way for better cross-domain transferability. However, improving the transferability in graphs is still an open area for more research.

Improving explainability. Explainability, also known as interpretability, denotes the ability to explain or present the behavior of models in human-understandable terms. LLMs exhibit improved explainability compared to GNNs when handling graph-related tasks, primarily due to the reasoning and explaining ability of LLMs to produce user-friendly explanations for graph reasoning. Several studies have examined explaining techniques within the prompting paradigm, such as in-context learning [Radford *et al.*, 2021] and chain-of-thought [Wei *et al.*, 2022b]. Further explorations should be conducted to enhance explainability.

Improving efficiency. While LLMs have demonstrated their effectiveness in learning on graphs, they may face inefficiencies in terms of time and space, particularly compared to dedicated graph learning models such as GNNs that inherently process graph structures. This is especially obvious when LLMs rely on sequential graph descriptions for predictions discussed in Section 4. Existing studies have tried to enable LLMs' efficient adaption via adopting parameter-efficient fine-tuning strategies, such as LoRA [Hu *et al.*, 2021] and prefix tuning [Li and Liang, 2021]. We believe that more efficient methods may unlock more power of applying LLMs on graph-related tasks with limited computational resources.

Analysis and improvement of expressive ability. Despite the recent achievements of LLMs in graph-related tasks, their theoretical expressive power remains largely unexplored. It is widely acknowledged that standard message-passing neural networks are as expressive as the 1-Weisfeiler-Lehman (WL) test, meaning that they fail to distinguish non-isomorphic graphs under 1-hop aggregation [Xu *et al.*, 2018]. Therefore, two fundamental questions arise: How effectively do LLMs understand graph structures? Can their expressive ability surpass those of GNNs or the WL-test?

7 Conclusion

The application of LLMs to graph-related tasks has emerged as a prominent area of research in recent years. In this survey, we aim to provide an in-depth overview of existing strategies for adapting LLMs to graphs. Firstly, we introduce a novel taxonomy that categorizes techniques involving both graph and text modalities into three categories based on the different roles played by LLMs, i.e., enhancer, predictor, and alignment component. Secondly, we systematically review the representative studies according to the taxonomy. Finally, we discuss some limitations and highlight several future research directions. Through this comprehensive review, we aspire to shed light on the advancements and challenges in the field of graph learning with LLMs, thereby encouraging further enhancements in this domain.

Acknowledgements

This work was supported by NSFC Grant No. 62206067, HKUST-HKUST(GZ) 20 for 20 Cross campus Collaborative

Research Scheme C019, Guangzhou-HKUST(GZ) Joint Funding Scheme 2023A03J0673, project MMT-p2-23 of the Shun Hing Institute of Advanced Engineering, the Research Grant Council of the Hong Kong Special Administrative Region, China (No. CUHK 14217622).

Contribution Statement

Yuhan Li, Zhixun Li, and Peisong Wang contribute equally to this work. Jia Li is the corresponding author (jialee@ust.hk).

References

[Brannon *et al.*, 2023] William Brannon, Suyash Fulay, et al. Congrat: Self-supervised contrastive pretraining for joint graph and text embeddings. *arXiv preprint arXiv:2305.14321*, 2023.

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020.

[Cao *et al.*, 2023] He Cao, Zijing Liu, et al. Instructmol: Multi-modal integration for building a versatile and reliable molecular assistant in drug discovery. *arXiv preprint arXiv:2311.16208*, 2023.

[Chai *et al.*, 2023] Ziwei Chai, Tianjie Zhang, et al. Graphllm: Boosting graph reasoning ability of large language model. *arXiv preprint arXiv:2310.05845*, 2023.

[Chandra *et al.*, 2020] Shantanu Chandra, Pushkar Mishra, et al. Graph-based modeling of online communities for fake news detection. *arXiv preprint arXiv:2008.06274*, 2020.

[Chen *et al.*, 2023] Zhikai Chen, Haitao Mao, et al. Exploring the potential of large language models (llms) in learning on graphs. *arXiv preprint arXiv:2307.03393*, 2023.

[Chen *et al.*, 2024] Zhikai Chen, Haitao Mao, et al. Label-free node classification on graphs with large language models (llms). In *ICLR*, 2024.

[Chien *et al.*, 2021] Eli Chien, Wei-Cheng Chang, et al. Node feature extraction by self-supervised multi-scale neighborhood prediction. *arXiv preprint arXiv:2111.00064*, 2021.

[Chowdhery *et al.*, 2022] Aakanksha Chowdhery, Sharan Narang, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[Das *et al.*, 2023] Debarati Das, Ishaan Gupta, et al. Which modality should i use–text, motif, or image?: Understanding graphs with large language models. *arXiv preprint arXiv:2311.09862*, 2023.

[Duan *et al.*, 2023] Keyu Duan, Qian Liu, et al. Simteg: A frustratingly simple approach improves textual graph learning. *arXiv preprint arXiv:2308.02565*, 2023.

[Dwivedi *et al.*, 2022] Vijay Prakash Dwivedi, Ladislav Rampášek, et al. Long range graph benchmark. *NeurIPS*, 35:22326–22340, 2022.

[Edwards *et al.*, 2021] Carl Edwards, ChengXiang Zhai, et al. Text2mol: Cross-modal molecule retrieval with natural language queries. In *EMNLP*, pages 595–607, 2021.

[Fatemi *et al.*, 2023] Bahare Fatemi, Jonathan Halcrow, et al. Talk like a graph: Encoding graphs for large language models. *arXiv preprint arXiv:2310.04560*, 2023.

[Guo *et al.*, 2023] Jiayan Guo, Lun Du, et al. Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking. *arXiv preprint arXiv:2305.15066*, 2023.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, et al. Inductive representation learning on large graphs. *NeurIPS*, 30, 2017.

[He *et al.*, 2024] Xiaoxin He, Xavier Bresson, et al. Explanations as features: Llm-based features for text-attributed graphs. In *ICLR*, 2024.

[Hu *et al.*, 2021] Edward J Hu, Phillip Wallis, et al. Lora: Low-rank adaptation of large language models. In *ICLR*, 2021.

[Hu *et al.*, 2023] Yuntong Hu, Zheng Zhang, et al. Beyond text: A deep dive into large language models' ability on understanding graph data. *arXiv preprint arXiv:2310.04944*, 2023.

[Huang *et al.*, 2023a] Jin Huang, Xingjian Zhang, et al. Can llms effectively leverage graph structural information: when and why. *arXiv preprint arXiv:2309.16595*, 2023.

[Huang *et al.*, 2023b] Xuanwen Huang, Kaiqiao Han, et al. Prompt-based node feature extractor for few-shot learning on text-attributed graphs. *arXiv preprint arXiv:2309.02848*, 2023.

[Ji *et al.*, 2021] Shaoxiong Ji, Shirui Pan, et al. A survey on knowledge graphs: Representation, acquisition, and applications. *TNNLS*, 33(2):494–514, 2021.

[Jin *et al.*, 2023a] Bowen Jin, Wentao Zhang, et al. Learning multiplex embeddings on text-rich networks with one text encoder. *arXiv preprint arXiv:2310.06684*, 2023.

[Jin *et al.*, 2023b] Bowen Jin, Wentao Zhang, et al. Patton: Language model pretraining on text-rich networks. *arXiv preprint arXiv:2305.12268*, 2023.

[Kenton and others, 2019] Jacob Devlin Ming-Wei Chang Kenton et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2016.

[Li and Liang, 2021] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*, pages 4582–4597, 2021.

[Li *et al.*, 2023a] Junnan Li, Dongxu Li, et al. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[Li *et al.*, 2023b] Yichuan Li, Kaize Ding, et al. Grenade: Graph-centric language model for self-supervised representation learning on text-attributed graphs. *arXiv preprint arXiv:2310.15109*, 2023.

[Liu and Wu, 2023] Chang Liu and Bo Wu. Evaluating large language models on graphs: Performance insights and comparative analysis. *arXiv preprint arXiv:2308.11224*, 2023.

[Liu *et al.*, 2022] Shengchao Liu, Weili Nie, et al. Multi-modal molecule structure-text model for text-based retrieval and editing. *arXiv preprint arXiv:2212.10789*, 2022.

[Liu *et al.*, 2023a] Jiawei Liu, Cheng Yang, et al. Towards graph foundation models: A survey and beyond. *arXiv preprint arXiv:2310.11829*, 2023.

[Liu *et al.*, 2023b] Pengfei Liu, Yiming Ren, et al. Git-mol: A multi-modal large language model for molecular science with graph, image, and text. *arXiv preprint arXiv:2308.06911*, 2023.

[Liu *et al.*, 2023c] Zhiyuan Liu, Sihang Li, et al. Molca: Molecular graph-language modeling with cross-modal projector and uni-modal adapter. *arXiv preprint arXiv:2310.12798*, 2023.

[Liu *et al.*, 2024] Hao Liu, Jiarui Feng, et al. One for all: Towards training one graph model for all classification tasks. In *ICLR*, 2024.

[Mavromatis *et al.*, 2023] Costas Mavromatis, Vassilis N Ioannidis, et al. Train your own gnn teacher: Graph-aware distillation on textual graphs. *arXiv preprint arXiv:2304.10668*, 2023.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, et al. Distributed representations of words and phrases and their compositionality. *NeurIPS*, 26, 2013.

[Oord *et al.*, 2018] Aaron van den Oord, Yazhe Li, et al. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[OpenAI, 2023] OpenAI. Gpt-4 technical report, 2023.

[Qian *et al.*, 2023] Chen Qian, Huayi Tang, et al. Can large language models empower molecular property prediction? *arXiv preprint arXiv:2307.07443*, 2023.

[Qin *et al.*, 2023] Yijian Qin, Xin Wang, et al. Disentangled representation learning with large language models for text-attributed graphs. *arXiv preprint arXiv:2310.18152*, 2023.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, et al. Learning transferable visual models from natural language supervision. In *ICLR*, pages 8748–8763, 2021.

[Ren *et al.*, 2023] Xubin Ren, Wei Wei, et al. Representation learning with large language models for recommendation. *arXiv preprint arXiv:2310.15950*, 2023.

[Salton and Buckley, 1988] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *IPM*, 24(5):513–523, 1988.

[Sen *et al.*, 2008] Prithviraj Sen, Galileo Namata, et al. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[Shayegani *et al.*, 2023] Erfan Shayegani, Md Abdullah Al Mamun, et al. Survey of vulnerabilities in large language models revealed by adversarial attacks. *arXiv preprint arXiv:2310.10844*, 2023.

[Shi *et al.*, 2023] Yaorui Shi, An Zhang, et al. Relm: Leveraging language models for enhanced chemical reaction prediction. *arXiv preprint arXiv:2310.13590*, 2023.

[Song *et al.*, 2020] Chao Song, Youfang Lin, et al. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *AAAI*, pages 914–921, 2020.

[Su *et al.*, 2022] Bing Su, Dazhao Du, et al. A molecular multimodal foundation model associating molecule graphs with natural language. *arXiv preprint arXiv:2209.05481*, 2022.

[Tan *et al.*, 2023] Yanchao Tan, Zihao Zhou, et al. Walklm: A uniform language model fine-tuning framework for attributed graph embedding. In *NeurIPS*, 2023.

[Tang *et al.*, 2023] Jiabin Tang, Yuhao Yang, et al. Graphgpt: Graph instruction tuning for large language models. *arXiv preprint arXiv:2310.13023*, 2023.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, et al. Attention is all you need. *NeurIPS*, 30, 2017.

[Velickovic *et al.*, 2018] Petar Velickovic, Guillem Cucurull, et al. Graph attention networks. *stat*, 1050:4, 2018.

[Wang *et al.*, 2023a] Haishuai Wang, Yang Gao, et al. Graph neural architecture search with gpt-4. *arXiv preprint arXiv:2310.01436*, 2023.

[Wang *et al.*, 2023b] Heng Wang, Shangbin Feng, et al. Can language models solve graph problems in natural language? *arXiv preprint arXiv:2305.10037*, 2023.

[Wei *et al.*, 2022a] Jason Wei, Yi Tay, et al. Emergent abilities of large language models. *TMLR*, 2022.

[Wei *et al.*, 2022b] Jason Wei, Xuezhi Wang, et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 35:24824–24837, 2022.

[Wei *et al.*, 2023] Wei Wei, Xubin Ren, et al. Llmrec: Large language models with graph augmentation for recommendation. *arXiv preprint arXiv:2311.00423*, 2023.

[Wen and Fang, 2023] Zhihao Wen and Yuan Fang. Prompt tuning on graph-augmented low-resource text classification. *arXiv preprint arXiv:2307.10230*, 2023.

[Wu *et al.*, 2018] Zhenqin Wu, Bharath Ramsundar, et al. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[Xie *et al.*, 2023] Han Xie, Da Zheng, et al. Graph-aware language model pre-training on a large graph corpus can help multiple graph applications. *arXiv preprint arXiv:2306.02592*, 2023.

[Xu *et al.*, 2018] Keyulu Xu, Weihua Hu, et al. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[Xue *et al.*, 2023] Rui Xue, Xipeng Shen, et al. Efficient large language models fine-tuning on graphs. *arXiv preprint arXiv:2312.04737*, 2023.

[Yang *et al.*, 2021] Junhan Yang, Zheng Liu, et al. Graphformers: Gnn-nested transformers for representation learning on textual graph. *NeurIPS*, 34:28798–28810, 2021.

[Yang *et al.*, 2023] Jingfeng Yang, Hongye Jin, et al. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *arXiv preprint arXiv:2304.13712*, 2023.

[Ye *et al.*, 2023] Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. Natural language is all a graph needs. *arXiv preprint arXiv:2308.07134*, 2023.

[Yu *et al.*, 2023] Jianxiang Yu, Yuxiang Ren, et al. Empower text-attributed graphs learning with large language models (llms). *arXiv preprint arXiv:2310.09872*, 2023.

[Zhang *et al.*, 2023a] Zeyang Zhang, Xin Wang, et al. Llm4dyg: Can large language models solve problems on dynamic graphs? *arXiv preprint arXiv:2310.17110*, 2023.

[Zhang *et al.*, 2023b] Ziwei Zhang, Haoyang Li, et al. Large graph models: A perspective. *arXiv preprint arXiv:2308.14522*, 2023.

[Zhao *et al.*, 2022] Jianan Zhao, Meng Qu, et al. Learning on large-scale text-attributed graphs via variational inference. *arXiv preprint arXiv:2210.14709*, 2022.

[Zhao *et al.*, 2023a] Haiteng Zhao, Shengchao Liu, et al. Gimlet: A unified graph-text model for instruction-based molecule zero-shot learning. *arXiv preprint arXiv:2306.13089*, 2023.

[Zhao *et al.*, 2023b] Jianan Zhao, Le Zhuo, et al. Graphtext: Graph reasoning in text space. *arXiv preprint arXiv:2310.01089*, 2023.

[Zhao *et al.*, 2023c] Wayne Xin Zhao, Kun Zhou, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[Zhu *et al.*, 2023] Jing Zhu, Xiang Song, et al. Touchup-g: Improving feature representation through graph-centric finetuning. *arXiv preprint arXiv:2309.13885*, 2023.

[Zou *et al.*, 2023] Tao Zou, Le Yu, et al. Pretraining language models with text-attributed heterogeneous graphs. *arXiv preprint arXiv:2310.12580*, 2023.