Graph Machine Learning in the Era of Large Language Models (LLMs)

Wenqi Fan, Shijie Wang, Jiani Huang, Zhikai Chen, Yu Song, Wenzhuo Tang,
Haitao Mao, Hui Liu, Xiaorui Liu, Dawei Yin, Qing Li

Abstract—Graphs play an important role in representing complex relationships in various domains like social networks, knowledge graphs, and molecular discovery. With the advent of deep learning, Graph Neural Networks (GNNs) have emerged as a cornerstone in Graph Machine Learning (Graph ML), facilitating the representation and processing of graph structures. Recently, LLMs have demonstrated unprecedented capabilities in language tasks and are widely adopted in a variety of applications such as computer vision and recommender systems. This remarkable success has also attracted interest in applying LLMs to the graph domain. Increasing efforts have been made to explore the potential of LLMs in advancing Graph ML's generalization, transferability, and few-shot learning ability. Meanwhile, graphs, especially knowledge graphs, are rich in reliable factual knowledge, which can be utilized to enhance the reasoning capabilities of LLMs and potentially alleviate their limitations such as hallucinations and the lack of explainability. Given the rapid progress of this research direction, a systematic review summarizing the latest advancements for Graph ML in the era of LLMs is necessary to provide an in-depth understanding to researchers and practitioners. Therefore, in this survey, we first review the recent developments in Graph ML. We then explore how LLMs can be utilized to enhance the quality of graph features, alleviate the reliance on labeled data, and address challenges such as graph heterogeneity and out-of-distribution (OOD) generalization. Afterward, we delve into how graphs can enhance LLMs, highlighting their abilities to enhance LLM pre-training and inference. Furthermore, we investigate various applications and discuss the potential future directions in this promising field.

consider a task to classify cat breed Photographs; photograph of cats would be in distribution, while photographs of dogs, humans, balls and other objects would be out-of-distribution.

Index Terms—Graph Machine Learning, Graph Foundation Models, Graph Learning, Large Language Models (LLMs), Pre-training and Fine-tuning, Prompting, Representation Learning.

1 Introduction

Graph data are widespread in many real-world applications [1], [2], including social graphs, knowledge graphs, and recommender systems [3]–[5]. Typically, graphs consist of nodes and edges, e.g., in a social graph, nodes represent users and edges represent relationships [6], [7]. In addition to the topological structure, graphs tend to possess various features of nodes, such as textual description, which provide valuable context and semantic information about nodes. To effectively model the graph, *Graph Machine Learning (Graph ML)* has garnered significant interest. With the advent of deep learning (DL), Graph Neural Networks (GNNs) have become a critical technique in Graph ML due to their message-passing mechanism. This mechanism allows each node to obtain its representation by recursively receiving and aggregating messages from neighboring nodes [8], [9], thereby capturing the high-order relationships and dependencies within the graph structure. To mitigate the reliance on supervised data, many research focused on developing self-supervised Graph ML methods to advance GNNs to capture transferable graph patterns, enhancing their generalization capabilities across various tasks [10]–[13]. Given the exponential growth of applications of graph data, researchers are actively working to develop more powerful Graph ML methods.

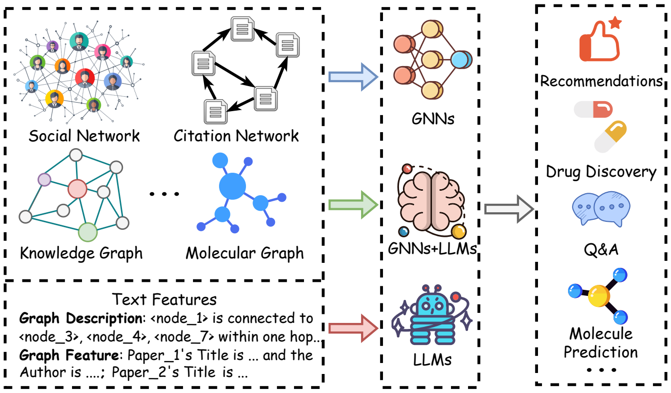
Social Network
Citation Network
Knowledge Graph
Molecular Graph
Text Features
Graph Description: <node_1> is connected to <node_3>, <node_4>, <node_7> within one hop...
Graph Feature: Paper_1's Title is ... and the Author is ....; Paper_2's Title is ...
GNNs
GNNs+LLMs
LLMs
Recommendations
Drug Discovery
Q&A
Molecule Prediction

Figure 1: Illustration of the application of Large Language Models (LLMs) in graph machine learning. The integration of LLMs with Graph Neural Networks (GNNs) is utilized to model an extensive range of graph data across various downstream tasks.

Recently, Large Language Models (LLMs) have started a new trend of AI and have shown remarkable capabilities in natural language processing (NLP) [14], [15]. With the evolution of these models, LLMs are not only being applied

- W. Fan is with the Department of Computing (COMP) and Department of Management and Marketing (MM), The Hong Kong Polytechnic University. E-mail: wenqifan03@gmail.com.
- S. Wang, and Q. Li are with the Department of Computing, The Hong Kong Polytechnic University. E-mail: shijie.wang@connect.polyu.hk; csqli@comp.polyu.edu.hk.
- J. Huang is with Wuhan University. E-mail: huangjiani@whu.edu.cn.
- Z. Chen, Y. Song, W. Tang, H. Mao, and H. Liu are with Michigan State University. E-mail: {chenzh85, songyu5, tangwen2, haitaoma, liuhui7}@msu.edu.
- X. Liu is with North Carolina State University. E-mail: xliu96@ncsu.edu.
- D. Yin is Senior Director of Engineering at Baidu Inc. E-mail: yindawei@acm.org.

(Corresponding authors: Wenqi Fan and Qing Li.)

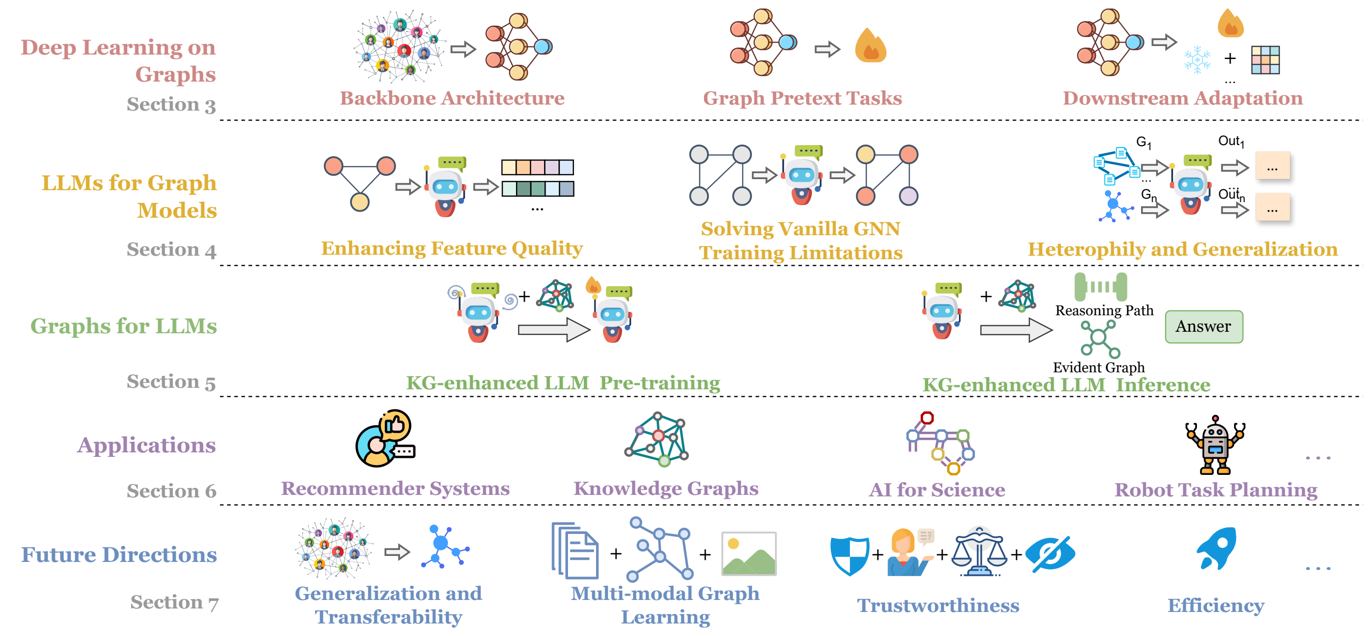

Figure 2: The outline of our survey. **Section 3 Deep Learning on Graphs** explores the development of DNN-based methods, focusing on the Backbone Architecture, Graph Pretext Tasks, and Downstream Adaption three aspects. **Section 4 LLMs for Graph Models** explore how current LLMs help the current Graph ML towards GFMs from Enhancing Feature Quality, Solving Vanilla GNN Training Limitations, and Heterophily and Generalization three aspects. **Section 5 Graph for LLMs** focuses on Knowledge Graph(KG)-enhanced LLM Pre-training and KG-enhanced LLM Inference. **Section 6 Applications** presents various applications, including Recommender System, Knowledge Graph, AI for Science, and Robot Task Planning. **Section 7 Future Directions** discusses potential future directions for LLMs in graph machine learning from the Generalization and Transferability, Multi-modal Graph Learning, Trustworthiness and Efficiency.

to language tasks but also showcasing great potentials in various applications such as CV [16], and Recommender System [17]. The effectiveness of LLMs in complex tasks is attributed to their extensive scale in both architecture and dataset size. For example, GPT-3 with 175 billion parameters demonstrates exciting capabilities by generating human-like text, answering complex questions, and coding. Furthermore, LLMs are able to grasp extensive general knowledge and sophisticated reasoning due to their vast training datasets. Therefore, their abilities in linguistic semantics and knowledge reasoning enable them to learn semantic information. Additionally, LLMs exhibit emergence abilities, excelling in new tasks and domains with limited or no specific training. This attribute is expected to provide high generalisability across different downstream datasets and tasks even in few-shot or zero-shot situations. Therefore, leveraging the capabilities of LLMs in Graph Machine Learning (Graph ML) has gained increasing interest and is expected to enhance Graph ML towards Graph Foundation Models (GFMs) [18].

GFMs are generally trained on extensive data and can be adapted for a wide range of downstream tasks [19]. By exploiting the ability of LLMs, it is expected to enhance the ability of Graph ML to generalize a variety of tasks, thus facilitating GFMs. Currently, researchers have made several initial efforts to explore the potential of LLMs in advancing Graph ML towards GFMs. Figure 1 demonstrates an example of integrating LLMs and GNNs for various graph tasks. Firstly, some methods leverage LLMs to alleviate the reliance of vanilla Graph ML on labeled data, where they make inferences based on implicit and explicit graph structure information [20]–[22]. For instance, InstructGLM [20] fine-tunes models like LlaMA [23] and T5 [24] by serializing graph data as tokens and encoding structural information about the graph to solve graph tasks. Secondly, to overcome the challenge of feature quality, some methods further employ LLMs to enhance the quality of graph features [25]–[27]. For example, SimTeG [25] fine-tunes LLMs on textual graphs datasets to obtain textual attribute embeddings, which are then utilized to augment the GNN for various downstream tasks. Additionally, some studies explore using LLMs to address challenges such as heterogeneity [28] and OOD [26] of graphs.

On the other hand, although LLM achieves great success in various fields, it still faces several challenges, including hallucinations, actuality awareness, and lacking explainability [29]–[31]. Graphs, especially knowledge graphs, capture extensive high-quality and reliable factual knowledge in a structured format [5]. Therefore, incorporating graph structure into LLMs could improve the reasoning ability of LLMs and mitigate these limitations. To this end, efforts have been made to explore the potential of graphs in augmenting LLMs' explainability [32], [33] and mitigating hallucination [34], [35]. Given the rapid evolution and significant potential of this field, a thorough review of recent advancements in graph applications and Graph ML in the era of LLMs is imperative.

Therefore, in this survey, we aim to provide a comprehensive review of Graph Machine Learning in the era of LLMs. The outline of the survey is shown in Figure 2. Section 2 reviews work related to graph machine learning and foundation models. Section 3 introduces the deep learning methods on graphs, which focus on various GNN models and self-supervised methods. Subsequently, the survey delves into how LLMs can be used to enhance Graph ML in Section 4 and how graphs can be adopted for augmenting LLMs in Section 5. Finally, some applications and potential future directions for Graph ML in the era of LLMs are discussed in Section 6 and Section 7, respectively. Our main contributions can be summarized as follows:

- We detail the evolution from early graph learning methods to the latest GFMs in the era of LLMs;
- We provide a comprehensive analysis of current LLMs enhanced Graph ML methods, highlighting their advantages and limitations, and offering a systematic categorization;
- We thoroughly investigate the potential of graph structures to address the limitations of LLMs;
- We explore the applications and prospective future directions of Graph ML in the era of LLMs, and discuss both research and practical applications in various fields.

Concurrent to our survey, Wei *et al.* [36] review the development of graph learning. Zhang *et al.* [37] provide a prospective review of large graph models. Jin *et al.* [38] and Li *et al.* [39] review different techniques for pre-training language models (in particular LLMs) on graphs and applications to different types of graphs, respectively. Liu *et al.* [40] review the Graph Foundation Models according to the pipelines. Mao *et al.* [18] focus on the fundamental principles and discuss the potential of GFMs. Different from these concurrent surveys, our survey provides a more comprehensive review with the following differences: (1) we present a more systematic review of the development of Graph Machine Learning and further exploration of LLMs for Graph ML towards GFMs; (2) we present a more comprehensive and fine-grained taxonomy of recent advancements of Graph ML in the era of LLMs; (3) we delve into the limitations of recent Graph ML, and provide insights into how to overcome these limitations from LLM's perspective; (4) we further explore how graphs can be used to augment LLMs; and (5) we thoroughly summarize a broad range of applications and present a more forward-looking discussion on the challenges and future directions.

2 RELATED WORK

In this section, we briefly review some related works in the fields of graph machine learning and foundation model techniques.

2.1 Graph Machine Learning

As one of the most active fields in artificial intelligence, graph learning has attracted considerable attention to its capability to model complex relationships and structures in data represented as graphs [41]. Nowadays, it has been widely adopted in various applications, including social network analysis [42], protein detection [43], recommender systems [44], [45], etc.

The initial phases of graph learning typically use Random Walks, which is a foundational method for exploring graph structures. This technique involves a stochastic process of moving from one node to another within a graph, which is instrumental in understanding node connectivity and influence within networks. Building upon Random Walks, Graph Embedding methods aim to represent nodes (or edges) as low-dimensional vectors while preserving graph topology and node relationships. Representative methods such as LINE [46], DeepWalk [47], and Node2Vec [48] leverage Random Walks to learn node representations, capturing local structures and community information effectively.

Due to the exceptional representation learning and modeling capabilities, GNNs bolstered by deep learning have brought significant advances in graph learning. For example, GCNs [49] introduce convolutional operations to graph data, enabling effective aggregation of neighborhood information for each node, thus enhancing node representation learning. GraphSAGE [50] learns a function to aggregate information from a node's local neighborhood in an inductive setting, allowing efficient embedding generation for unseen nodes. GAT [51] further advances GNNs by integrating attention mechanisms, assigning varying weights to nodes in a neighborhood, thereby sharpening the model's ability to focus on significant nodes. Inspired by the success of transformers [52] in NLP and CV, several studies [53]–[57] adopt self-attention mechanisms to graph data, providing a more global perspective of graph structures and interactions. Recent works [58]–[62] further leverage transformer architectures to enhance graph data modeling. For example, GraphFormer [58] integrates GNN within each layer in the transformer, enabling simultaneous consideration of textual and graph information.

The advancements in LLMs have given rise to graph learning. Recent works [20], [21], [26], [63] apply techniques from these advanced language models like LLaMA [23] or ChatGPT to graph data, resulting in models capable of understanding and handling graph structures in a manner similar to natural language processing. A typical approach, GraphGPT [22], tokenizes graph data for insertion into LLMs (i.e. Vicuna [64] and LLaMA [23]) thus providing a powerful generalization capability. GLEM [65] further integrates the graph models and LLMs, specifically DeBERTa [66], within a variational Expectation-Maximization (EM) framework. It alternates between updating LLM and GNN in the E-step and M-step, thereby scaling efficiently and improving effectiveness in downstream tasks.

2.2 Foundation Models (FMs)

Foundation Models (FMs) represent a significant breakthrough in the field of artificial intelligence, characterized by their ability to be extensively pre-trained on large-scale datasets and adapted to a variety of downstream tasks. These models are distinguished by their extensive pre-training on large-scale datasets and their adaptability to a wide range of downstream tasks. It is worth noting that FMs are not limited to a single field and can be found in natural language [14], [15], vision [67], [68], and graph domains [18], [40], serving as a promising research direction.

In the realm of vision, Visual Foundation Models (VFMs) have gained significant success, making substantial impacts on areas such as image recognition, object detection, and scene understanding. Specifically, VFMs benefit from pre-training on extensive and diverse image datasets, allowing them to learn intricate patterns and features. For instance, models such as DALL-E [69], and CLIP [67] leverage self-supervised learning to understand and generate images based on textual descriptions, demonstrating remarkable cross-modal understanding capabilities. Recent Visual Chat-GPT [68] integrates ChatGPT with a series of Visual Foundation Models (VFMs), making it perform a variety of complex visual tasks. These VFMs allow models to learn from a broader range of visual data, thereby improving their generalizability and robustness.

In the sphere of Natural Language Processing (NLP), Large Language Models (LLMs) such as ChatGPT and LLaMA have also revolutionized the field [70]. Characterized by their extensive scale, LLMs are trained on billions of parameters using extensive textual datasets, which enable them to excel in comprehending and generating natural language. The landscape of pre-trained language models is diverse, such as GPT (Generative Pre-trained Transformer) [14], BERT (Bidirectional Encoder Representations from Transformers) [15] and T5 (Text-To-Text Transfer Transformer) [24]. These models can broadly fall into three categories: encoder-only, decoder-only, and encoder-decoder models. Encoder-only models, such as BERT, specialize in understanding and interpreting language. In contrast, decoder-only models like GPT excel in generating coherent and contextually relevant text. Encoder-decoder models, like T5, combine both abilities, efficiently performing various NLP tasks from translation to summarization.

As an encoder-only model, BERT introduces a paradigm in NLP with its innovative bi-directional attention mechanism, which analyzes text from both directions simultaneously, unlike its predecessors like transformer which processed text in a single direction (either left-to-right or right-to-left). This feature allows BERT to attain a comprehensive context understanding, significantly improving its language nuance comprehension. On the other hand, decoder-only models such as GPT, including variants like ChatGPT, utilize a unidirectional self-attention mechanism. This design makes them particularly effective in predicting subsequent words in a sequence, thus excelling in tasks like text completion, creative writing, and code generation. Additionally, as an encoder-decoder model, T5 uniquely transforms a variety of NLP tasks as text generation problems. For example, it reframes sentiment analysis from a classification task to a text generation task, where input like ”Sentiment: Today is sunny” would prompt T5 to generate an output such as ”Positive”. This text-to-text approach underscores T5’s versatility and adaptability across diverse language tasks.

The evolution of LLMs has seen the emergence of advanced models like GPT-3 [92], LaMDA [93], PaLM [94], and Vicuna [64]. These models represent significant advances in NLP, distinguished by their enhanced capabilities in comprehending and generating complex, fine-grained language. Their training methods are usually more sophisticated, involving larger datasets and more powerful computational resources. This scaling up has led to unprecedented language understanding and generation capabilities, exhibiting emergent properties such as in-context learning (ICL), adaptability, and flexibility. Furthermore, recent advancements demonstrate the successful integration of LLMs with other models, like recommender system [17], reinforcement learning (RL) [95], GNNs [25], [96]–[98]. This integration enables LLMs to tackle both traditional and novel challenges, proposing prospective avenues for applications.

LLMs have found applications in diverse sectors like chemistry [99], [100], education [101], [102], and finance [103], [104]. In these fields, they contribute to various tasks from data analysis to personalized learning. Particularly, LLMs exhibit great potential in graph tasks such as graph classification and link prediction, demonstrating their versatility and broad applicability. Specifically, several studies like Simteg [25], GraD [97], Graph-Toolformer [96], and Graphologue [98] have notably advanced graph learning. These models utilize LLMs for textual graph learning, graph-aware distillation, and graph reasoning, illustrating the potential of LLMs in enhancing the understanding of and interaction with complex graph structures.

Although FMs have revolutionized Vision and NLP domains, the development of Graph Foundation Models (GFMs) is still in the nascent stages. With the rapid evolution and significant potential of this field, it is imperative to continue exploring and developing advanced techniques that can further enhance Graph ML towards GFMs.

3 Deep Learning on Graphs

With the rapid development of deep neural networks (DNNs), GNN techniques modeling graph structure and node attributes for representation learning have been widely explored and have become one key technology in Graph ML. While vanilla GNNs demonstrate proficiency in various graph tasks, they still encounter several challenges such as scalability, generalization to unseen data, and limited capability in capturing complex graph structures. To overcome these limitations, many efforts have been made to improve GNN with the self-supervised paradigm. Therefore, to provide a comprehensive review of these methods, in this section, we first introduce the backbone architecture, including GNN-based models and graph transformer-based models. After that, we explore two important aspects of self-supervised graph ML models: graph pretext tasks and downstream adaptation. Note that a comprehensive summary of these methods is presented in Table 1.

3.1 Backbone Architecture

As one of the most active fields in the artificial intelligence (AI) community, various GNN methods have been proposed to solve various tasks. The powerful capability of these models is largely dependent on the development of their backbone architectures. Therefore, in this subsection, we focus on two broadly used architectures: neighborhood aggregation-based models and graph transformer-based models.

3.1.1 Neighborhood Aggregation-based Model

Neighborhood aggregation-based models are the most popular graph learning architectures that have been extensively studied and applied in various downstream tasks. These models operate based on the message-passing

Table 1: A comparison of various DNN-based models. We present Models and their Architecture, Pretext Task, Adaptation Method, and Downstream Tasks. **URL** in **Adaptation Method** indicates Unsupervised Representation Learning.

Model	Architecture	Pretext Task	Adaptation Method	Downstream Tasks
DGI [71]	GNN	Contrastive Learning	URL	Node
GRACE [72]	GNN	Contrastive Learning	URL	Node
GraphMAE [73]	GNN	Graph Generation	URL	Node, Graph
MVGRL [74]	GNN	Contrastive Learning	URL	Node, Graph
GraphCL [10]	GNN	Contrastive Learning	Fine-tuning	Node, Graph
CSSL [11]	GNN	Contrastive Learning	Fine-tuning	Graph
GCC [13]	GNN	Contrastive Learning	URL&Fine-tuning	Node, Graph
G-BERT [75]	BERT	Graph Generation	Fine-tuning	Recommendation
AdapterGNN [76]	GNN	Multi-task	Fine-tuning	Graph
GROVER [77]	Graph Transformer	Property Prediction	Fine-tuning	Graph
Graph-Bert [57]	Graph Transformer	Graph Generation	URL&Fine-tuning	Node
G-Adapter [78]	Graph Transformer	Multi-task	Fine-tuning	Graph
GraphGPT [79]	Graph Transformer	Graph Generation	Fine-tuning	Node, Edge, Graph
MoMu [80]	BERT, GNN	Contrastive Learning	Fine-tuning	Graph
TOUCHUP-G [81]	BERT, ViT, GNN	Contrastive Learning	Fine-tuning	Node, Edge
GraphPrompt [82]	GNN	Contrastive Learning	Prompt Tuning	Node, Graph
GPPT [83]	GNN	Contrastive Learning	Prompt Tuning	Node
PGCL [84]	GNN	Contrastive Learning	Prompt Tuning	Node, Edge, Graph
GPF [85]	GNN	Multi-task	Prompt Tuning	Graph
ProG [86]	GNN, Graph Transformer	Contrastive Learning	Prompt Tuning	Node, Edge, Graph
ULTRA-DP [87]	GNN	Multi-task	Prompt Tuning	Node
SAP [88]	GNN	Contrastive Learning	Prompt Tuning	Node, Graph
PRODIGY [89]	GNN	Multi-task	Prompt Tuning	Node, Edge
SGL-PT [90]	GNN	Multi-task	Prompt Tuning	Node, Graph
DeepGPT [91]	Graph Transformer	Graph Regression	Prompt Tuning	Graph

mechanism [105], which updates a node's representation by aggregating the features of its neighboring nodes along with its own features. Formally, this process can be represented as:

$$m_u = Aggregate(f_v, v \in \mathcal{N}_u), \quad (1)$$

$$f'_u = Update(m_u, f_u), \quad (2)$$

where, for each node u, a message m_u is generated through the aggregation function from its neighboring nodes. Subsequently, the graph signal f is updated with the message.

GCN is a typical method designed to leverage both the graph structure and the node attributes. This architecture updates node representations by aggregating neighboring features with the node's own. As the number of network layers increases, each captures an increasingly larger neighborhood. Owing to the efficiency and performance, GCN [49] has been widely applied by several methods such as CSSL [11] and PRODIGY [89]. GraphSAGE [50] is another notable neighborhood aggregation-based model. Due to its inductive paradigms, GraphSAGE can easily generalize to unseen nodes or graphs, making it widely employed by many studies such as PinSage [106] for inductive learning. Additionally, several studies [73], [86], [89] incorporate Graph Attention Networks (GATs) [51] as the backbone architecture. GATs integrate attention mechanisms into GNNs, assigning variable weights to neighboring nodes, thereby focusing on the most relevant parts of the input graph for improved node representations. As another important model in the family of GNNs, Graph Isomorphism Network (GIN) [107] has also been widely used [10], [13], [82], [90], due to its powerful representation ability. Its unique architecture guarantees the expressiveness equivalent to the Weisfeiler-Lehman isomorphism test, making it widely chosen as the backbone model for a lot of structure-intensive tasks.

Although these models are widely adopted to solve graph tasks, they still suffer from some inherent limitations, such as over-smoothing and lack of generalization. In addition, the lower amount of parameters also limits the modeling capacity as the backbone model to serve multiple datasets and tasks.

3.1.2 Graph Transformer-based Model

While neighborhood aggregation-based GNN models have shown remarkable performance in processing graph-structured data, they suffer from some limitations. A significant challenge for these models is their difficulty in handling large graphs due to their reliance on local neighborhood information and their limited capacity in capturing long-range dependencies within the graph [61], [108], [109]. To overcome these problems, inspired by the success of the transformer model in various NLP tasks, graph transformer-based models have been proposed [54], [59], [61]. These models leverage the self-attention mechanism to adaptly capture both local and global graph structures, which allows the model to stack multiple layers without over-smoothing. Due to the lower inductive bias, graph transformer-based models can learn the structural patterns from data rather than solely relying on the graph structure. Additionally, transformers have demonstrated great scaling behavior in CV and NLP, suggesting that their performance can keep improving with more data and parameters.

Graph transformer-based models have been widely applied as a backbone architecture in various tasks [57], [77], [78], [91], [110]. For example, Graph-Bert [57] employs a transformer to pre-train on the graph dataset with feature and edge reconstruction tasks and then fine-tunes for various downstream tasks. Similarly, GROVER [77] introduces a

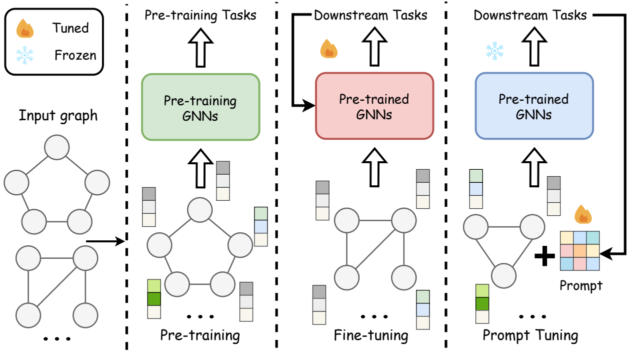

Figure 3: A comparison of pre-training, fine-tuning, and prompt tuning. (a) Pre-training involves training the GNN model based on specific pre-training tasks. (b) Fine-tuning updates the parameters of the pre-trained GNN model according to the downstream tasks. (c) Prompt tuning generates and updates the features of the prompt according to the downstream tasks, while keeping the pre-trained GNN model fixed and without any modification.

self-supervised graph transformer-based model designed specifically for large-scale molecular data. It pre-trains on extensive molecular datasets and then fine-tunes for specific downstream tasks. GraphGPT [79] employs a (semi-)Eulerian path to transform the graph into a sequence of tokens, and then feeds the sequence into the transformer. Specifically, it constructs a dataset-specific vocabulary such that each node can correspond to a unique node ID.

Despite graph transformer-based models that can somehow address the limitations of traditional GNNs, they also face several challenges. One of the challenges is the quadratic complexity caused by self-attention, which becomes particularly problematic for large-scale graphs. In addition, there is a risk of losing some information about the original graph structure when serializing the graph.

3.2 Self-Supervised Learning on Graphs

To adapt GNNs to various graph tasks, many self-supervised learning methods have been proposed and extensively studied. These approaches enable GNNs to learn graph representations from the pre-training task and transfer them to various downstream tasks, such as node classification, graph classification, and link prediction. Therefore, in this subsection, we will introduce graph self-supervised learning methods from pretext tasks and downstream adaptation, respectively.

3.2.1 *Graph Pretext Tasks*

Graph Contrastive Learning aims to learn augmentation representations by contrasting similar and dissimilar graph data pairs, effectively identifying nuanced relationships and structural patterns. We can review graph contrastive learning from two perspectives: graph augmentations and the scale of contrast.

the intentional manipulation or disturbance of data in a dataset during model training.

Generally, graph augmentations can be broadly categorized into two types: 1) *feature perturbation* and 2) *topology perturbation*. They assume that tiny changes in the feature or structural space do not change the semantics of the Node/Edge/(sub)graph. Feature perturbation involves perturbing the features of the nodes in the graph. For example, GRACE [72] randomly masks the node features to learn more robust representations. On the other hand, topology perturbation mainly involves modifying the structure of the graph. A typical example is CSSL [11] which employs strategies like edge perturbation or node dropping to adopt graph-graph level contrast, thereby enhancing the robustness of representations.

Regarding the scale of contrast, the approaches can be divided into node-level and graph-level. For example, GRACE [72] computes the similarities between node-level embeddings to learn discriminative node representations. GCC [13] also works at the node level but learns local structural patterns by sampling a node's neighbors to obtain subgraphs (positive pairs) and contrasting them with randomly selected non-contextual subgraphs (negative pairs). In contrast, DGI [71] contrasts node-level embeddings with graph-level embedding to capture global graph structures. GraphCL [10] takes a different approach by implementing graph-to-graph level contrast, thereby learning robust representations. The scale used for pre-training has a huge impact on the downstream performance. When adopting contrastive learning as the pre-training task, one key challenge is how to design the objective such that the embeddings learned can account for downstream tasks of different scales.

Graph Generation methods aim to learn the distribution of graph data to enable graph generation or reconstruction. In contrast to models in CV that predict masked image patches, or in NLP that predict the next token in a sequence, graph data presents a unique challenge due to its interconnected nature. Consequently, graph generation methods typically work on the feature or structural space. Feature generation methods focus on masking the features of one or a subset of nodes and then training the model to recover the masked features. For instance, GraphMAE [73] utilizes a masked autoencoder framework to reconstruct masked graph portions based on their context, effectively capturing the underlying node semantics and their connection patterns. Alternatively, structure generation methods concentrate on training the model to recover the graph structure. The method GraphGPT [79] encodes the graph into sequences of tokens and then employs a transformer decoder to predict the next token of the sequence to recover the connectivity of the graph. In addition, Graph-Bert [57] is trained on both node attribute recovery and graph structure recovery tasks to ensure that the model captures local node attribute information while maintaining a global view of the graph structure.

Graph Property Prediction methods gain guidance from the node-, edge-, and graph-level properties, which are inherently present in the graph data. These methods follow a training approach similar to supervised learning, as both utilize ”sample-label” pairs for training. The key distinction lies in the origin of the labels: in supervised learning, labels are manually annotated by human experts which can be costly in real scenarios, whereas in property-based learning, the labels are automatically generated from the graph using some heuristics or algorithms. For example, GROVER [77] utilizes professional software to extract the information on graph motifs as labels for classification. Similarly, [111] leverage statistical properties of the graph for graph self-supervised learning.

3.2.2 Downstream Adaptation

Unsupervised Representation Learning (URL) is a common method due to the scarcity of labeled data in the real world [71]–[74]. In URL, the pre-trained graph encoder is frozen and only a task-specific layer is learned during downstream tuning. The learned representations are then directly fed into decoders. This pattern allows URLs to be efficiently applied to downstream tasks. For example, DGI [71] trains an encoder model to learn node representations within graph-structured. Node representations can then be used for downstream tasks. However, due to the gap between the pretext task and downstream tasks, URL can also lead to suboptimal performance.

Fine-tuning is the default method to adapt a pre-trained model to a certain downstream task. As shown in Figure 3, it adds a randomly initialized task header (e.g., a classifier) on top of the pre-trained model, and during fine-tuning, both the backbone model and the header are jointly trained [10], [11], [57]. Compared with URL, fine-tuning provides more flexibility as it allows changes in the backbone parameters, and one can choose the layers to be tuned while keeping others fixed. Additionally, recent studies [10], [76], [78] further explore advanced graph fine-tuning methods that go beyond naive fine-tuning. For instance, AdapterGNN [76] introduces two trainable adapters in parallel before and after the message passing. It freezes the GNN model during fine-tuning while only tuning the adapters, enabling parameter-efficient fine-tuning with minimal influence on the downstream performance.

Prompt-tuning: ”Pre-training & fine-tuning” is prevalent in adapting pre-trained models to specific downstream tasks, but it overlooks the gap between pre-training and downstream tasks, potentially limiting generalization capabilities. Moreover, fine-tuning for different tasks also leads to significant time and computational costs. Inspired by recent advancements in NLP, several methods [82]–[88], [90], [91] have presented the potential of introducing prompts to adapt pre-trained models to specific tasks as illustrated in Figure 3. Specifically, Prompt-tuning first unifies the downstream task with the pre-trained task into the same paradigm, followed by the introduction of learnable prompts for tuning. For example, GPPT [83] first reframe node classification as link predictions. GraphPrompt [82] further extends graph classification into link prediction. On the other hand, Prog [86] unifies all the downstream tasks into subgraph classification. The inserting prompt including vectors [82], [83], [85], node [90] and sub-graph [86]. By inserting these prompts, the pre-trained parameters can be utilized in a way that aligns more closely with the requirements of the downstream tasks.

4 LLMS FOR GRAPH MODELS

Despite great potential, Graph ML based on the GNNs has its inherent limitations. Firstly, vanilla GNN models commonly demand labeled data for supervision, and obtaining these annotations can be resource-intensive in terms of time and cost. Secondly, real-world graphs often contain abundant textual information, which is crucial for downstream tasks. However, GNNs typically rely on shallow text embeddings for semantic extraction, thereby limiting their capacity to capture intricate semantics and text features. Moreover, the diversity of graphs presents challenges for GNN models in terms of generalization across diverse domains and tasks.

Recently, LLMs have achieved remarkable success in handling natural language, with exciting features like (1) conducting zero/few-shot predictions and (2) providing a unified feature space. These capabilities present a potential solution to address the above challenges faced by Graph ML and GFMs. Therefore, this section aims to investigate the contributions that current LLMs can make to enhance Graph ML's progress towards GFMs, while also examining their current limitations, as Figure 4 shows.

4.1 Enhancing Feature Quality

Graphs encompass diverse attribute information, spanning text, images, audio, and other multi-modal modes. The semantics of these attributes play a crucial role in a range of downstream tasks. In comparison with earlier pre-trained models, LLMs stand out due to their substantial parameter volume and training on extensive datasets, endowing them with rich open-world knowledge. Consequently, researchers are exploring the potential of LLMs to improve feature quality and align feature space. This section delves into research endeavors aimed at leveraging LLMs to accomplish these goals.

4.1.1 Enhancing Feature Representation

Researchers utilize the powerful language understanding capabilities of LLMs to generate better representations for text attributes compared to traditional shallow text embeddings [26], [112], [113]. For example, Chen et al. [26] utilize LLMs as text encoders and the GNN model as a predictor, validating the effectiveness of LLMs as an enhancer in node classification tasks. In LKPNR [112], an LK-Aug news encoder enhances the news recommender system by concatenating LLM embeddings with entity embeddings within the news text to obtain an enriched news representation. Several researchers explore fine-tuning LLMs to obtain text representations better suited for downstream graph tasks. SimTeG [25] treats node classification and link prediction tasks as text classification and text similarity tasks, fine-tuning PLMs using LoRA [146] on the TAG dataset. The fine-tuned PLMs are then used to generate embeddings for text attributes, followed by GNN training for downstream tasks.

4.1.2 Generating Augmented Information

Several studies investigate leveraging the generation capabilities and general knowledge of LLMs to generate augmented information from original textual attributes. TAPE [114] first leverages LLM to generate potential node labels and explanations, utilizing text attributes (such as title and abstract) as input. These labels and explanations generated by the LLM are regarded as augmented attributes. Subsequently, these augmented attributes are encoded by a fine-tuned language model (LM) and processed by a GNN model, which integrates the graph structure for making final predictions.

Heterophily means the connected nodes tend to have different class labels and dissimilar features. Opp to this, homophily means the connected nodes tend to have the same class labels and similar features.

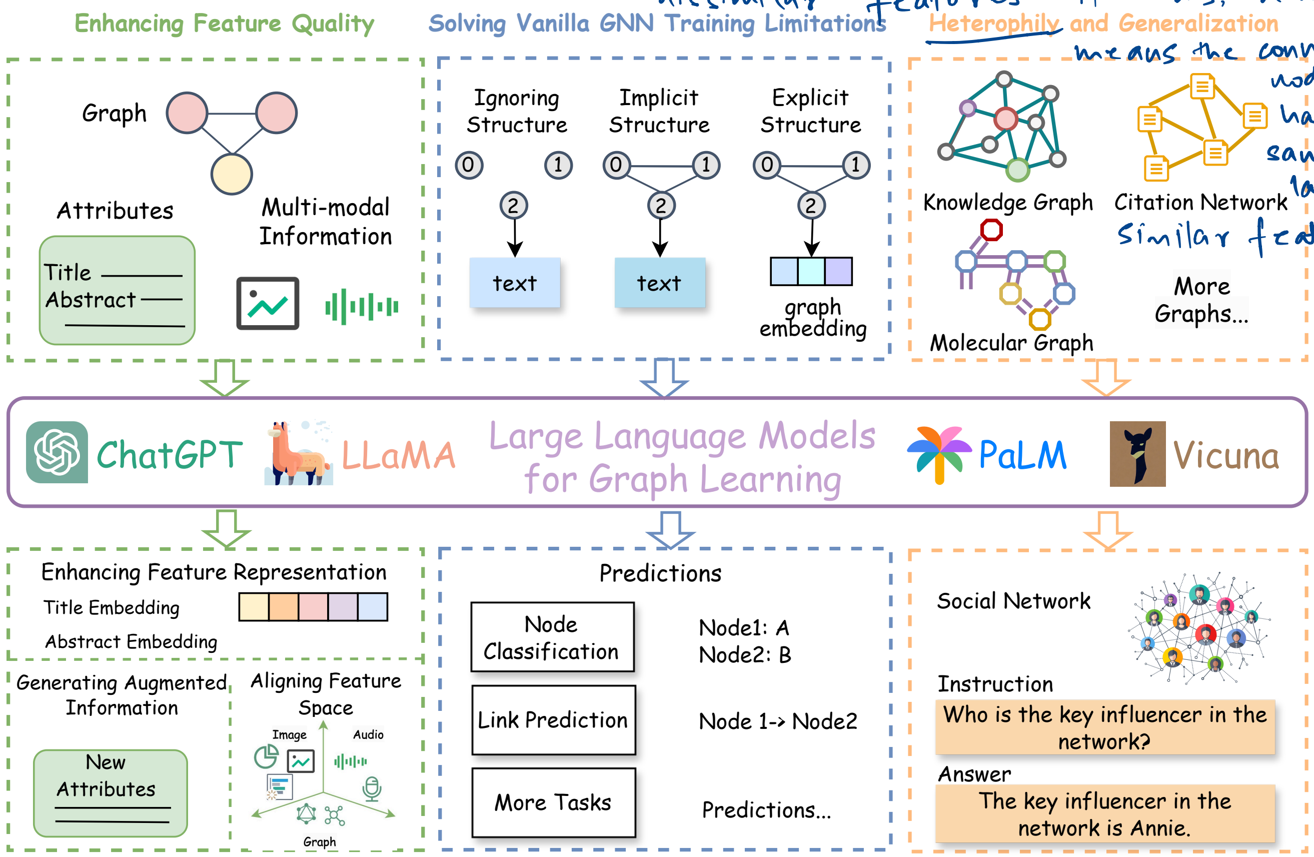

Figure 4: Illustration of LLMs for Graph ML. (1) Methods using LLMs for *Enhancing Feature Quality* by enhancing feature representation, generating augmented information, and aligning feature space. (2) Explorations for *solving Vanilla GNN Training Limitations* are categorized based on how structural information in the graph is processed: ignoring structural information, implicit structural information, and explicit structural information. (3) Research about employing LLMs to alleviate the limitations of *Heterophily and Generalization*.

In contrast to TAPE, KEA [26] does not directly predict node labels with LLM. Instead, LLM extracts terms mentioned in textual attributes and provides detailed descriptions of these terms.

In the domain of molecular property prediction, both LLM4Mol [63] and GPT-MolBERTa [118] adopt a similar approach, where LLMs generate interpretations for input Simplified Molecular-Input Line-Entry System (SMILES) notations as augmented attributes.

In the realm of recommender systems, several methods leverage LLMs to enhance the textual attributes of both users and items. LLM-Rec [117] enables LLMs to produce more detailed item descriptions by explicitly stating the recommendation intent within the prompt. RLMRec [115] explores using LLM to enhance user preference. Specifically, the LLM receives user and item information as input, generates user preferences, potential types of users that the item may attract, and the reasoning process. LLMRec [116] employs a similar approach to enhance item and user attributes in recommender systems. For instance, based on historical behavior information, LLM outputs user profiles like age, gender, country, language, and preferred or disliked genres. For item attributes, taking movie information such as title as input, LLM generates outputs such as movie director, country, and language.

In addition to generating augmented text attributes, researchers also employ LLMs to enhance graph topological structures by generating or refining nodes and edges. In ENG [119], LLM is employed to generate new nodes and their corresponding text attributes for each node category. To integrate the generated nodes into the original graph, the authors train an edge predictor using relations in the original dataset as supervised signals. Sun et al. [120] leverage LLMs to refine graph structures. Specifically, they let LLMs remove unreliable edges by predicting the semantic similarity between node attributes. Additionally, they utilize pseudo-labels generated by LLMs to aid the GNN in learning proper edge weights.

4.1.3 Aligning Feature Space

In real-world scenarios, the text attributes of graphs across different domains exhibit considerable diversity. Additionally, beyond text modal attributes, the graph may encompass various other modal attributes. Employing Pretrained Models (PMs) directly for encoding cross-domain and multi-modal features may not produce satisfactory results. Therefore, LLMs are employed to align feature space and provide better representations. TouchUp-G [81] introduces a graph-centric fine-tuning strategy aimed at enhancing multi-modal features for graph-related tasks.

Table 2: A summary of LLM for Graph ML research. We present the GNN model, LLM model, predictor, domain, task, datasets, and project link. **FT** is Fine-tuning, refers to whether modifications are made to the parameters of the LLM model while **PR** is Prompting, involves inputting textual prompts to the LLM to obtain responses. In the context of **task**, ”node” denotes node-level tasks such as node classification, ”edge” signifies edge-level tasks like link prediction, ”graph” represents graph-level tasks such as graph classification, and ”structure” pertains to structure understanding tasks, such as node degree counting.

Role	Sub Category	Method	GNN Model	LLM Model	FT	PR	Domain	Task	Datasets	Link
Enhancing Feature Quality	Enhancing Feature Representation	Chen et al. [26]	GCN, GAT, MLP	ChatGPT, LLaMA	✗	✗	Citation, E-commerce	Node	5	link
		SimTeG [25]	SAGE, MLP, etc.	all- MiniLM-L6-v2, etc.	✓	✗	Citation, E-commerce	Node, Edge	3	link
		LKPNR [112]	wiki KG	ChatGLM2, RWKV, LLaMA2	✓	✗	Recommendation	Click Prediction	1	link
		GRID [113]	GAT	INSTRUCTOR	✗	✗	Robotics	Robot Task Planning	2	link
	Generating Augmented Information	TAPE [114]	GCN, SAGE,MLP, RevGAT	ChatGPT, LLaMA2	✗	✓	Citation, E-commerce	Node	5	link
		KEA [26]	GCN, GAT, MLP	ChatGPT	✗	✓	Citation, E-commerce	Node	5	link
		RLMRec [115]	LightGCN	ChatGPT	✗	✓	Recommendation	Recommendatoon	3	link
		LLMRec [116]	LightGCN	ChatGPT	✗	✓	Recommendation	Recommendation	2	link
		LLM-Rec [117]	-	text-davinci-003	✓	✗	Recommendation	Recommendation	2	-
		LLM4Mo [63]	-	ChatGPT	✗	✓	Molecular	Molecular Property Prediction	3	link
		GPT-MolBERTa [118]	-	ChatGPT	✗	✓	Molecular	Molecular Property Prediction	9	-
		ENG [119]	GCN,GAT	ChatGPT	✗	✓	Citation	Node	-	
		Sun et al. [120]	GAT, GCN, etc.	ChatGPT	✗	✓	Citation	Node	4	-
	Aligning Feature Space	TouchUp-G [81]	SAGE, GAT, etc.	BERT, etc.	✓	✗	Citation, E-commerce	Node	4	link
		OFA [121]	GCN,GAT,etc.	LLaMA-2-7B, etc.	✓	✗	Citation, Molecular, etc.	Node, Edge, Graph	9	link
Solving Vanilla GNN Training Limitations	Ignoring Structural Information	Hu et al. [122]	-	ChatGPT,GPT-4	✗	✓	Citation,KG	Node, Edge	5	-
	Implicit Structural Information	GPT4Graph [21]	-	text-davinci-003	✗	✓	-	Structure,Graph,Node,KGQA	4	link
		GraphText [28]	-	ChatGPT, LLaMA2-7B	✓	✓	Citation, Web	Node	7	-
		NLGraph [123]	-	ChatGPT, GPT-4, etc.	✗	✓	-	Structure	3	link
		InstructGLM [20]	-	Flan T5, LLaMA	✓	✓	Citation	Node	3	link
		LLMtoGraph [124]	-	ChatGPT, GPT-4, Vicuna-13B, etc.	✗	✓	-	Structure	-	link
		Graph Agent [125]	-	GPT-4, embedding-ada-002	✗	✓	Citation, Bioinformatics	Node, Edge	2	-
		LLM-Prop [126]	-	T5	✓	✗	materials science	Crystal Property Prediction	-	link
		GLRec [127]	-	BELLE-LLaMA-7B	✓	✓	Recommendation	Recommendation	1	-
		ReLM [128]	TAG,GCN	GPT-3.5, Vicuna	✗	✓	Chemistry	Chemical Reaction Prediction	5	link
		Chen et al. [26]	-	ChatGPT	✗	✓	Citation, E-commerce	Node	5	link
		Hu et al. [122]		ChatGPT,GPT-4	✗	✓	Citation,KG	Node, Edge	5	-
		Huang et al. [129]	-	ChatGPT	✗	✓	Citation, E-commerce	Node	5	link
		Fatemi et al. [130]		PaLM2 XXS, PaLM 62B	✗	✓	-	Structure	-	-
		MolReGPT [131]	-	ChatGPT	✗	✓	Molecular	molecule-caption translation	1	link
		LLaGA [132]	-	Vicuna-7B	✗	✓	Citation, E-commerce	Node, Edge	4	link
		GraphEdit [133]	GCN	Vicuna-v1.5	✓	✓	Citation	Node	3	link
	Explicit Structure Information	GraphGPT [22]	Graph Transformer	Vicuna-7B	✓	✓	Citation	Node	3	link
		GraphLLM [134]	Graph Transformer	LLaMA2	✗	✓	-	Structure	4	link
		GNP [135]	GAT	Flan T5	✓	✓	KG	KGQA	4	-
		DrugChat [136]	GAT, etc.	Vicuna-13B	✗	✓	Drug	QA	-	link
		KoPA [137]	RotateE	Alpaca-7B	✓	✓	Knowledge Graph	Knowledge Graph Completion	3	link
		GIMLET [138]	-	T5	✓	✓	Molecular	Molecular Property Prediction	14	link
		GIT-Mol [139]	GIN	MolT5	✓	✓	Molecular	Molecular Generation, etc.	6	-
		BioMedGPT [140]	GIN	LLaMA2-7B-Chat	✓	✓	Biomedical	QA	3	link
		ProteinChat [141]	GVP-GNN	Vicuna-13B	✗	✓	Protein	QA	1	link
		DGTL [142]	Disentangled GNN	LLaMA-2-13B-chat	✗	✓	Citation, E-commerce	Node	3	-
		G-Retriever [143]	GAT	LLaMA-2-7B	✓	✓	-	QA	3	link
		GraphToken [144]	GCN,GIN,etc.	PaLM 2 S	✗	✓	-	Structure	1	-
HG	Heterophily	Chen et al. [26]	-	ChatGPT	✗	✓	Citation, E-commerce	Node	5	link
		GraphText [28]	-	ChatGPT, LLaMA2-7B	✓	✓	Citation, Web	Node	7	-
	Generalization	OpenGraph [145]	Graph Transformer	Not mentioned	✗	✓	Citation, Drug, etc.	Node, Edge	7	link

Initially, they present a novel feature homophily measure to quantify the alignment between node features and the graph structure. Building upon this measure, the authors devise a structure-aware loss function to optimize the PM by minimizing discrepancies between features and graphs. The work of [121] introduces OFA, a unified framework for classification tasks in graphs across different domains. OFA collects nine text-attributed graph datasets covering diverse domains and represents nodes and relations in natural language. LLMs are then employed to embed those cross-domain graph information into the same embedding space. Moreover, OFA proposes a graph prompting paradigm, which incorporates a prompt graph containing downstream task information into the original input graph, allowing the GNN model to adaptively perform different tasks based on the prompt graph.

4.2 Solving Vanilla GNN Training Limitations

The training of vanilla GNNs relies on labeled data. However, obtaining high-quality labeled data has long been associated with substantial time and costs. In contrast to GNNs, LLMs showcase robust zero/few-shot capabilities and possess expansive open-world knowledge. This unique characteristic empowers LLMs to directly leverage node information for prediction, without relying on extensive annotated data. Therefore, researchers have explored employing LLMs to generate annotations or predictions, alleviating dependence on human supervision signals in Graph ML. According to how structural information in graph data is processed, we categorize the methods into the following three categories:

- Ignoring structural information: utilize node attributes exclusively for constructing textual prompts, disregarding neighboring labels and relations.
- Implicit Structural information: describe neighbor information and graph topology structure in natural language;
- Explicit Structural information: employ GNN models to encode graph structure.

4.2.1 Ignoring Structural Information

The fundamental distinction between graphs and text lies in the structural information inherent in graphs. Given that the LLM processes text as its input, an intuitive approach involves leveraging the textual attributes of the target node, disregarding the structural information within the graph, and making predictions directly. For instance, the work of [122] explores the effectiveness of LLMs in solving graph tasks without using structure information. In the citation network, they employ the article's title and abstract to construct a prompt and instruct the LLM to predict the article's category. Since this kind of paradigm does not incorporate the structural information of the graph, the actual task performed by the LLM is text classification rather than a graph-related task.

4.2.2 Implicit Structural Information

Researchers implicitly leverage structural information to solve graph tasks by describing graph structure in natural language. For example, Hu et al. [122] propose two kinds of methods for utilizing structural information. The first method involves directly inputting the data of all neighboring nodes into LLM, while the second method employs a retrieval-based prompt to guide the LLM to focus solely on relevant neighbor data. Similarly, Huang et al. [129] employ an LLM to assign scores to neighboring nodes and subsequently choose high-scoring nodes as structural information. NLGraph [123] introduces a Build-a-Graph prompting strategy to improve the LLM's understanding of graph structure. This strategy entails appending "Let's construct a graph with the nodes and edges first." after providing the graph data description. The work of [20] introduces InstructGLM, which utilizes natural language for graph description and fine-tunes Flan-T5 through instruction tuning. They generate a set of 31 prompts by combining four configuration parameters: task type, inclusion of node features, maximum hop order, and utilization of node connections. Notably, maximum hop order and node connections implicitly convey graph structure information to the LLM. GraphEdit [133] leverages LLMs to understand graph structure and refine it by removing noisy edges and uncovering implicit node connections. Specifically, it employs an edge predictor to identify the top k candidate edges for each node, and these candidate edges, along with the original edges of the graph, are then fed into the LLM. The LLM is prompted to determine which edges should be integrated into the final graph structure.

In addition to employing natural language expression, several researchers leverage structured languages for graph description. GPT4Graph [21], for instance, utilizes Graph Modelling Language [147] and Graph Markup Language [148] to represent graph structure in XML format. GraphText [28] constructs a graph syntax tree for each graph, containing node attributes and relations information. By traversing this tree, structural graph-text sequences can be generated. The advantage of GraphText lies in the ability to integrate the typical inductive bias of GNNs through the construction of various graph syntax trees.

4.2.3 Explicit Structural Information

While implicitly describing structure in natural language has achieved preliminary success, these methods still face certain limitations. Firstly, due to the constraint of input length, LLMs can only get local structural information, and lengthy contexts might diminish their reasoning [149] and instruction-following abilities [26]. Secondly, for different tasks and datasets, substantial effort is often required for prompt engineering. A prompt that performs well on one dataset may not generalize effectively to others, resulting in a lack of robustness. Consequently, researchers investigate representing graph structure explicitly, typically comprising three essential modules: *encoding module*, *fusion module*, and *LLM module*. More specifically, the encoding module aims to process the graph-structured and textual information, generating graph embeddings and text embeddings, respectively. Afterward, the fusion module takes these two embeddings as input, producing a modality fusion embedding. At last, the modality fusion embedding, which contains both graph information and instruction information, is fed into the LLM to obtain the final answer. Given the research focus is on how LLMs explicitly utilize graph structure information, we will delve into the encoding and fusion modules of various studies in detail, without primarily focusing on the LLM model itself.

Encoding Module. The encoding module is responsible for both graph and text encoding, and we will provide separate summaries for each.

- *Graph Encoding.* Pre-trained GNN models are commonly used for graph encoding. For instance, GIT-Mol [139] employs the GIN model from the pre-trained MoMu model [80] to encode molecular graphs. KoPA [137] utilizes the pre-trained RotateE model to obtain embeddings for entities and relations in the knowledge graph. In addition, GIMLET [138] presents a unified graph-text model without the need for additional graph encoding modules. Particularly, GIMLET proposes a distance-based joint position embedding method, where the shortest graph distance is utilized to represent the relative positions between graph nodes, enabling the Transformer encoder to encode both graph and text. GraphToken [144] evaluates a series of GNN models as graph encoders, including GCN, MPNN [105], GIN, Graph Transformer, HGT [56], etc.
- *Text Encoding.* Due to the tremendous capability of LLMs in understanding textual information, most existing methods, such as ProteinChat [141] and DrugChat [136], directly employ LLMs as text encoders. In GraphLLM [134], the tokenizer and frozen embedding table of LLM are leveraged to obtain the representation of node text attributes, aligning with the downstream frozen LLM.

Fusion Module. The goal of the fusion module is to align the graph and text modalities, generating a fusion embedding as input for the LLM. To achieve the goal, a straightforward solution is to design a linear projection layer to directly transform the graph representation generated by GNN into an LLM-compatible soft prompt vector [136], [137], [140]. Additionally, inspired by BLIP2's Q-Former [150], [139] propose a GIT-Former, which aligns graph, image, and text with the target text modality using self-attention and cross-attention mechanisms.

In addition to the above methods, G-Retriever is proposed to integrate both explicit and implicit structural information [143]. To be specific, GAT is employed to encode the graph

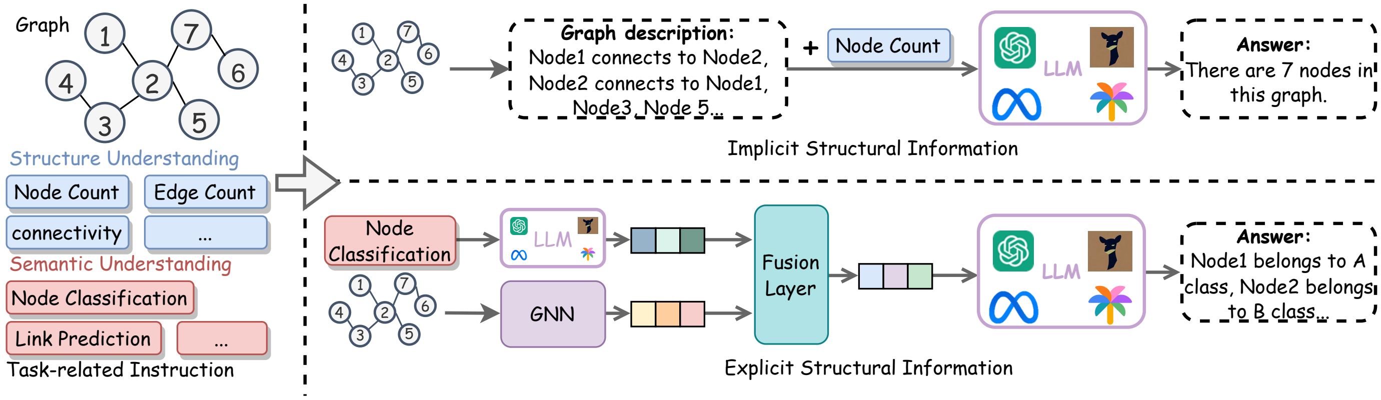

Figure 5: The illustration of employing LLMs with implicit and explicit structural information. (1) Methods leveraging *implicit structural information* describe nodes and graph structure information in natural language and combine task-specific instructions to form a textual prompt, which is then input into the LLM to generate prediction results. (2) Methods employing *explicit structural information* use GNNs and LLMs to encode graph and instruction information separately. Then, fusion layers are added to align the graph and text modalities, and the fused embedding is input into the LLM for prediction.

structure, while representing node and relationship details through textual prompts. To accommodate real-world graphs with larger scales, G-Retriever introduces a RAG module specifically designed for retrieving subgraphs relevant to user queries.

4.3 Heterophily and Generalization

Despite achieving promising performance in graph tasks, GNNs exhibit several shortcomings. A notable drawback involves the inadequacy of the neighbor information aggregation mechanism, especially when dealing with heterogeneous graphs. GNN performance notably diminishes when faced with instances where adjacent nodes lack similarity. Additionally, GNN encounters challenges in out-of-distribution (OOD) generalization, leading to a degradation in model performance on distributions beyond the training data. This challenge is particularly prevalent in practical applications, primarily due to the inherent difficulty of encompassing all possible graph structures within limited training data. Consequently, when GNNs infer on unseen graph structures, their performance may experience a substantial decline. This reduced generalization capability renders GNNs relatively fragile when confronted with evolving graph data in real-world scenarios. For example, GNNs may encounter difficulties handling newly emergent social relationships in social networks.

LLMs have been utilized to mitigate the above limitations. In particular, GraphText [28] effectively decouples depth and scope by encapsulating node attributes and relationships in the graph syntax tree. This approach yields superior results compared to the GNN baseline, particularly on heterogeneous graphs. Chen et al. [26] investigate the LLM's ability to handle OOD generalization scenarios. They utilize the GOOD [151] benchmark as the criterion and results demonstrate that LLMs exhibit promising performances in addressing OOD generalization issues. OpenGraph [145] aims at solving zero-shot graph tasks across different domains. In this model, LLMs are leveraged to generate synthetic graphs of data scarcity scenarios, thereby enhancing the pre-training process of OpenGraph.

5 Graphs for LLMs

LLMs have demonstrated impressive language generation and understanding capabilities across various domains. Nevertheless, they still face several pressing challenges, including factuality awareness, hallucinations, limited explainability in the reasoning process, and beyond. To alleviate these issues, one potential approach is to take advantage of the *Knowledge Graphs* (KGs), which store high-quality, human-curated factual knowledge in a structured format [5]. Recent reviews [152]–[154] have summarized the research on using KGs to enhance LMs. Hu et al. [152] present a review on knowledge-enhanced pre-training language models for natural language understanding and natural language generation. Agrawal et al. [153] systematically review research on mitigating hallucination in LLMs by leveraging KGs across three dimensions: inference process, learning algorithm, and answer validation. Pan et al. [154] provides a comprehensive summary of the integration of KGs and LLMs from three distinct perspectives: KG-enhanced LLMs, LLM-augmented KGs, and the synergized LLMs and KGs, where LLMs and KGs mutually reinforce each other. In this section, we will delve into relevant research that explores the usage of KGs to achieve knowledge-enhanced language model pre-training, mitigate hallucinations, and improve inference explainability.

5.1 KG-enhanced LLM Pre-training

While LLMs excel in text understanding and generation, they may still produce grammatically accurate yet factually incorrect information. Explicitly incorporating knowledge from KGs during LLM pre-training holds promise for augmenting LLM's learning capacity and factual awareness [155]–[157]. In this subsection, we will outline the research advancements in KG-enhanced pre-trained language models (PLMs). While there is limited work on KG-enhanced pre-training for LLMs, research on KG-enhanced PLMs can offer insights for LLM pretraining. Existing KG-enhanced pre-training methods can be classified into three main categories: modifying input data, modifying model structures, and modifying pre-training tasks.

5.1.1 Modifying Input Data

Several researchers investigate integrating KG knowledge by modifying input data while keeping the model architecture unchanged. For instance, Moiseev et al. [158] directly train PLMs on mixed corpora consisting of factual triples from KGs and natural language texts. E-BERT [159] aligns entity vectors with BERT's wordpiece vector space, preserving the structure and refraining from additional pre-training tasks. KALM [160] utilizes an entity-name dictionary to identify entities within sentences and employs an entity tokenizer to tokenize them. The input of the Transformer consists of the original word embeddings and entity embeddings. Moreover, K-BERT [161] integrates the original sentence with relevant triples by constructing a sentence tree, where the trunk represents the original sentence and the branches represent the triples. To convert the sentence tree into model input, K-BERT introduces both a hard-position index and a soft-position index within the embedding layer to differentiate between original tokens and triple tokens.

5.1.2 Modifying Model Structures

Some research designs knowledge-specific encoders or fusion modules to better inject knowledge into PLMs. ERNIE [162] introduces a K-Encoder to inject knowledge into representations. This involves feeding token embeddings and the concatenation of token embeddings and entity embeddings into a fusion layer for generating new token embeddings and entity embeddings. In contrast, CokeBERT [163] extends this approach by incorporating relation information from KGs during pre-training. It introduces a semantic-driven GNN model to assign relevant scores to relations and entities based on the given text. Finally, it fuses the selected relations and entities with text using a K-Encoder similar to ERNIE. KLMO [164] propose Knowledge Aggregator to fuse text modality and KG modality during pre-training. To incorporate the structural information in KG embeddings, KLMO utilizes KG attention, which integrates a visibility matrix with a conventional attention mechanism, facilitating interaction among adjacent entities and relations within the KG. Subsequently, the token embeddings and contextual KG embeddings are aggregated with entity-level cross-KG attention.

Several studies refrain from modifying the overall structures of the language model but introduce additional adapters to inject knowledge. To preserve the original knowledge within PLMs, Wang et al. [165] propose K-Adapter as a pluggable module to leverage KG knowledge. During pre-training, the parameters of the K-Adapter are updated while the parameters of the PLMs remain frozen. KALA [166] introduces a Knowledge-conditioned Feature Modulation layer, which functions similarly to an adapter module, by scaling and shifting the intermediate hidden representations of PLMs with retrieved knowledge representations. To further control the activation levels of adapters, DAKI [167] incorporates an attention-based knowledge controller module, which is an adapter module with additional linear layers.

5.1.3 Modifying Pre-training Tasks

To explicitly model the interactions between text and KG knowledge, various pre-training tasks are proposed. Three major lines of work in this direction include entity-centric tasks [162], [168]–[171], relation-centric tasks [155], and beyond.

For entity-centric tasks, ERNIE [162] randomly masks some token-entity alignments and then requires the model to predict all corresponding entities based on aligned tokens. LUKE [168] uses Wikipedia articles as training corpora and treats hyperlinks within them as entity annotations, training the model to predict randomly masked entities. KILM [169] also utilizes hyperlinks in Wikipedia articles as entities. However, it inserts entity descriptions after corresponding entities, tasking the model with reconstructing the masked description tokens rather than directly masking entities. In addition to predicting masked entities, GLM [170] further introduces a distractor-suppressed ranking task. This task leverages negative entity samples from KGs as distractors, enhancing the model's ability to distinguish various entities.

Relation-centric tasks are also commonly utilized in KG-enhanced PLMs. For instance, JAKET [172] proposes relation prediction and entity category prediction tasks for enhancing knowledge modeling. Dragon [173] is pre-trained in a KG link prediction task. Given a text-KG pair, the model needs to predict the masked relations in KG and the masked tokens in the sentence. ERICA [174] introduces a relation discrimination task aiming at semantically distinguishing the proximity between two relations. Specifically, it adopts a contrastive learning manner, wherein the relation representations of entity pairs belonging to the same relations are encouraged to be closer.

Additionally, there are several innovative pre-training tasks for KG-enhanced pre-training. KEPLER [175] proposes a knowledge embedding task to enhance knowledge-awareness of PLMs. Specifically, it uses PLMs to encode entity descriptions as entity embeddings and jointly train the knowledge embedding and masked language modeling tasks on the same PLM. ERNIE 2.0 [176] constructs a series of continuous pre-training tasks from word, structure, and semantic perspectives.

5.2 KG-enhanced LLM Inference

Knowledge within KGs can be dynamically updated, whereas updating the knowledge in LLMs often necessitates adjustment of model parameters, which demands substantial computational resources and time. Therefore, many studies opt to utilize KGs during LLMs inference stage. The "black-box" nature of LLMs poses a significant challenge in understanding how the model made a specific prediction or generated a specific text. Additionally, LLMs have often been criticized for generating false, erroneous, or misleading content, typically referred to as hallucination [29], [30], [177]. Given the structured and fact-based nature of KGs, integrating them during the inference stage can enhance the explainability of LLM answers and consequently mitigate hallucinations.

While several methods extract relevant triples from KGs based on user queries and describe these triples in natural language within prompts [178], [179], these approaches overlook the structured information inherent in KGs, and still fail to elucidate how LLMs arrive at their answers. Consequently, extensive studies utilize KGs to aid LLMs in reasoning and generate intermediary information like relation paths, evidence subgraphs, and rationales, forming the basis for explaining the LLM's decision-making process and checking for hallucinations [32], [34], [35], [180]–[182].

Several researchers investigate enabling LLMs to directly reason on KGs and generate relation paths to interpret LLM's answers. The reasoning path at each step helps to enhance the explainability of the answer and the transparency of the reasoning process. Through observing the reasoning decisions made at each step, it becomes possible to identify and address hallucinations arising from LLMs' reasoning. Both RoG [32], Knowledge Solver [181], and Keqing [33] employ relation paths as explanations for LLM's responses. Specifically, given the KG schema and user query, RoG [32] guides LLMs to predict multiple relation paths using textual prompts like "Please generate helpful relation paths for answering the question". Subsequently, LLMs generate the final answer based on the retrieving results of the valid relation path. Conversely, the Knowledge Solver method [181] differs in that it enables LLMs to generate the relation path step by step. Keqing [33] initially decomposes complex questions into several sub-questions, each of which can be addressed by pre-defined logical chains on KGs, and then LLMs will generate final answers with relation paths based on the answers of sub-questions. Mindmap [180] uses evident subgraphs to explain the answers generated by LLMs, where path-based and neighbor-based methods are introduced to obtain several evident subgraphs. The LLM in Mindmap is prompted to merge these evident subgraphs, utilizing the merged graph to generate the final answer. In contrast to previous methods which involve gradually retrieving knowledge and obtaining answers, KGR [34] takes a different approach. Initially, the LLM directly generates a draft answer. Subsequently, it extracts the claims requiring verification from this answer and retrieves KG's information to correct claims with hallucinations. Based on the corrected claims, the LLM adjusts the draft answer to get the final answer.

The above research employs relation paths or evident graphs as the basis for explaining the LLM's decision-making process and checking hallucinations. In contrast, several research explore using inherently interpretable models rather than LLMs to make final predictions. ChatGraph [183] presents an innovative approach to enhance both the text classification capabilities and explainability of ChatGPT. It utilizes ChatGPT to extract triples from unstructured text and subsequently constructs KGs based on these triples. To ensure the explainability of the classified results, ChatGraph avoids employing LLMs directly for predictions. Instead, it leverages a graph model without non-linear activation functions and trains the model on text graphs to get predictions. Given a question and a list of possible answers, XplainLLM [184] propose an explainer model to explain why LLMs choose a particular answer while rejecting others. Specifically, the approach involves constructing an element graph based on the entities present in the question and the candidate answers. Subsequently, a GCN model is employed to assign attention scores to each node within the element graph. Nodes exhibiting high attention scores are identified as reason elements, and LLMs are then prompted to provide explanations based on these selected reason elements.

To assess the transparency and interpretability of LLMs, various benchmarks have been proposed. For example, Li et al. [35] introduce a novel task named Knowledge-aware Language Model Attribution (KaLMA) and develop a corresponding benchmark dataset. This benchmark evaluates the LLM's capability to derive citation information from KG to support its answers. KaLMA also provides an automatic evaluation covering aspects such as text quality, citation quality, and text-citation alignment of the answers. In addition, XplainLLM [184] introduces a dataset for better understanding LLMs' decision-making from the perspectives of "why-choose" and "why-not-choose".

6 Applications

In this section, we will present practical applications that demonstrate the potential and value of GFMs and LLMs. As shown in Table 2, recommender systems, knowledge graphs, AI for science, and robot task planning emerge as the most prevalent domains. We will provide a comprehensive summary of each of these applications.

6.1 Recommender Systems

Recommender systems leverage user historical behaviors to predict items that users are likely to appreciate [185]–[187]. Graphs play a crucial role in recommender systems, wherein items can be regarded as nodes and collaborative behaviors such as clicks and purchases can be viewed as edges. Recently, an increasing amount of research is exploring the use of LLMs for direct recommendation [188]–[191] or leveraging LLMs to enhance graph models or datasets for recommendation tasks [112], [115], [116], [192], [193].

For directly using LLMs as recommendation models, liu et al. [194] construct task-specific prompts to evaluate ChatGPT's performance on five common recommendation tasks, encompassing rating prediction, sequential recommendation, direct recommendation, explanation generation, and review summarization. Bao et al. [195] employ prompt templates to guide LLM to decide whether the user will like the target item based on their historical interactions and perform instruction tuning on the LLM to improve its recommendation capability.

For using LLMs to enhance traditional recommendation methods or datasets, KAR [192] leverages LLMs to generate factual knowledge of items and reasoning basis of user preferences; these knowledge texts are then encoded into vectors and integrated into existing recommendation models. Methods like LLM-Rec [117], RLMRec [115], and LLM-Rec [116] enrich recommendation datasets by incorporating LLM-generated descriptions. In contrast, Wu et al. [193] utilize LLMs to condense recommendation datasets, in which LLMs are employed to synthesize a condensed dataset for the content-based recommendation, aiming at addressing the challenge of resource-intensive training on large datasets.

While the previously discussed methods have explored utilizing LLMs for certain recommendation tasks or domains, an emerging research direction aims to develop foundation models for recommendation. Tang et al. [189] propose an LLM-based domain-agnostic framework for sequential recommendation. Their approach integrates user behavior across domains, and leverages LLMs to model user behaviors based on multi-domain historical interactions and item titles. Hua et al. [196] attempt to address the potential unfairness of recommender systems introduced by LLM bias. They propose a Counterfactually Fair Prompting method to develop an unbiased foundation model for recommendation. To summarize the progress in the area of recommendation foundation model, Huang et al. [197] provide a systematic overview of the existing approaches, categorizing them into three main types: language foundation models, personalized agent foundation models, and multi-modal foundation models.

6.2 Knowledge Graphs

LLMs with robust text generation and language understanding capabilities have found extensive applications in KG-related tasks, including KG completion [137], [198], [199], KG question answering [179], [181], [200]–[202], KG reasoning [203] and beyond. Meyer et al. [204] introduce LLM-KG-Bench, a framework that automatically evaluates the model's proficiency in KG engineering tasks such as fixing errors in Turtle files, facts extraction, and dataset generation. KG-LLM [199] is proposed to evaluate LLMs' performance on KG completion, including triple classification, relation prediction, and link prediction tasks. Kim et al. [200] propose KG-GPT, using LLMs for complex reasoning tasks on knowledge graphs. ChatKBQA [201] introduces a generate-then-retrieve framework for LLMs on knowledge base question answering. Wu et al. [179] present a KG-enhanced LLM framework for KG question answering, which involves fine-tuning an LLM to convert structured triples into free-form text, enhancing LLMs' understanding of KG data. The successful application of LLMs in tasks such as KG construction, completion, and question answering offers robust support for advancing the understanding and exploration of KGs.

Drawing inspiration from foundation models in language and vision, researchers are delving into the development of foundation models tailored for KGs. These GFMs aim to generalize to any unseen relations and entities within KGs. Galkin et al. [205] propose Ultra, which learns universal graph representations by leveraging interactions between relations. This study is based on the insight that those interactions remain similar and transferable across different datasets.

6.3 AI for Science

The rapid advancement of AI has led to an increasing number of studies leveraging AI to assist scientific research [206], [207]. Recent research has applied LLMs and GFMs for scientific purposes, such as drug discovery, molecular property prediction, and material design. Notably, these applications encompass scenarios involving graph-structured data.

The molecular graph is a way of representing molecules, where the nodes represent atoms, and the edges represent the bonds between the atoms. With the emergence of LLMs, researchers have explored their performance in tasks related to molecular graphs. Methods like MolReGPT [131] and GPT-MolBERTa [118] adopt a similar approach, converting molecular graphs into textual descriptions using SMILES language. They create prompts based on SMILES data, asking the LLM to provide detailed information about functional groups, shapes, chemical properties, etc. This information is then used to train a smaller LM for molecular property prediction. In contrast to methods directly using LLMs for prediction, ReLM [128] first uses GNNs to predict high-probability candidate products, and then leverages LLMs to make the final selection from these candidates.

In addition to the above research, LLMs are further utilized in drug discovery and materials design. Bran et al. [100] present ChemCrow, a chemistry agent integrating LLMs and 18 specialized tools for diverse tasks across drug discovery, materials design, and organic synthesis. InstructMol [208] presents a two-stage framework for aligning language and molecule graph modalities in drug discovery. Initially, the framework maintains the LLM and graph encoder parameters fixed, focusing on training the projector to align molecule-graph representations. Subsequently, instruction tuning is conducted on the LLM to address drug discovery tasks. Zhao et al. [209] propose ChemDFM, the first dialogue foundation model for chemistry. Trained on extensive chemistry literature and general data, ChemDFM exhibits proficiency in various chemistry tasks such as molecular recognition, molecular design, and beyond.

6.4 Robot Task Planning

Robot task planning aims to decompose tasks into a series of high-level operations for the step-by-step completion by a robot [210]. During task execution, the robot needs to perceive information about the surrounding environment, typically represented using *scene graphs*. In a scene graph, nodes represent scene objects like people and tables, while edges describe the spatial or functional relationships between objects. Enabling LLMs for robot task planning crucially depends on how to represent the environmental information in the scene graph.

Many studies have explored using textual descriptions of scene information and constructing prompts for LLMs to generate task plans. Chalvatzaki et al. [211] introduce the Graph2NL mapping table, representing attributes with different numerical ranges using corresponding textual expressions. For instance, a distance value greater than 5 is represented as "distant", and smaller than 3 is represented as "reachable". SayPlan [212] describes the scene graph in JSON as a text sequence, iteratively invoking LLM to generate plans and allowing for self-correction. Zhen et al. [213] propose an effective prompt template, Think_Net_Prompt, to enhance LLM performance in task planning. In contrast to methods that rely on language to describe scene graph information, GRID [113] employs the graph transformer to encode scene graphs. It utilizes cross-modal attention to align the graph modality with user instruction, ultimately outputting action tokens through a decoder layer. The powerful understanding and reasoning capabilities of LLMs showcase significant potential in robot task planning. However, as task complexity increases, the search space explosively expands, posing a challenge in efficiently generating viable task plans with LLMs.

7 Future Directions

In this survey, we have thoroughly reviewed the latest developments of Graph Machine Learning in the era of LLMs, revealing significant advancements and potential in this field. By harnessing the power of LLMs, it is potential to enhance Graph ML to enable GFMs. As this research direction is still in the exploratory stage, future directions in this field can be diverse and innovative. Therefore, in this section, we delve into several potential future directions of this promising field.

7.1 Generalization and Transferability

While Graph ML has deployed for various graph tasks, a notable problem is their limited capacity for generalization and transferability across different graph domains [37]. Different from fields such as NLP and CV, where data often adhere to a uniform format (e.g., a sequence of tokens or a grid of pixels), graphs can be highly heterogeneous in nature. This heterogeneity manifests in varying graph sizes, densities, and types of nodes and edges, which presents a significant challenge in developing a universal model capable of performing optimally across various graph structure data. Currently, LLMs have demonstrated great potentials in improving the generalization ability of the graph model. For example, OFA [121] provides a solution for classification tasks across several certain domains. Nevertheless, there is still scarce exploration of the generalizability of GFMs compared to LLMs. Therefore, future research should aim to develop more adaptable and flexible models that can effectively apply learned patterns from one graph type, such as social networks, to another, like molecular structures, without extensive retraining.

7.2 Multi-modal Graph Learning

Recent LLMs have shown significant potential in advancing GFMs. Many efforts have been made to transform graph data into formats suitable for LLM input, such as tokens or text [26], [79], [123]. However, many nodes in graphs are enriched with diverse modalities of information, including text, images, and videos. Understanding this multi-modal data can potentially benefit graph learning. For example, on social media platforms, a user's post could include textual content, images, and videos, all of which are valuable for comprehensive user modeling. Given the importance of multi-modal data, a promising direction for future research is to empower LLMs to process and integrate graph structure with multi-modal data. Currently, TOUCHUP-G [81] makes an initial exploration of multi-modal (i.e., texts, images) in graph learning. In the future, we expect the development of a unified model capable of modeling universal modalities for more advanced GFMs.

7.3 Trustworthiness

The recent applications of LLMs for Graph ML have significantly enhanced graph modeling capabilities and broadened their utility in various fields. Despite these advancements, with the growing reliance on these models, it is important to ensure their trustworthiness, particularly in critical areas like healthcare, finance, and social network analysis [214], [215]. Robustness is fundamental in safeguarding the models against adversarial attacks, ensuring consistent reliability. Explainability is essential for users to understand and trust the decisions made by these models. Fairness is crucial for the model's ethical and effective use in various applications. Privacy is important for legal compliance and key to maintaining user trust. Therefore, the development of trustworthy LLMs on graphs must be equipped with *Robustness&Safety*, *Explainability*, *Fairness*, and *Privacy*, ensuring their safe and effective use in various applications.

7.3.1 *Robustness&Safety*

Recently, integrating LLMs into Graph ML has shown promising performance in various downstream tasks, but they are also highly vulnerable to adversarial perturbations, raising significant concerns about their robustness and safety. To enhance the resilience of these models, some studies add adversarial perturbations to GNNs [216], [217] or LLMs [218], [219] for adversarial training. However, these methods may not be effective for the new paradigm of Graph ML integrating LLMs, as vulnerabilities can arise from both graphs, such as graph poisoning attacks [220], [221] and graph modification attacks [222], [223], and from the language model, like prompt attacks [224] and misleading text data [225]. To address these issues, more sophisticated detection and defense mechanisms need to be developed by considering both the intricacies of LLMs and graphs to ensure the comprehensive safety and robustness of Graph ML.

7.3.2 *Explainability*

Nowadays, LLMs are increasingly employed in Graph ML across various applications, such as recommender systems [15], [226] and molecular discovery [80], [131]. However, due to privacy and security concerns, an application provider may prefer to provide an API version without revealing the architecture and parameters of the LLM, such as with ChatGPT. This lack of transparency can make it challenging for users to understand the model's results, leading to confusion and dissatisfaction. Therefore, it's important to enhance the explainability of Graph ML, especially with LLMs. Owing to their reasoning and interpretive capabilities, LLMs are promising to provide better explainability in graph-related tasks. For example, P5 [226] can provide reasons for its recommendations in recommendation tasks. Future efforts should be directed toward making the inner workings of these models more transparent and explainable to better comprehend their decision-making processes.

7.3.3 *Fairness*

As LLMs become prevalent in enhancing Graph ML towards GFMs, concerns about their fairness are growing. Fairness is crucial to ensure these models operate without biases or discrimination, especially when dealing with complex, interconnected graph data [215]. Recent studies demonstrate that both language models [227], [228] and GNN models [229] can potentially be discriminatory and unfair [39]. Therefore, it is necessary to maintain fairness in both textual and graph contexts. To enhance the fairness of

LLMs, recent studies include retraining strategies that adjust model parameters for unbiased outputs [230], implementing alignment constraints [231], and adopting contrastive learning to diminish bias in model training [232]. Concurrently, studies like FairNeg [229] also explore improving the fairness of recommendation data. Despite these efforts, achieving fairness in GFMs is still a significant challenge that needs further exploration.

7.3.4 Privacy

Privacy is a critical issue in Graph ML, particularly given the risk of these models inadvertently leaking sensitive information contained in graph data [233]–[235]. For example, Graph ML integrated with LLMs could potentially expose private user data, like browsing histories or social connections when generating outputs. This concern is especially pressing in highly data-sensitive areas such as healthcare or finance. To mitigate these privacy risks, [236] introduces Privacy-Preserving Prompt Tuning (RAPT) to protect user privacy through local differential privacy. Future exploration in LLM-enhanced Graph ML should also focus on integrating privacy-preserving technologies like differential privacy and federated learning to strengthen data security and user privacy.

7.4 Efficiency

While LLMs have proven effective in constructing GFMs, their operational efficiency, particularly in processing large and complex graphs, is still a significant challenge [237]. For example, the use of APIs like GPT4 for large-scale graph tasks can lead to high costs under current billing models. Additionally, deploying open-source large models (e.g., LLaMa) for parameter updates or just inference in local environments demands substantial computational resources and storage. Therefore, enhancing the efficiency of LLMs for graph tasks remains a critical issue. Recent studies introduce techniques like LoRA [146] and QLoRA [238] to fine-tune LLM parameters more efficiently. Furthermore, model pruning [239], [240] is also a promising method to increase efficiency by removing redundant parameters or structures from LLMs, thereby simplifying their application in graph machine learning.

8 Conclusion

In this survey, we have thoroughly reviewed the recent progress of graph applications and Graph ML in the era of LLMs, an emerging field in graph learning. We first review the evolution of Graph ML, and then delve into various methods of LLMs enhancing Graph ML. Due to the remarkable capabilities in various fields, LLMs have great potential to enhance Graph ML towards GFMs. We further explore the augmenting of LLMs with graphs, highlighting their ability to enhance LLM pre-training and inference. Additionally, we demonstrate their potential in diverse applications such as molecule discovery, knowledge graphs, and recommender systems. Despite their success, this field is still evolving and presents numerous opportunities for further advancements. Therefore, we further discuss several challenges and potential future directions. Overall, our survey aims to provide a systematic and comprehensive review to researchers and practitioners, inspiring future explorations in this promising field.

References

[1] H. Mao, J. Li, H. Shomer, B. Li, W. Fan, Y. Ma, T. Zhao, N. Shah, and J. Tang, "Revisiting link prediction: A data perspective," *arXiv preprint arXiv:2310.00793*, 2023.

[2] M. Hashemi, S. Gong, J. Ni, W. Fan, B. A. Prakash, and W. Jin, "A comprehensive survey on graph reduction: Sparsification, coarsening, and condensation," *arXiv preprint arXiv:2402.03358*, 2024.

[3] W. Fan, X. Zhao, Q. Li, T. Derr, Y. Ma, H. Liu, J. Wang, and J. Tang, "Adversarial attacks for black-box recommender systems via copying transferable cross-domain user profiles," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[4] J. Wu, W. Fan, J. Chen, S. Liu, Q. Li, and K. Tang, "Disentangled contrastive learning for social recommendation," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 4570–4574.

[5] J. Chen, W. Fan, G. Zhu, X. Zhao, C. Yuan, Q. Li, and Y. Huang, "Knowledge-enhanced black-box attacks for recommendations," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 108–117.

[6] T. Derr, Y. Ma, W. Fan, X. Liu, C. Aggarwal, and J. Tang, "Epidemic graph convolutional network," in *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM)*, 2020, pp. 160–168.

[7] Y. Ma and J. Tang, *Deep learning on graphs*. Cambridge University Press, 2021.

[8] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[9] J. Zhang, R. Xue, W. Fan, X. Xu, Q. Li, J. Pei, and X. Liu, "Linear-time graph neural networks for scalable recommendations," *arXiv preprint arXiv:2402.13973*, 2024.

[10] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *Advances in neural information processing systems*, vol. 33, pp. 5812–5823, 2020.

[11] J. Zeng and P. Xie, "Contrastive self-supervised learning for graph classification," in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 10 824–10 832.

[12] X. Xu, F. Zhou, K. Zhang, and S. Liu, "Ccgl: Contrastive cascade graph learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 4539–4554, 2022.

[13] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "Gcc: Graph contrastive coding for graph neural network pre-training," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 1150–1160.

[14] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[16] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. Corso, and J. Gao, "Unified vision-language pre-training for image captioning and vqa," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 13 041–13 049.

[17] W. Fan, Z. Zhao, J. Li, Y. Liu, X. Mei, Y. Wang, Z. Wen, F. Wang, X. Zhao, J. Tang, and Q. Li, "Recommender systems in the era of large language models (llms)," Aug. 2023.

[18] H. Mao, Z. Chen, W. Tang, J. Zhao, Y. Ma, T. Zhao, N. Shah, M. Galkin, and J. Tang, "Graph foundation models," *arXiv preprint arXiv:2402.02216*, 2024.

[19] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[20] R. Ye, C. Zhang, R. Wang, S. Xu, and Y. Zhang, "Natural language is all a graph needs," Aug. 2023.

[21] J. Guo, L. Du, H. Liu, M. Zhou, X. He, and S. Han, "Gpt4graph: Can large language models understand graph structured data ? an empirical evaluation and benchmarking," Jul. 2023.

[22] J. Tang, Y. Yang, W. Wei, L. Shi, L. Su, S. Cheng, D. Yin, and C. Huang, "Graphgpt: Graph instruction tuning for large language models," Oct. 2023.

[23] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[24] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[25] K. Duan, Q. Liu, T.-S. Chua, S. Yan, W. T. Ooi, Q. Xie, and J. He, "Simteg: A frustratingly simple approach improves textual graph learning," *arXiv preprint arXiv:2308.02565*, 2023.

[26] Z. Chen, H. Mao, H. Li, W. Jin, H. Wen, X. Wei, S. Wang, D. Yin, W. Fan, H. Liu, and J. Tang, "Exploring the potential of large language models (llms) in learning on graphs," Aug. 2023.

[27] E. Chien, W.-C. Chang, C.-J. Hsieh, H.-F. Yu, J. Zhang, O. Milenkovic, and I. S. Dhillon, "Node feature extraction by self-supervised multi-scale neighborhood prediction," *arXiv preprint arXiv:2111.00064*, 2021.

[28] J. Zhao, L. Zhuo, Y. Shen, M. Qu, K. Liu, M. Bronstein, Z. Zhu, and J. Tang, "Graphtext: Graph reasoning in text space," Oct. 2023.

[29] R. L. Logan IV, N. F. Liu, M. E. Peters, M. Gardner, and S. Singh, "Barack's wife hillary: Using knowledge-graphs for fact-aware language modeling," *arXiv preprint arXiv:1906.07241*, 2019.

[30] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg *et al.*, "Sparks of artificial general intelligence: Early experiments with gpt-4," *arXiv preprint arXiv:2303.12712*, 2023.

[31] H. Zhao, H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, and M. Du, "Explainability for large language models: A survey," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 2, pp. 1–38, 2024.

[32] L. Luo, Y.-F. Li, G. Haffari, and S. Pan, "Reasoning on graphs: Faithful and interpretable large language model reasoning," Oct. 2023.

[33] C. Wang, Y. Xu, Z. Peng, C. Zhang, B. Chen, X. Wang, L. Feng, and B. An, "keqing: knowledge-based question answering is a nature chain-of-thought mentor of llm," *arXiv preprint arXiv:2401.00426*, 2023.

[34] X. Guan, Y. Liu, H. Lin, Y. Lu, B. He, X. Han, and L. Sun, "Mitigating large language model hallucinations via autonomous knowledge graph-based retrofitting," *arXiv preprint arXiv:2311.13314*, 2023.

[35] X. Li, Y. Cao2, L. Pan, Y. Ma, and A. Sun, "Towards verifiable generation: A benchmark for knowledge-aware language model attribution," Oct. 2023.

[36] S. Wei, Y. Zhao, X. Chen, Q. Li, F. Zhuang, J. Liu, F. Ren, and G. Kou, "Graph learning and its advancements on large language models: A holistic survey," *arXiv preprint arXiv:2212.08966*, 2022.

[37] Z. Zhang, H. Li, Z. Zhang, Y. Qin, X. Wang, and W. Zhu, "Large graph models: A perspective," Aug. 2023.

[38] B. Jin, G. Liu, C. Han, M. Jiang, H. Ji, and J. Han, "Large language models on graphs: A comprehensive survey," *arXiv preprint arXiv:2312.02783*, 2023.

[39] Y. Li, Z. Li, P. Wang, J. Li, X. Sun, H. Cheng, and J. X. Yu, "A survey of graph meets large language model: Progress and future directions," *arXiv preprint arXiv:2311.12399*, 2023.

[40] J. Liu, C. Yang, Z. Lu, J. Chen, Y. Li, M. Zhang, T. Bai, Y. Fang, L. Sun, P. S. Yu *et al.*, "Towards graph foundation models: A survey and beyond," *arXiv preprint arXiv:2310.11829*, 2023.

[41] L. Wang, W. Fan, J. Li, Y. Ma, and Q. Li, "Fast graph condensation with structure-based neural tangent kernel," *arXiv preprint arXiv:2310.11046*, 2023.

[42] W. Fan, S. Wang, X.-y. Wei, X. Mei, and Q. Li, "Untargeted black-box attacks for social recommendations," *arXiv preprint arXiv:2311.07127*, 2023.

[43] M. Tsubaki, K. Tomii, and J. Sese, "Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences," *Bioinformatics*, vol. 35, no. 2, pp. 309–318, 2019.

[44] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The world wide web conference*, 2019, pp. 417–426.

[45] W. Fan, Y. Ma, Q. Li, J. Wang, G. Cai, J. Tang, and D. Yin, "A graph neural network framework for social recommendations," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[46] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.

[47] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

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

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

[50] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[51] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[53] Y. Li, X. Liang, Z. Hu, Y. Chen, and E. P. Xing, "Graph transformer," 2018.

[54] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," *Advances in neural information processing systems*, vol. 32, 2019.

[55] J. Baek, M. Kang, and S. J. Hwang, "Accurate learning of graph representations with graph multiset pooling," in *ICLR*, 2021.

[56] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proceedings of the web conference 2020*, 2020, pp. 2704–2710.

[57] J. Zhang, H. Zhang, C. Xia, and L. Sun, "Graph-bert: Only attention is needed for learning graph representations," *arXiv preprint arXiv:2001.05140*, 2020.

[58] J. Yang, Z. Liu, S. Xiao, C. Li, D. Lian, S. Agrawal, A. Singh, G. Sun, and X. Xie, "Graphformers: Gnn-nested transformers for representation learning on textual graph," *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 798–28 810, 2021.

[59] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, "Do transformers really perform badly for graph representation?" *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 877–28 888, 2021.

[60] S. Mitheran, A. Java, S. K. Sahu, and A. Shaikh, "Introducing self-attention to target attentive graph neural networks," *arXiv preprint arXiv:2107.01516*, 2021.

[61] D. Kreuzer, D. Beaini, W. Hamilton, V. Létourneau, and P. Tossou, "Rethinking graph transformers with spectral attention," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 618–21 629, 2021.

[62] V. P. Dwivedi and X. Bresson, "A generalization of transformer networks to graphs," *arXiv preprint arXiv:2012.09699*, 2020.

[63] C. Qian, H. Tang, Z. Yang, H. Liang, and Y. Liu, "Can large language models empower molecular property prediction?" Jul. 2023.

[64] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez *et al.*, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

[65] J. Zhao, M. Qu, C. Li, H. Yan, Q. Liu, R. Li, X. Xie, and J. Tang, "Learning on large-scale text-attributed graphs via variational inference," *arXiv preprint arXiv:2210.14709*, 2022.

[66] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: Decoding-enhanced bert with disentangled attention," *arXiv preprint arXiv:2006.03654*, 2020.

[67] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[68] C. Wu, S. Yin, W. Qi, X. Wang, Z. Tang, and N. Duan, "Visual chatgpt: Talking, drawing and editing with visual foundation models," *arXiv preprint arXiv:2303.04671*, 2023.

[69] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *ICML*, 2021.

[70] Y. Ding, Y. Ma, W. Fan, Y. Yao, T.-S. Chua, and Q. Li, "Fashionregen: Llm-empowered fashion report generation," *arXiv preprint arXiv:2403.06660*, 2024.

[71] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," *arXiv preprint arXiv:1809.10341*, 2018.

[72] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Deep graph contrastive representation learning," *arXiv preprint arXiv:2006.04131*, 2020.

[73] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, and J. Tang, "Graphmae: Self-supervised masked graph autoencoders," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 594–604.

[74] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *International conference on machine learning*. PMLR, 2020, pp. 4116–4126.

[75] J. Shang, T. Ma, C. Xiao, and J. Sun, "Pre-training of graph augmented transformers for medication recommendation," *arXiv preprint arXiv:1906.00346*, 2019.

[76] S. Li, X. Han, and J. Bai, "Adaptergnn: Efficient delta tuning improves generalization ability in graph neural networks," *arXiv preprint arXiv:2304.09595*, 2023.

[77] Y. Rong, Y. Bian, T. Xu, W. Xie, Y. Wei, W. Huang, and J. Huang, "Self-supervised graph transformer on large-scale molecular data," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 559–12 571, 2020.

[78] A. Gui, J. Ye, and H. Xiao, "G-adapter: Towards structure-aware parameter-efficient transfer learning for graph transformer networks," *arXiv preprint arXiv:2305.10329*, 2023.

[79] Q. Zhao, W. Ren, T. Li, X. Xu, and H. Liu, "Graphgpt: Graph learning with generative pre-trained transformers," *arXiv preprint arXiv:2401.00529*, 2023.

[80] B. Su, D. Du, Z. Yang, Y. Zhou, J. Li, A. Rao, H. Sun, Z. Lu, and J.-R. Wen, "A molecular multimodal foundation model associating molecule graphs with natural language," *arXiv preprint arXiv:2209.05481*, 2022.

[81] J. Zhu, X. Song, V. N. Ioannidis, D. Koutra, and C. Faloutsos, "Touchup-g: Improving feature representation through graph-centric finetuning," *arXiv preprint arXiv:2309.13885*, 2023.

[82] Z. Liu, X. Yu, Y. Fang, and X. Zhang, "Graphprompt: Unifying pre-training and downstream tasks for graph neural networks," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 417–428.

[83] M. Sun, K. Zhou, X. He, Y. Wang, and X. Wang, "Gppt: Graph pre-training and prompt tuning to generalize graph neural networks," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1717–1727.

[84] C. Gong, X. Li, J. Yu, C. Yao, J. Tan, C. Yu, and D. Yin, "Prompt tuning for multi-view graph contrastive learning," *arXiv preprint arXiv:2310.10362*, 2023.

[85] T. Fang, Y. M. Zhang, Y. Yang, C. Wang, and C. Lei, "Universal prompt tuning for graph neural networks," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[86] X. Sun, H. Cheng, J. Li, B. Liu, and J. Guan, "All in one: Multi-task prompting for graph neural networks," 2023.

[87] M. Chen, Z. Liu, C. Liu, J. Li, Q. Mao, and J. Sun, "Ultra-dp: Unifying graph pre-training with multi-task graph dual prompt," *arXiv preprint arXiv:2310.14845*, 2023.

[88] Q. Ge, Z. Zhao, Y. Liu, A. Cheng, X. Li, S. Wang, and D. Yin, "Enhancing graph neural networks with structure-based prompt," *arXiv preprint arXiv:2310.17394*, 2023.

[89] Q. Huang, H. Ren, P. Chen, G. Kržmanc, D. Zeng, P. Liang, and J. Leskovec, "Prodigy: Enabling in-context learning over graphs," *arXiv preprint arXiv:2305.12600*, 2023.

[90] Y. Zhu, J. Guo, and S. Tang, "Sgl-pt: A strong graph learner with graph prompt tuning," *arXiv preprint arXiv:2302.12449*, 2023.

[91] R. Shirkavand and H. Huang, "Deep prompt tuning for graph transformers," *arXiv preprint arXiv:2309.10131*, 2023.

[92] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *NeurIPS*, 2020.

[93] R. Thoppilan, D. De Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du *et al.*, "Lamda: Language models for dialog applications," *arXiv preprint arXiv:2201.08239*, 2022.

[94] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[95] N. Shinn, F. Cassano, A. Gopinath, K. R. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[96] J. Zhang, "Graph-toolformer: To empower llms with graph reasoning ability via prompt augmented by chatgpt," *arXiv preprint arXiv:2304.11116*, 2023.

[97] C. Mavromatis, V. N. Ioannidis, S. Wang, D. Zheng, S. Adeshina, J. Ma, H. Zhao, C. Faloutsos, and G. Karypis, "Train your own gnn teacher: Graph-aware distillation on textual graphs," *arXiv preprint arXiv:2304.10668*, 2023.

[98] P. Jiang, J. Rayan, S. P. Dow, and H. Xia, "Graphologue: Exploring large language model responses with interactive diagrams," *arXiv preprint arXiv:2305.11473*, 2023.

[99] C. M. Castro Nascimento and A. S. Pimentel, "Do large language models understand chemistry? a conversation with chatgpt," *Journal of Chemical Information and Modeling*, vol. 63, no. 6, pp. 1649–1655, 2023.

[100] A. M. Bran, S. Cox, A. D. White, and P. Schwaller, "Chemcrow: Augmenting large-language models with chemistry tools," *arXiv preprint arXiv:2304.05376*, 2023.

[101] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günnemann, E. Hüllermeier *et al.*, "Chatgpt for good? on opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, p. 102274, 2023.

[102] A. Gilson, C. W. Safranek, T. Huang, V. Socrates, L. Chi, R. A. Taylor, D. Chartash *et al.*, "How does chatgpt perform on the united states medical licensing examination? the implications of large language models for medical education and knowledge assessment," *JMIR Medical Education*, vol. 9, no. 1, p. e45312, 2023.

[103] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann, "Bloomberggpt: A large language model for finance," *arXiv preprint arXiv:2303.17564*, 2023.

[104] Y. Yang, M. C. S. Uy, and A. Huang, "Finbert: A pretrained language model for financial communications," *arXiv preprint arXiv:2006.08097*, 2020.

[105] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*. PMLR, 2017, pp. 1263–1272.

[106] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 974–983.

[107] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.

[108] H. Maron, H. Ben-Hamu, H. Serviansky, and Y. Lipman, "Provably powerful graph networks," *Advances in neural information processing systems*, vol. 32, 2019.

[109] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe, "Weisfeiler and leman go neural: Higher-order graph neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 4602–4609.

[110] P. Li, J. Wang, Y. Qiao, H. Chen, Y. Yu, X. Yao, P. Gao, G. Xie, and S. Song, "An effective self-supervised framework for learning expressive molecular global representations to drug discovery," *Briefings in Bioinformatics*, vol. 22, no. 6, p. bbab109, 2021.

[111] Z. Hu, C. Fan, T. Chen, K.-W. Chang, and Y. Sun, "Pre-training graph neural networks for generic structural feature extraction," *arXiv preprint arXiv:1905.13728*, 2019.

[112] C. hao, X. Runfeng, C. Xiangyang, Y. Zhou, W. Xin, X. Zhanwei, and Z. Kai, "Lkpnr: Llm and kg for personalized news recommendation framework," Aug. 2023.

[113] Z. Ni, X.-X. Deng, C. Tai, X.-Y. Zhu, X. Wu, Y.-J. Liu, and L. Zeng, "Grid: Scene-graph-based instruction-driven robotic task planning," Sep. 2023.

[114] X. He, X. Bresson, T. Laurent, A. Perold, Y. LeCun, and B. Hooi, "Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning," Oct. 2023.

[115] X. Ren, W. Wei, L. Xia, L. Su, S. Cheng, J. Wang, D. Yin, and C. Huang, "Representation learning with large language models for recommendation," Oct. 2023.

[116] W. Wei, X. Ren, J. Tang, Q. Wang, L. Su, S. Cheng, J. Wang, D. Yin, and C. Huang, "Llmrec: Large language models with graph augmentation for recommendation."

[117] H. Lyu, S. Jiang, H. Zeng, Y. Xia, and J. Luo, "Llm-rec: Personalized recommendation via prompting large language models," *arXiv preprint arXiv:2307.15780*, 2023.

[118] S. Balaji, R. Magar, Y. Jadhav, and A. B. Farimani, "Gpt-molberta: Gpt molecular features language model for molecular property prediction," Oct. 2023.

[119] J. Yu, Y. Ren, C. Gong, J. Tan, X. Li, and X. Zhang, "Empower text-attributed graphs learning with large language models (llms)," Oct. 2023.

[120] S. Sun, Y. Ren, C. Ma, and X. Zhang, "Large language models as topological structure enhancers for text-attributed graphs," *arXiv preprint arXiv:2311.14324*, 2023.

[121] H. Liu, J. Feng, L. Kong, N. Liang, D. Tao, Y. Chen, and M. Zhang, "One for all: Towards training one graph model for all classification tasks," *arXiv preprint arXiv:2310.00149*, 2023.

[122] Y. Hu, Z. Zhang, and L. Zhao, "Beyond text: A deep dive into large language models' ability on understanding graph data," Oct. 2023.

[123] H. Wang, S. Feng, T. He, Z. Tan, X. Han, and Y. Tsvetkov, "Can language models solve graph problems in natural language?" *arXiv preprint arXiv:2305.10037*, 2023.

[124] C. Liu and B. Wu, "Evaluating large language models on graphs: Performance insights and comparative analysis," Sep. 2023.

[125] Q. Wang, Z. Gao, and R. Xu, "Graph agent: Explicit reasoning agent for graphs," *arXiv preprint arXiv:2310.16421*, 2023.

[126] A. N. Rubungo, C. Arnold, B. P. Rand, and A. B. Dieng, "Llm-prop: Predicting physical and electronic properties of crystalline solids from their text descriptions," Oct. 2023.

[127] L. Wu, Z. Qiu, Z. Zheng, H. Zhu, and E. Chen, "Exploring large language model for graph data understanding in online job recommendations," *arXiv preprint arXiv:2307.05722*, 2023.

[128] Y. Shi, A. Zhang, E. Zhang, Z. Liu, and X. Wang, "Relm: Leveraging language models for enhanced chemical reaction prediction," *arXiv preprint arXiv:2310.13590*, 2023.

[129] J. Huang, X. Zhang, Q. Mei, and J. Ma, "Can llms effectively leverage graph structural information: When and why," Sep. 2023.

[130] B. Fatemi, J. Halcrow, and B. Perozzi, "Talk like a graph: Encoding graphs for large language models," Oct. 2023.

[131] J. Li, Y. Liu, W. Fan, X.-Y. Wei, H. Liu, J. Tang, and Q. Li, "Empowering molecule discovery for molecule-caption translation with large language models: A chatgpt perspective," *arXiv preprint arXiv:2306.06615*, 2023.

[132] R. Chen, T. Zhao, A. Jaiswal, N. Shah, and Z. Wang, "Llaga: Large language and graph assistant," *arXiv preprint arXiv:2402.08170*, 2024.

[133] Z. Guo, L. Xia, Y. Yu, Y. Wang, Z. Yang, W. Wei, L. Pang, T.-S. Chua, and C. Huang, "Graphedit: Large language models for graph structure learning," *arXiv preprint arXiv:2402.15183*, 2024.

[134] Z. Chai, T. Zhang, L. Wu, K. Han, X. Hu, X. Huang, and Y. Yang, "Graphllm: Boosting graph reasoning ability of large language model," Oct. 2023.

[135] Y. Tian, H. Song, Z. Wang, H. Wang, Z. Hu, F. Wang, N. V. Chawla, and P. Xu, "Graph neural prompting with large language models," Sep. 2023.

[136] Y. Liang, R. Zhang, L. Zhang, and P. Xie, "Drugchat: Towards enabling chatgpt-like capabilities on drug molecule graphs," May 2023.

[137] Y. Zhang, Z. Chen, W. Zhang, and H. Chen, "Making large language models perform better in knowledge graph completion," Oct. 2023.

[138] H. Zhao, S. Liu, C. Ma, H. Xu, J. Fu, Z.-H. Deng, L. Kong, and Q. Liu, "Gimlet: A unified graph-text model for instruction-based molecule zero-shot learning," Bioinformatics, Preprint, Jun. 2023.

[139] P. Liu, Y. Ren, and Z. Ren, "Git-mol: A multi-modal large language model for molecular science with graph, image, and text," Aug. 2023.

[140] Y. Luo, J. Zhang, S. Fan, K. Yang, Y. Wu, M. Qiao, and Z. Nie, "Biomedgpt: Open multimodal generative pre-trained transformer for biomedicine," *arXiv preprint arXiv:2308.09442*, 2023.

[141] H. Guo, M. Huo, R. Zhang, and P. Xie, "Proteinchat: Towards achieving chatgpt-like functionalities on protein 3d structures," 2023.

[142] Y. Qin, X. Wang, Z. Zhang, and W. Zhu, "Disentangled representation learning with large language models for text-attributed graphs," *arXiv preprint arXiv:2310.18152*, 2023.

[143] X. He, Y. Tian, Y. Sun, N. V. Chawla, T. Laurent, Y. LeCun, X. Bresson, and B. Hooi, "G-retriever: Retrieval-augmented generation for textual graph understanding and question answering," *arXiv preprint arXiv:2402.07630*, 2024.

[144] B. Perozzi, B. Fatemi, D. Zelle, A. Tsitsulin, M. Kazemi, R. Al-Rfou, and J. Halcrow, "Let your graph do the talking: Encoding structured data for llms," *arXiv preprint arXiv:2402.05862*, 2024.

[145] L. Xia, B. Kao, and C. Huang, "Opengraph: Towards open graph foundation models," *arXiv preprint arXiv:2403.01121*, 2024.

[146] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[147] M. Himsolt, "Gml: Graph modelling language," *University of Passau*, 1997.

[148] U. Brandes, M. Eiglsperger, J. Lerner, and C. Pich, "Graph markup language (graphml)," 2013.

[149] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," *arXiv preprint arXiv:2307.03172*, 2023.

[150] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *arXiv preprint arXiv:2301.12597*, 2023.

[151] S. Gui, X. Li, L. Wang, and S. Ji, "Good: A graph out-of-distribution benchmark," *Advances in Neural Information Processing Systems*, vol. 35, pp. 2059–2073, 2022.

[152] L. Hu, Z. Liu, Z. Zhao, L. Hou, L. Nie, and J. Li, "A survey of knowledge enhanced pre-trained language models," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–19, 2023.

[153] G. Agrawal, T. Kumarage, Z. Alghami, and H. Liu, "Can knowledge graphs reduce hallucinations in llms?: A survey," *arXiv preprint arXiv:2311.07914*, 2023.

[154] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying large language models and knowledge graphs: A roadmap," Jun. 2023.

[155] Y. Sun, S. Wang, S. Feng, S. Ding, C. Pang, J. Shang, J. Liu, X. Chen, Y. Zhao, Y. Lu *et al.*, "Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation," *arXiv e-prints*, pp. arXiv–2107, 2021.

[156] L. Fang, Y. Luo, K. Feng, K. Zhao, and A. Hu, "A knowledge-enriched ensemble method for word embedding and multi-sense embedding," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[157] Q. Li, D. Wang, S. F. K. Song, Y. Zhang, and G. Yu, "Oerl: Enhanced representation learning via open knowledge graphs," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[158] F. Moiseev, Z. Dong, E. Alfonseca, and M. Jaggi, "Skill: Structured knowledge infusion for large language models," May 2022.

[159] N. Poerner, U. Waltinger, and H. Schütze, "E-bert: Efficient-yet-effective entity embeddings for bert," *arXiv preprint arXiv:1911.03681*, 2019.

[160] C. Rosset, C. Xiong, M. Phan, X. Song, P. Bennett, and S. Tiwary, "Knowledge-aware language model pretraining," *arXiv preprint arXiv:2007.00655*, 2020.

[161] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang, "K-bert: Enabling language representation with knowledge graph," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 03, pp. 2901–2908, Apr. 2020.

[162] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "Ernie: Enhanced language representation with informative entities," *arXiv preprint arXiv:1905.07129*, 2019.

[163] Y. Su, X. Han, Z. Zhang, Y. Lin, P. Li, Z. Liu, J. Zhou, and M. Sun, "Cokebert: Contextual knowledge selection and embedding towards enhanced pre-trained language models," *AI Open*, vol. 2, pp. 127–134, Jan. 2021.

[164] L. He, S. Zheng, T. Yang, and F. Zhang, "Klmo: Knowledge graph enhanced pretrained language model with fine-grained relationships," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4536–4542.

[165] R. Wang, D. Tang, N. Duan, Z. Wei, X. Huang, J. Ji, G. Cao, D. Jiang, and M. Zhou, "K-adapter: Infusing knowledge into pre-trained models with adapters," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 1405–1418.

[166] M. Kang, J. Baek, and S. J. Hwang, "Kala: knowledge-augmented language model adaptation," *arXiv preprint arXiv:2204.10555*, 2022.

[167] Q. Lu, D. Dou, and T. H. Nguyen, "Parameter-efficient domain knowledge integration from multiple sources for biomedical pre-trained language models," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 3855–3865.

[168] I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto, "Luke: Deep contextualized entity representations with entity-aware self-attention," *arXiv preprint arXiv:2010.01057*, 2020.

[169] Y. Xu, M. Namazifar, D. Hazarika, A. Padmakumar, Y. Liu, and D. Hakkani-Tür, "Kilm: Knowledge injection into encoder-decoder language models," Feb. 2023.

[170] T. Shen, Y. Mao, P. He, G. Long, A. Trischler, and W. Chen, "Exploiting structured knowledge in text via graph-guided representation learning," *arXiv preprint arXiv:2004.14224*, 2020.

[171] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: Improving pre-training by representing and predicting spans," *Transactions of the association for computational linguistics*, vol. 8, pp. 64–77, 2020.

[172] D. Yu, C. Zhu, Y. Yang, and M. Zeng, "Jaket: Joint pre-training of knowledge graph and language understanding," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, pp. 11 630–11 638, Jun. 2022.

[173] M. Yasunaga, A. Bosselut, H. Ren, X. Zhang, C. D. Manning, P. S. Liang, and J. Leskovec, "Deep bidirectional language-knowledge graph pretraining," *Advances in Neural Information Processing Systems*, vol. 35, pp. 37 309–37 323, 2022.

[174] Y. Qin, Y. Lin, R. Takanobu, Z. Liu, P. Li, H. Ji, M. Huang, M. Sun, and J. Zhou, "Erica: Improving entity and relation understanding for pre-trained language models via contrastive learning," *arXiv preprint arXiv:2012.15022*, 2020.

[175] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, and J. Tang, "Kepler: A unified model for knowledge embedding and pre-trained language representation," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 176–194, 2021.

[176] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, and H. Wang, "Ernie 2.0: A continual pre-training framework for language understanding," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 8968–8975.

[177] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[178] J. Baek, A. F. Aji, and A. Saffari, "Knowledge-augmented language model prompting for zero-shot knowledge graph question answering," *arXiv preprint arXiv:2306.04136*, 2023.

[179] Y. Wu, N. Hu, S. Bi, G. Qi, J. Ren, A. Xie, and W. Song, "Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering," Sep. 2023.

[180] Y. Wen, Z. Wang, and J. Sun, "Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models," Sep. 2023.

[181] C. Feng, X. Zhang, and Z. Fei, "Knowledge solver: Teaching llms to search for domain knowledge from knowledge graphs," Sep. 2023.

[182] S. Saxena, S. Prasad, M. Prakash, A. Shankar, V. Vaddina, S. Gopalakrishnan *et al.*, "Minimizing factual inconsistency and hallucination in large language models," *arXiv preprint arXiv:2311.13878*, 2023.

[183] Y. Shi, H. Ma, W. Zhong, G. Mai, X. Li, T. Liu, and J. Huang, "Chatgraph: Interpretable text classification by converting chatgpt knowledge to graphs," *arXiv preprint arXiv:2305.03513*, 2023.

[184] Z. Chen, J. Chen, M. Gaidhani, A. Singh, and M. Sra, "Xplainllm: A qa explanation dataset for understanding llm decision-making," *arXiv preprint arXiv:2311.08614*, 2023.

[185] W. Fan, Q. Li, and M. Cheng, "Deep modeling of social relations for recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[186] W. Fan, T. Derr, Y. Ma, J. Wang, J. Tang, and Q. Li, "Deep adversarial social recommendation," in *28th International Joint Conference on Artificial Intelligence (IJCAI-19)*. International Joint Conferences on Artificial Intelligence, 2019, pp. 1351–1357.

[187] W. Fan, Y. Ma, D. Yin, J. Wang, J. Tang, and Q. Li, "Deep social collaborative filtering," in *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, pp. 305–313.

[188] B. Yin, J. Xie, Y. Qin, Z. Ding, Z. Feng, X. Li, and W. Lin, "Heterogeneous knowledge fusion: A novel approach for personalized recommendation via llm," in *Proceedings of the 17th ACM Conference on Recommender Systems*, 2023, pp. 599–601.

[189] Z. Tang, Z. Huan, Z. Li, X. Zhang, J. Hu, C. Fu, J. Zhou, and C. Li, "One model for all: Large language models are domain-agnostic recommendation systems," Oct. 2023.

[190] S. Dai, N. Shao, H. Zhao, W. Yu, Z. Si, C. Xu, Z. Sun, X. Zhang, and J. Xu, "Uncovering chatgpt's capabilities in recommender systems," *arXiv preprint arXiv:2305.02182*, 2023.

[191] H. Wang, X. Liu, W. Fan, X. Zhao, V. Kini, D. Yadav, F. Wang, Z. Wen, J. Tang, and H. Liu, "Rethinking large language model architectures for sequential recommendations," *arXiv preprint arXiv:2402.09543*, 2024.

[192] Y. Xi, W. Liu, J. Lin, J. Zhu, B. Chen, R. Tang, W. Zhang, R. Zhang, and Y. Yu, "Towards open-world recommendation with knowledge augmentation from large language models," *arXiv preprint arXiv:2306.10933*, 2023.

[193] J. Wu, Q. Liu, H. Hu, W. Fan, S. Liu, Q. Li, X.-M. Wu, and K. Tang, "Leveraging large language models (llms) to empower training-free dataset condensation for content-based recommendation," *arXiv preprint arXiv:2310.09874*, 2023.

[194] J. Liu, C. Liu, R. Lv, K. Zhou, and Y. Zhang, "Is chatgpt a good recommender? a preliminary study," *arXiv preprint arXiv:2304.10149*, 2023.

[195] K. Bao, J. Zhang, Y. Zhang, W. Wang, F. Feng, and X. He, "Tallrec: An effective and efficient tuning framework to align large language model with recommendation," *arXiv preprint arXiv:2305.00447*, 2023.

[196] W. Hua, Y. Ge, S. Xu, J. Ji, and Y. Zhang, "Up5: Unbiased foundation model for fairness-aware recommendation," *arXiv preprint arXiv:2305.12090*, 2023.

[197] C. Huang, T. Yu, K. Xie, S. Zhang, L. Yao, and J. McAuley, "Foundation models for recommender systems: A survey and new perspectives," *arXiv preprint arXiv:2402.11143*, 2024.

[198] R. Yang, L. Fang, and Y. Zhou, "Cp-kgc: Constrained-prompt knowledge graph completion with large language models," Oct. 2023.

[199] L. Yao, J. Peng, C. Mao, and Y. Luo, "Exploring large language models for knowledge graph completion," Sep. 2023.

[200] J. Kim, Y. Kwon, Y. Jo, and E. Choi, "Kg-gpt: A general framework for reasoning on knowledge graphs using large language models," Oct. 2023.

[201] H. Luo, H. E, Z. Tang, S. Peng, Y. Guo, W. Zhang, C. Ma, G. Dong, M. Song, and W. Lin, "Chatkbqa: A generate-then-retrieve framework for knowledge base question answering with fine-tuned large language models," Oct. 2023.

[202] J. Jiang, K. Zhou, Z. Dong, K. Ye, W. X. Zhao, and J.-R. Wen, "Structgpt: A general framework for large language model to reason over structured data," *arXiv preprint arXiv:2305.09645*, 2023.

[203] L. Luo, J. Ju, B. Xiong, Y.-F. Li, G. Haffari, and S. Pan, "Chatrule: Mining logical rules with large language models for knowledge graph reasoning," Sep. 2023.

[204] L.-P. Meyer, J. Frey, K. Junghanns, F. Brei, K. Bulert, S. Gründer-Fahrer, and M. Martin, "Developing a scalable benchmark for assessing large language models in knowledge graph engineering," Aug. 2023.

[205] M. Galkin, X. Yuan, H. Mostafa, J. Tang, and Z. Zhu, "Towards foundation models for knowledge graph reasoning," *arXiv preprint arXiv:2310.04562*, 2023.

[206] W. Tang, R. Liu, H. Wen, X. Dai, J. Ding, H. Li, W. Fan, Y. Xie, and J. Tang, "A general single-cell analysis framework via conditional diffusion generative models," *bioRxiv*, pp. 2023–10, 2023.

[207] Y. Cao, Z.-Q. Yang, X.-L. Zhang, W. Fan, Y. Wang, J. Shen, D.-Q. Wei, Q. Li, and X.-Y. Wei, "Identifying the kind behind smiles—anatomical therapeutic chemical classification using structure-only representations," *Briefings in Bioinformatics*, vol. 23, no. 5, p. bbac346, 2022.

[208] H. Cao, Z. Liu, X. Lu, Y. Yao, and Y. Li, "Instructmol: Multi-modal integration for building a versatile and reliable molecular assistant in drug discovery," *arXiv preprint arXiv:2311.16208*, 2023.

[209] Z. Zhao, D. Ma, L. Chen, L. Sun, Z. Li, H. Xu, Z. Zhu, S. Zhu, S. Fan, G. Shen *et al.*, "Chemdfm: Dialogue foundation model for chemistry," *arXiv preprint arXiv:2401.14818*, 2024.

[210] C. Galindo, J.-A. Fernández-Madrigal, J. González, and A. Saffiotti, "Robot task planning using semantic maps," *Robotics and autonomous systems*, vol. 56, no. 11, pp. 955–966, 2008.

[211] G. Chalvatzaki, A. Younes, D. Nandha, A. T. Le, L. F. R. Ribeiro, and I. Gurevych, "Learning to reason over scene graphs: A case study of finetuning gpt-2 into a robot language model for grounded task planning," *Frontiers in Robotics and AI*, vol. 10, p. 1221739, Aug. 2023.

[212] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning," Sep. 2023.

[213] Y. Zhen, S. Bi, L. Xing-tong, P. Wei-qin, S. Hai-peng, C. Zi-rui, and F. Yi-shu, "Robot task planning based on large language model representing knowledge with directed graph structures," Jun. 2023.

[214] H. Liu, Y. Wang, W. Fan, X. Liu, Y. Li, S. Jain, Y. Liu, A. K. Jain, and J. Tang, "Trustworthy ai: A computational perspective," *arXiv preprint arXiv:2107.06641*, 2021.

[215] W. Fan, X. Zhao, X. Chen, J. Su, J. Gao, L. Wang, Q. Liu, Y. Wang, H. Xu, L. Chen *et al.*, "A comprehensive survey on trustworthy recommender systems," *arXiv preprint arXiv:2209.10117*, 2022.

[216] W. Fan, W. Jin, X. Liu, H. Xu, X. Tang, S. Wang, Q. Li, J. Tang, J. Wang, and C. Aggarwal, "Jointly attacking graph neural network and its explanations," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023.

[217] W. Fan, T. Derr, X. Zhao, Y. Ma, H. Liu, J. Wang, J. Tang, and Q. Li, "Attacking black-box recommendations via copying cross-domain user profiles," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 1583–1594.

[218] N. Jain, A. Schwarzschild, Y. Wen, G. Somepalli, J. Kirchenbauer, P.-y. Chiang, M. Goldblum, A. Saha, J. Geiping, and T. Goldstein, "Baseline defenses for adversarial attacks against aligned language models," *arXiv preprint arXiv:2309.00614*, 2023.

[219] D. Bespalov, S. Bhabesh, Y. Xiang, L. Zhou, and Y. Qi, "Towards building a robust toxicity predictor," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, 2023, pp. 581–598.

[220] Y. Chen, H. Yang, Y. Zhang, K. Ma, T. Liu, B. Han, and J. Cheng, "Understanding and improving graph injection attack by promoting unnoticeability," *arXiv preprint arXiv:2202.08057*, 2022.

[221] X. Zou, Q. Zheng, Y. Dong, X. Guan, E. Kharlamov, J. Lu, and J. Tang, "Tdgia: Effective injection attacks on graph neural networks," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2461–2471.

[222] H. Dai, H. Li, T. Tian, X. Huang, L. Wang, J. Zhu, and L. Song, "Adversarial attack on graph structured data," in *International conference on machine learning*. PMLR, 2018, pp. 1115–1124.

[223] D. Zügner, A. Akbarnejad, and S. Günnemann, "Adversarial attacks on neural networks for graph data," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 2847–2856.

[224] A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How does llm safety training fail?" *arXiv preprint arXiv:2307.02483*, 2023.

[225] Z. Zhang, G. Zhang, B. Hou, W. Fan, Q. Li, S. Liu, Y. Zhang, and S. Chang, "Certified robustness for large language models with self-denoising," *arXiv preprint arXiv:2307.07171*, 2023.

[226] S. Geng, S. Liu, Z. Fu, Y. Ge, and Y. Zhang, "Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5)," in *Proceedings of the 16th ACM Conference on Recommender Systems*, 2022, pp. 299–315.

[227] H. Liu, J. Dacon, W. Fan, H. Liu, Z. Liu, and J. Tang, "Does gender matter? towards fairness in dialogue systems," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 4403–4416.

[228] G. Zhang, Y. Zhang, Y. Zhang, W. Fan, Q. Li, S. Liu, and S. Chang, "Fairness reprogramming," in *Thirty-sixth Conference on Neural Information Processing Systems*, 2022.

[229] X. Chen, W. Fan, J. Chen, H. Liu, Z. Liu, Z. Zhang, and Q. Li, "Fairly adaptive negative sampling for recommendations," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 3723–3733.

[230] K. Webster, X. Wang, I. Tenney, A. Beutel, E. Pitler, E. Pavlick, J. Chen, E. Chi, and S. Petrov, "Measuring and reducing gendered correlations in pre-trained models," *arXiv preprint arXiv:2010.06032*, 2020.

[231] Y. Guo, Y. Yang, and A. Abbasi, "Auto-debias: Debiasing masked language models with automated biased prompts," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 1012–1023.

[232] J. He, M. Xia, C. Fellbaum, and D. Chen, "Mabel: Attenuating gender bias using textual entailment data," *arXiv preprint arXiv:2210.14975*, 2022.

[233] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. B. Brown, D. Song, U. Erlingsson *et al.*, "Extracting training data from large language models." in *USENIX Security Symposium*, vol. 6, 2021.

[234] J. Huang, H. Shao, and K. C.-C. Chang, "Are large pre-trained language models leaking your personal information?" *arXiv preprint arXiv:2205.12628*, 2022.

[235] J. Zhang, L. Wang, S. Wang, and W. Fan, "Graph unlearning with efficient partial retraining," *arXiv preprint arXiv:2403.07353*, 2024.

[236] Y. Li, Z. Tan, and Y. Liu, "Privacy-preserving prompt tuning for large language model services," *arXiv preprint arXiv:2305.06212*, 2023.

[237] R. Xue, X. Shen, R. Yu, and X. Liu, "Efficient large language models fine-tuning on graphs," *arXiv preprint arXiv:2312.04737*, 2023.

[238] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *arXiv preprint arXiv:2305.14314*, 2023.

[239] X. Ma, G. Fang, and X. Wang, "Llm-pruner: On the structural pruning of large language models," *arXiv preprint arXiv:2305.11627*, 2023.

[240] M. Xia, T. Gao, Z. Zeng, and D. Chen, "Sheared llama: Accelerating language model pre-training via structured pruning," *arXiv preprint arXiv:2310.06694*, 2023.