Beyond Text: A Deep Dive into Large Language Models' Ability on Understanding Graph Data

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large language models (LLMs) have achieved impressive performance on many natural language processing tasks. However, their capabilities on graph-structured data remain relatively unexplored. In this paper, we conduct a series of experiments benchmarking leading LLMs on diverse graph prediction tasks spanning node, edge, and graph levels. We aim to assess whether LLMs can effectively process graph data and leverage topological structures to enhance performance, compared to specialized graph neural networks. Through varied prompt formatting and task/dataset selection, we analyze how well LLMs can interpret and utilize graph structures. By comparing LLMs' performance with specialized graph models, we offer insights into the strengths and limitations of employing LLMs for graph analytics. Our findings provide insights into LLMs' capabilities and suggest avenues for further exploration in applying them to graph analytics.

1 Introduction

In recent years, there have been unprecedented advancements in large language models (LLMs) [28] such as Transformers [33], BERT [7], GPT [4], and their variants. LLMs can be treated as foundation models that can be readily applied to diverse downstream tasks with little adaptation [4, 16, 19]. These models have achieved state-of-the-art results on many natural language processing tasks including text classification, machine translation, sentiment analysis, and text summarization [42]. Significantly, advancements in architectures and training methodologies have given rise to emergent capabilities, setting state-of-the-art models like GPT-3.5 [4], GPT-4 [26], Claude-2 [2], BARD [10], LlaMA [31], and LlaMA-2 [32] apart from their predecessors. For instance, in-context learning [22] and zero-shot capabilities [16, 35] enable these models to generalize across tasks for which they were not explicitly trained. This is confirmed by their excellent performance in complex activities such as mathematical reasoning and Question Answering (QA) systems.

However, most of the tasks that Large Language Models (LLMs) surpassed previous benchmarks are Natural Language Processing (NLP) tasks involving sequential data. Graph-structured data presents additional complexity beyond sequences as it contains rich topological connections between entities that must be modeled along with node, edge, and graph attributes. Graph-structured data is ubiquitous across many domains, including social networks [24], knowledge graphs [27], molecular structures [37], and transportation networks [3]. While LLMs have shown powerful reasoning and generalization capabilities in sequential data, it remains unclear if they can handle structural information beyond context when applied to graph-structured data. This raises a compelling research question: Can the strengths of LLMs be extended to graph-structured data, enabling them to exhibit significant predictive ability? Further, can they compete with state-of-the-art models specialized for graph data, such as Graph Neural Networks (GNNs)?

Submitted to the Workshop on New Frontiers in Graph Learning at NeurIPS 2023. Do not distribute.

To comprehensively study the capabilities of LLMs on graph-structured data, we conduct a series of empirical experiments with leading LLMs on diverse graph-based tasks that span node-, edge-, and graph-level predictions. By comparing their performance to specialized graph models like GNNs, we aim to assess the potential strengths and limitations of LLMs in this domain. Critically, by altering the input prompt formats, we aim to evaluate how effectively LLMs can extract and leverage the underlying structural information from the graph to enhance their performance in subsequent tasks. Additionally, we explore the importance of the structural data across different task dimensions spanning node, edge, and graph levels as well as diverse dataset domains such as citation networks, social networks, and chemical networks.

Broadly, this paper focuses on studying the central question of investigating the capabilities of LLMs on graph-structured data from three perspectives:

- **Can LLMs effectively process graph analytics tasks even without explicit graph structure?** Given that LLMs have already shown the capability to leverage contextual information for human-like reasoning in many NLP tasks, it becomes intriguing to assess whether they can attain substantial predictive performance on graph data tasks, even in the absence of structural information.
- **How well can LLMs interpret graph structures to enhance downstream task performance?** It is essential to investigate to what extent LLMs can perceive and interpret important graph structures. Furthermore, it is imperative to understand whether such recognition can influence and enhance performance in subsequent tasks.
- **How do task dimensions and dataset domains affect LLMs' ability to handle structured data?** LLMs' ability in identifying pivotal structural information for predictions can be influenced by specific tasks and data domains. For example, node-level tasks may heavily rely on entity attribute interpretation, while graph-level tasks may demand comprehensive understanding of intricate inter-node interations. Also, the distinct topologies properties to various dataset domains, whether derived from intricate social networks or sophisticated molecular structures, further influence the proficiency with which LLMs decipher and manage structured data.

The subsequent sections of this paper are structured as follows: We initiate with an extensive literature review, highlighting the recent advancements of LLMs within graph domains. Subsequent to this, we present our comprehensive findings on benchmarking LLMs on graph data, aiming to address the aforementioned research questions. This is accompanied by a detailed discussion, delving into the depth of our discoveries across varied experimental setups. We conclude by summarizing the key points and proposing ideas for future explorations.

2 Related Works

Large language models for graph-structured data. In recent literature, a few preliminary studies [40, 5, 36, 11] have made attempts to uncover the potential of LLMs in handling graph-structured data. Unfortunately, a comprehensive examination of LLMs' capacity to extract and harness crucial topological structures across diverse prompt settings, task levels, and datasets remains underexplored. Both Chen et al.[5] and Guo et al.[11] proposed to apply LLMs directly on graph data. Their research primarily focus on the node classification task, constrained to a selected few datasets within the citation network domain, and thereby fails to offer a thorough exploration of LLMs' ability over diverse task levels and datasets. In addition, Ye et al.[40] fine-tuned LLMs on a designated dataset to outperform GNN, underscoring a distinct research objective compared to our study which emphasizes the intrinsic proficiency of LLMs in understanding and exploiting graph structures. Meanwhile, Wei et al.[36] treated LLMs as autonomous agents within graph data, which is less relevant to the core focus of our paper.

Graph neural networks. In recent years, graph neural networks (GNNs) [14, 6, 25, 9, 12, 38, 23, 41, 18] have emerged as a powerful deep learning approach for graph analysis and learning. GNNs operate by propagating information along edges of the graph and aggregating neighborhood representations for each node. The expressive power of GNNs to learn from graph structure makes them well-suited for analyzing complex relational data [38, 43, 20]. Unlike standard deep neural networks which operate on regular grids, GNNs can leverage the topological structure of graphs and have achieved state-of-the-art performance on tasks such as node classification [14], link prediction [17], and graph classification [8].

3 Experiments

Datasets. We conducted the experiments on 5 commonly used graph benchmark datasets for node-level, edge-level and graph-level tasks: CORA [30], PUBMED [30], OGBN-ARXIV [13], WORDNET18 [21] and REDDIT [12]. Brief descriptions of the datasets are shown in Table 1.

We selected these five datasets for our preliminary experiments due to their rich contextual information present in the attributes of nodes, edges, and graphs. Specifically, CORA, PUBMED and OGBN-ARXIV are citation network, where each node represents a research paper while an edge between two nodes indicates that there is a citation relationship between them. Edge in WORDNET18 links two synsets that are regarded as nodes. REDDIT came from Reddit posts, in which each node represents a post and two nodes are connected if the same user comments on two posts. The specifics regarding their textual features are as follows:

- CORA: Each node represents a paper in the domain of Artificial Intelligence, containing the information about its title and abstract. Each paper belongs to one of the following 7 categories: ['Case_Based', 'Theory', 'Genetic_Algorithms', 'Probabilistic_Methods', 'Neural_Networks', 'Rule_Learning', 'Reinforcement_Learning']. An edge from one node to another indicates the first paper cited the second one.
- PUBMED: Each node represents a scientific publication from PubMed database pertaining to diabetes. The node textual information contains keywords from its abstract and text body. Each paper belongs to one of the following 3 categories: ['Diabetes Mellitus, Experimental', 'Diabetes Mellitus Type 1', 'Diabetes Mellitus Type 2']. An edge from one node to another indicates the first paper cited the second one.
- OGBN-ARXIV: Each node represents a research paper, containing the information about its title and abstract. Each paper belongs to one of 40 categories on arxiv.cs such as 'AI' (Artificial Intelligence). An edge leading from one node to another signifies that the first paper cites the second one.
- WORDNET18: Each node represents a synset, containing a description. An edge between two nodes indicate their relation such as 'furniture', 'includes', or 'bed'. Each edge belongs to one of 18 relationships.
- REDDIT: Each node corresponds to a post made by a user, which contains descriptions or discussions about a particular topic. Each graph symbolizes a subreddit (or community), with affiliations to one out of 29,651 distinct communities, for instance, 'math'.

Choices of LLMs. We opted to utilize OpenAI's state-of-the-art models, GPT-3.5 (GPT) and GPT-4, via their API system, based on a balance between performance and cost considerations. We adopted GPT with the latest versions (*gpt-3.5-turbo-16k* and *gpt-4*) in experiments.

Implementation Details. For node classification task, we follow the same train-test split of CORA, PUBMED and OGBN-ARXIV as established in semi-supervised GNN methods [14, 38]. For link prediction on CORA, PUBMED and WORDNET18, a random 15% of the links from the graph and the same number of negative-edge node pairs are packed into the test sets. For graph classification,

Dataset	**#Node**	**#Edge**	**#Task**	**Metric**
CORA	2,708	5,278	7-class node classifi. &Link Prediction	Accuracy
PUBMED	19,717	44,324	3-class node classifi. &Link Prediction	Accuracy
OGBN-ARXIV	169,343	1,166,243	40-class node classifi.	Accuracy
Dataset	**#Entity**	**#Relation**	**#Task**	**Metric**
WORDNET18	40,943	18	18-class link classifi.	Accuracy
Dataset	**#Node**	**#Subgraph**	**#Task**	**Metric**
REDDIT	3,848,330	29,651	70-class subgraph classifi.	Accuracy

Table 1: Statistics of the datasets. For REDDIT, it actually contains 29,651 subreddits (classes). Here we only randomly sampled 70 communities for graph classification task in each run.

in each run, we randomly selected 70 communities. Experiments conducted on WORDNET18 and the retrieval test for CORA employed few-shot prompts. Conversely, all other experiments leveraged zero-shot prompts. We executed each experiment thrice, subsequently averaging the results."

Comparison GNN Methods. On node-level tasks, we choose the semi-supervised results from Graph Neural Network (GNN) [29], Graph Convolutional Network (GCN) [15] and Graph Attention Network (GAT) [34] to compare with performance from LLMs. On edge-level tasks, we consider Graph Auto-Encoder (GAE) [1], Graph InfoClust (GIC) [39]. It is worth noting this is not an absolutely fair comparison. Since LLMs operate under zero-shot or few-shot settings, where GNNs require a training set for parameter optimization. Additionally, potential data leakage during the LLMs' training process remains a concern. However, these studies aim to offer a foundational understanding of LLMs' proficiency in understanding graph data structures and forecasting downstream tasks.

Model	Node-level		
	CORA	PUBMED	ARXIV
GCN-64*	0.814	0.790	0.731
GAT	**0.832**	0.790	**0.742**
GNN	0.829	0.738	0.721
GPT-3.5	0.627	0.673	0.516
GPT-4	0.432	0.821	0.642
GPT-3.5*	0.647	0.712	0.509
GPT-4*	0.531	**0.833**	0.673
GPT-3.5$^\oplus$	0.656	0.704	0.445
GPT-4$^\oplus$	-	0.814	-
GPT-3.5$^\bullet$	0.054	-	-
GPT-4$^\bullet$	0.047	-	-

Table 2: Average accuracy on node classification tasks. [**No structure information**] **GPT-3.5** and **GPT-3.5*** mean zero-shot and few-shot prompt strategy. [**With structure information**] **GPT-3.5**$^\oplus$ and **GPT-3.5**$^\bullet$ mean prompts contain neighbors' information and retrieval requires, respectively.

Model	Edge-level		
	CORA	PUBMED	WORDNET
GAE	0.793	**0.923**	-
GIC	0.812	0.775	-
GNN	0.739	0.528	-
GPT-3.5	0.512	0.116	0.134
GPT-4	0.578	0.132	0.169
GPT-3.5°	0.646	0.114	-
GPT-4°	**0.967**	0.143	-

Table 3: Average accuracy on link prediction tasks. **GPT-3.5**$^\circ$ means adding structural information into prompt like Table 6. There are only triples for entries in WORDNET18, which makes there is no connected graph structure for it.

Model	Number of labels									
	1	5	10	15	20	30	40	50	60	70
GPT-3.5	0.773	0.662	0.618	0.594	0.628	0.604	0.638	0.536	0.618	0.507
GPT-4	0.957	0.886	0.895	0.843	0.795	-	-	-	-	-

Table 4: Average performance of GPT-3.5 and GPT-4 on REDDIT when possible labels increase from 1 to 70. Results on GPT-4 with more labels are not available due to input limit of prompt length.

3.1 Node-level task

Driven by the goal of investigating LLMs' capabilities in discerning patterns within textual graphs and leveraging this for downstream tasks, we crafted three distinct prompts for our node-level prediction task experiments: (1) absence of graph topology descriptions; (2) straightforward presentation of all neighborhood data to the LLM; and (3) a retrieval-based prompt guiding the LLM to extract task-centric structural details. Examples of these prompts can be found in Table 6.

LLMs' zero-shot or few-shot ability on node classificaiton tasks is still usually weaker than the semi-supervised GNN performance. This may arguably suggests that general LLMs still can not surpass the specialized graph models on node classification task. As indicated in Table 2, GPT-4 outperforms the GNN models in zero-shot and few-shot settings on PUBMED, but this superiority isn't observed on CORA and OGBN-ARXIV. We hypothesize three potential reasons for this discrepancy: 1. Fewer categories; 2. Less semantic overlap between categories; 3. Questionable groundtruth categories. GPT's 'wrong' predictions about citation networks are mostly reasonable. Particularly,

Node-level Task	Structure?	Prompt to GPT
Zero-shot (CORA & PUBMED & OGBN-ARXIV)	No	The title of one paper is <Title> and its abstract is <Abstract>. Here are possible categories: <Categories>. Which category does this paper belong to? You can only output one category.
	Yes	The title of one paper is <Title> and its abstract is <Abstract>. This paper cited following papers: <TitleList> and abstracts of these papers are <AbstractList>. Here are possible categories: <Categories>. Which category does this paper belong to? You can only output one category.
	Yes	The title of one paper is <Title> and its abstract is <Abstract>. This paper is cited by following papers: <TitleList1>. Each of these papers belongs to one categories in: <Categories>. You need to 1.Analyse the papers' topic based on the give title and abstract; 2.Analyse the pattern of citation information based on on their titles, and retrieve the citation information you think is important to help you determine the category of the first given paper. Now you need to combine the information from 1 and 2 to predict the category of the first given paper. You should only output one category.
Few-shot	Yes	Here is a list of labeled papers: The title and abstract of the first paper are <Title1> and <Abstract1> respectively, and this pape r belongs to <Category1>$\cdots$ Here is a new paper whose title is <Title> and its abstract is <Abstract>. Here are possible categories: <Categories>. Which category does this paper belong to? You can only output one category.

Table 5: Examples of different prompts used in node classification experiments. For few-shot tasks, we randomly sampled two papers for each category due to the limit of input length.

papers in OGBN-ARXIV with lable of Computation and Language always are classified into other categories like Artificial Intelligence and Machine Learning. These prediction error papers always mentioned many machine learning concepts in their abstracts. We also argued weather datasets are 'out of fashion' compared with the information in the GPT's corpus. We prompted GPT to use pre-2000 categorization criteria on CORA, but this does not lead to improvements. Intriguingly, GPT-4 did not consistently surpass GPT-3.5 in terms of predictive accuracy, hinting that the extent of pre-trained knowledge could significantly influence predictions in zero-shot scenarios.

Incorporating structural information can slightly enhance the performance of GPT on node-level tasks to a certain degree. As evidenced in Table 2, the inclusion of neighborhood information enhances node classification performance in certain instances. However, this improvement lacks consistency across different LLM selections and datasets. Such observations could suggest that structural information might not be a pivotal factor in node-classification tasks. Additionally, we assessed the capability of GPT to retrieve information by incorporating retrieval requirements into the prompts for CORA. Regrettably, this led to both GPT-3.5 and GPT-4 struggling significantly, rendering them largely unable to provide accurate predictions.

3.2 Edge-level task

Contrary to node-level tasks, the structural information of a graph seems to be more crucial for edge-level tasks. When only node data, excluding neighborhood information, is presented to GPT, the link prediction accuracy on CORA stands at 51.2% for GPT-3.5 and 57.8% for GPT-4. Remarkably, these figures signicantly increased to 64.3% and 96.7% respectively when we incorporate the nodes' neighbors information. Notably, the zero-shot result of GPT-4 even surpass the performances of all trained GNN models. It is worth noting that some wrongly predicted links can be attributed to the absence of neighbor information for these nodes in the dataset. Table 6 illustrates the difference between prompts used on link prediction tasks. It is also interesting that when we further increase the information of neighboring nodes (e.g., the abstract of an article), the prediction accuracy becomes worse than only with information of titles. For the link prediction task on WORDNET18, we randomly selected 5 entries for each relationship and presented this information to GPT. Unfortunately, both GPT-3.5 and GPT-4 struggled to achieve a high predictive accuracy based on the provided data. A

Edge-level Task	Structure?	Prompt to GPT
Zero-shot (CORA & PUBMED)	No	There are two papers. Title of the first paper is <Title1> and its abstract is <Abstract1>. Title of the second paper is <Title2>, and its abstract is <Abstract2>. You need to predict whether the second paper or not. Answer 'YES' or 'NO'.
	Yes	There are two papers. Title of the first paper is <Title1> and its abstract is <Abstract1>. Title of the second paper is <Title2>, and its abstract is <Abstract2>. The first paper cited following papers: <Titles>. You need to predict whether the second paper or not. Answer 'YES' or 'NO'.
Few-shot (WORDNET18)	No	We define two descriptions should have a relationship, such as furniture <includes> bed. There are some samples: <Entries>. Here are possible relations: <Relationships>. The first entry is <Entry1> and the second entry is <Entry2>. You must output only one relationship between these two entries.

Table 6: Examples of different prompts used in link prediction experiments. Structural information plays an important role in link prediction task.

plausible explanation for this could be GPT's heavy reliance on its pre-trained knowledge, especially when not fine-tuned for specific tasks.

3.3 Graph-level task

For graph-level tasks, we only conducted experiments on REDDIT due to its text richness and semantic ambiguity. Only GPT-3.5 was tested since the information of one community is large even we have summarized the information from each user. We selected *top-k* post summaries of the most replied users as representative information of a community. As shown in Table 4, when GPT needs to make predictions from full 70 communities, the accuracy was 50.7%. The accuracy decreased from 77.3% to 50.7% when possible communities in the <SubReddits> list increased from 1 to 70.

Graph-level Task	Structure?	Prompt to GPT
Zero-shot (REDDIT)	Yes	There are texts from representative users of one Reddit community: <Posts>. There are following communities: <SubReddits>. Which community does these texts belong to? You should only output one community from given communities.

Table 7: Example prompt used in graph classification experiments. Structural information is given by a list of *top-k* important nodes a graph.

4 Conclusion and Future Works

This research presented a systematic empirical evaluation of leading LLMs on diverse graph learning tasks spanning node, edge, and graph levels. Through careful variation of prompt design and dataset selection, we assessed the capabilities of models such as GPT-3.5 and GPT-4 in interpreting and leveraging graph structural information to enhance predictive performance. Our results demonstrate that while LLMs exhibit reasonable node classification capabilities even without explicit graph data, likely by relying on contextual clues, their zero-shot performance continues to lag behind state-of-the-art GNNs specialized for this domain. However, incorporating graph topology information can significantly boost performance on edge-level link prediction tasks, with GPT-4 even surpassing certain GNNs in select cases. On more complex graph classification tasks, limitations emerge in handling increased output complexity. In summary, this research provides valuable evidence that LLMs have promising capabilities on graph analytics, while also revealing clear areas for improvement compared to specialized graph models.

Our future work should explore more rigorous benchmarking LLMs on graph learning tasks with graph specialized models, novel prompt designs to focus on topological structures, evaluating on additional graph tasks, and even fine-tuning open-sourced LLMs on graphs. By exploring these avenues, the full potential of large language models for advancing graph representation learning and analytics can be more promising.

References

[1] Seong Jin Ahn and MyoungHo Kim. Variational graph normalized autoencoders. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 2827–2831, 2021.

[2] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[3] Jayanth R Banavar, Amos Maritan, and Andrea Rinaldo. Size and form in efficient transportation networks. *Nature*, 399(6732):130–132, 1999.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

[5] Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, et al. Exploring the potential of large language models (llms) in learning on graphs. *arXiv preprint arXiv:2307.03393*, 2023.

[6] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *arXiv preprint arXiv:1606.09375*, 2016.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[8] Federico Errica, Marco Podda, Davide Bacciu, and Alessio Micheli. A fair comparison of graph neural networks for graph classification. *arXiv preprint arXiv:1912.09893*, 2019.

[9] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272. PMLR, 2017.

[10] Google AI. BARD: Deepmind's conversational ai assistant. `https://blog.google/products/search/introducing-bard-google-ai`, 2022. Accessed: October 1, 2023.

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

[12] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216*, 2017.

[13] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

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

[15] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[16] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[17] Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas. Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and its Applications*, 553:124289, 2020.

[18] Chen Ling, Junji Jiang, Junxiang Wang, My T Thai, Renhao Xue, James Song, Meikang Qiu, and Liang Zhao. Deep graph representation learning and optimization for influence maximization. In *International Conference on Machine Learning*, pages 21350–21361. PMLR, 2023.

[19] Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, et al. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *arXiv preprint arXiv:2305.18703*, 2023.

[20] Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 338–348, 2020.

[21] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[22] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*, 2021.

[23] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4602–4609, 2019.

[24] Mark EJ Newman, Duncan J Watts, and Steven H Strogatz. Random graph models of social networks. *Proceedings of the national academy of sciences*, 99(suppl_1):2566–2572, 2002.

[25] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023. PMLR, 2016.

[26] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

[27] Heiko Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508, 2017.

[28] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897, 2020.

[29] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[30] Prithviraj Sen, Galileo Mark Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[31] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023.

[32] Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta,

Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023.

[33] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[35] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[36] Lanning Wei, Zhiqiang He, Huan Zhao, and Quanming Yao. Unleashing the power of graph learning through llm-based autonomous agents. *arXiv preprint arXiv:2309.04565*, 2023.

[37] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[38] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[39] Hong Yang, Shirui Pan, Peng Zhang, Ling Chen, Defu Lian, and Chengqi Zhang. Binarized attributed network embedding. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1476–1481. IEEE, 2018.

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

[41] Zheng Zhang and Liang Zhao. Representation learning on spatial networks. *Advances in Neural Information Processing Systems*, 34:2303–2318, 2021.

[42] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[43] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804, 2020.