

873	<i>Supplement to</i>	
874		
875	“Query-Aware Subgraph Packing: A Knapsack Optimization	
876	Paradigm for Graph Retrieval-Augmented Generation”	
877	Appendix organization:	
878		
879	A Details of Method	23
880	B Related Work Extended	23
881	B.1 Classification-based GraphLLM	23
882	B.2 Graph Neural Networks	24
883	C Details of Experiments	24
884	C.1 Experimental Settings	24
885	C.2 Datasets	24
886	C.3 Baseline	26
887	D Further Details about Experiments	26
888	D.1 More Results for Overall Performance	26
889	D.2 Discussion of Retrieval Strategy	27
890	D.3 Performance in Low-resource Zero-shot Setting	27
891	D.4 Details of Ablation Study	27
892		

A Details of Method

In this section, we present a detailed description of the retrieval method used in GraphPack. We first introduce the selection of hyperparameters.

Anchor Node Selection. In node classification tasks, the goal is to predict the category or label of a single target node based on its structural and semantic context within the graph. Since the focus is on a single node, selecting only one anchor node ($\text{top-}k = 1$) is sufficient to capture the most relevant local subgraph around the target node. This approach ensures that the retrieved subgraph is compact and focused, reducing noise while maintaining high precision.

In graph question answering tasks, the query often involves reasoning about relationships between multiple nodes or entities. For example, determining whether two papers are co-cited or identifying the relationship between two knowledge entities requires considering multiple nodes simultaneously. By setting $\text{top-}k = 3$, we can retrieve a broader range of relevant nodes, which helps in capturing more diverse and interconnected information. This larger set of anchor nodes allows the model to explore richer structural patterns and improve its ability to answer complex queries that involve multi-hop reasoning.

Maximum Hop Distance for Local Subgraph. This parameter controls the maximum number of hops from each anchor node when extracting the local subgraph. A larger value increases coverage but may also introduce irrelevant nodes. For node classification, we set n -hop to 1, and for QA tasks, we set it to 4.

Knapsack Capacity \mathcal{C} . This parameter defines the capacity of the knapsack constraint used during optimization, limiting the total structural cost of selected nodes and edges. By default, we set \mathcal{C} to 20. The impact of varying knapsack capacities on performance is further explored in Section 4.3.

Max Score. This parameter determines how rewards are assigned to elements within each subgraph. We set it to 5 by default.

In the retrieval process, if the textual graph is a graph with both node and edge attributes, we treat nodes and edges as equally important elements. We assign rewards separately to the node set and the edge set. For each retrieved edge, we include both of its endpoint nodes in the subgraph node set. Subsequently, we incorporate all edges that involve any of the candidate nodes to form the final subgraph.

B Related Work Extended

B.1 Classification-based GraphLLM

Also referred to as "LLMs as Prefix" or "LLMs as Enhancer", is a common approach that leverages prompting techniques to query LLMs for generating semantically rich auxiliary information—such as explanations, knowledge entities, and pseudo-labels—to enhance the training of GNN. For instance, TAPE [He et al., 2024a] employs customized prompts to query LLMs and generate predictions along with textual explanations for each node. OpenGraph [Xia et al., 2024] utilizes LLMs to synthesize nodes and edges, thereby alleviating the issue of sparse training data. LLM-GNN [Chen et al., 2024c] treats LLMs as annotators that produce node class predictions with confidence scores, which are then used to train GNN. LLMRec [Wei et al., 2024] enhances user-item interaction edges using LLMs, addressing the challenges of data sparsity and quality in graph-based recommendation systems.

An alternative line of work directly uses the text embeddings generated by LLMs as initial node representations for GNN training. GALM [Xie et al., 2023] encodes node textual features using pre-trained language models and further pretrains them through unsupervised learning tasks such as link prediction. G-Prompt [Huang et al., 2023b] introduces a graph adapter at the end of a pre-trained language model to extract graph-aware node features. SimTeG [Duan et al., 2023] first applies parameter-efficient fine-tuning on the text embeddings obtained from LLMs for downstream tasks, and subsequently feeds the node embeddings into a GNN for inference. WalkLM [Tan et al., 2023] fine-tunes LLMs on textual sequences formed via random walks and extracts representations from

the LLMs. OFA [Liu et al., 2024a] harnesses the powerful capabilities of LLMs to unify the encoding of graph data from diverse sources, facilitating cross-domain learning.

Despite the strong performance of classification-based GraphLLMs on TAG-style classification tasks, their inherent reliance on classification limits their applicability to more general knowledge-intensive graph tasks, particularly in question-answering scenarios. Moreover, these methods incur substantial computational costs when applied to large-scale datasets. Taking explanation-based approaches as an example, for a graph with N nodes, they require querying the LLM API N times.

B.2 Graph Neural Networks

GNNs have established themselves as a powerful framework for modeling and analyzing graph-structured data. [Kipf and Welling, 2016, Veličković et al., 2018, Hamilton et al., 2017, Yang et al., 2023] employ robust message passing and aggregation mechanisms to process graph data, achieving exemplary performance in tasks such as graph classification. Many efforts have also been made to adapt GNNs to the Knowledge Graph mining domain and achieved great success due to GNN’s efficiency and inductive learning capabilities [Zhu et al., 2022, Kong et al., 2023]. Despite these strengths, GNN face several limitations. They typically require task-specific heads or fine-tuning for downstream applications, which complicates multi-task scenarios and may lead to poor generalization [Ju et al., 2023]. Additionally, while GNN excel in classification tasks, the focus of current research is shifting towards addressing more complex user requirements [Yih et al., 2016, He et al., 2024b]. These advanced tasks not only demand higher expressive power from models but also necessitate effective sharing and integration of information across multiple tasks. However, existing GNN designs often fall short in handling these complexities, thereby limiting their widespread adoption in real-world applications.

C Details of Experiments

In this section, we provide detailed settings for all experiments shown in the paper. The code for GraphPack can be found in the compressed file within the supplement. All experiments were fine-tuned using the llama-2-7b model.

C.1 Experimental Settings

We test the impact of GraphPack on the performance of LLMs across various scenarios. Specifically, we first evaluate GraphPack in the **Supervised Fine-Tuning** (SFT) setting. This ensures that our query-aware graph encoder can be fully preserved. For all datasets, we set the number of layers in the graph encoder to 4 and the hidden dimension to 1024, aligning it with the text encoder. The pooled graph embedding dimension is set to 4096, consistent with the dimensionality of the base model LLaMA2 that we use. Figure 13 illustrates the prompt design for SFT.

To evaluate the generalization capability of our framework under **zero-shot** settings, we design two types of cross-domain and cross-task transfer experiments:

Cross-Domain Generalization. In this setting, we train the model on one type of graph (e.g., citation graphs such as Cora) and directly evaluate it on another type of graph (e.g., social networks such as Instagram), without any fine-tuning on the target domain. This setup assesses whether the model can generalize across different graph modalities and structures.

Cross-Task Generalization. Here, the model is trained on one task formulation (e.g., QA-style prompts over knowledge graphs like CWQ) and tested on structurally similar graphs but with different user intents or textual templates (e.g., category prediction on WikiCS). This evaluates the model’s ability to adapt to new tasks by leveraging the structural and semantic knowledge learned during retrieval and modulation. Figure 14 illustrates the prompt design for zero-shot settings.

C.2 Datasets

Cora. The Cora dataset is a citation network consisting of research papers in the field of computer science and their citation relationships. In this dataset, each node represents a research paper, and the

Table 6: Statistics and training settings (first).

Dataset	Cora	Citeseer	WikiCS	Instagram	Ogbn-arxiv
Domain	Citation	Citation	Web link	Social	Citation
Graphs	1	1	1	1	1
Avg. #nodes	2,708	3,186	11,701	11,339	169,343
Avg. #edges	5,429	4,277	216,123	144,010	1,166,243
Metric	Accuracy,F1	Accuracy,F1	Accuracy,F1	Accuracy,F1	Accuracy,F1
Epoch	20	20	10	10	3
Learning rate	1e-4	1e-4	1e-4	1e-4	1e-4

Table 7: Statistics and training settings (second).

Dataset	WebQSP	CWQ
Graphs	4,737	34,689
Avg. #nodes	1,370	1,255
Avg. #edges	4,252	4,001
Metric	F1,Hit@1,Recall	F1,Hit@1,Recall
Epoch	20	10
Learning rate	1e-5	1e-5

Table 8: Statistics and training settings (third).

Dataset	MultihopQA	MusiqueQA
# of Tokens	1,434,889	3,280,174
# of Questions	2,556	3,000
Metric	Accuracy,Recall	Accuracy,Recall
Epoch	5	5
Learning rate	1e-5	1e-5

original text features associated with each node include the title and abstract of the corresponding paper. An edge in the Cora dataset denotes a citation relationship between two papers. The label assigned to each node corresponds to the paper’s category. The original text data of the Cora dataset is sourced from the GitHub repository provided by [Chen et al., 2024b] and we use the same split as [Li et al., 2024].

Citeseer. The Citeseer dataset is a citation network that contains research papers and their citation relationships within the field of computer science. Each node represents a research paper, and each edge denotes a citation relationship between two papers. The original text data of the Cora dataset is sourced from the GitHub repository provided by [Chen et al., 2024b]. We use the same split as [Li et al., 2024].

WikiCS. The WikiCS dataset is a internet network in which each node represents a Wikipedia page, and each edge corresponds to a citation link between pages. The raw text associated with each node includes the name and content of the Wikipedia entry. The label assigned to each node corresponds to the category of the entry. We use the same split as [Li et al., 2024] and the raw text data for the WikiCS dataset was collected from [Kong et al., 2025].

Instagram. The Instagram dataset is a social network in which nodes represent users and edges represent follow relationships. The raw text associated with each node includes the personal introduction of this user. Each node is labeled to indicate whether the user is commercial or normal. The raw text data for the Instagram dataset was collected from [Huang et al., 2024]. We keep the same split as [Li et al., 2024].

Ogbn-arxiv. The Ogbn-Arxiv dataset is a citation network consisting of research papers collected from the Arxiv platform and their citation relationships. Each node represents a research paper, and each edge denotes a citation relationship between two papers. The raw text data of the ogbn-arxiv was collected using the same protocol as the GitHub repository provided in [Huang et al., 2023a]. The raw text data for the Ogbn-arxiv dataset was sourced from the GitHub repository provided in [Kong et al., 2025]. We keep the same split as [Li et al., 2024].

WebQSP. The WebQSP dataset is a large-scale, multi-hop knowledge graph question answering dataset containing 4,737 questions. This dataset, proposed by [Yih et al., 2016], builds upon the work of [Luo et al., 2024], and uses a subset of Freebase that includes facts within a two-hop neighborhood of the entities mentioned in the questions. The task involves answering questions that require multi-hop reasoning.

CWQ. The CWQ dataset is a new dataset for complex question answering, built upon the WebQSP dataset. [Talmor and Berant, 2018] extracted SPARQL queries from WebQSP and automatically generated more complex query patterns. Natural language questions were then created using Amazon Mechanical Turk (AMT), resulting in a dataset of 34,689 question-answer pairs that require up to four-hop reasoning.

MultihopQA. MultihopQ is a dataset composed of a knowledge base, a large number of multi-hop queries, their ground-truth answers, and associated supporting evidence. [Tang and Yang, 2024] constructed the dataset by leveraging an English news article corpus as the underlying RAG knowledge base. We use the code from [Zhou et al., 2025] to build the graph structure based on entities and triples, in order to form the final graph question-answering format.

MusiqueQA. The MusiqueQA dataset [Trivedi et al., 2022] constructs multi-hop questions by composing single-hop questions, effectively forming complex multi-hop questions as combinations of simpler ones. This composition-based approach allows for better control over the quality and structure of the generated multi-hop questions. MusiqueQA contains 25K questions with 2 to 4 hops, built from seed questions sourced from five existing single-hop datasets. Similarly, we use the code from [Zhou et al., 2025] to build a graph structure based on entities and triples, in order to form the final graph question-answering format.

C.3 Baseline

In our comparative analysis, we benchmark our framework against two categories of state-of-the-art models to ensure a comprehensive evaluation.

First, we compare our approach with GraphLLMs that demonstrate strong performance on specialized tasks. These include the powerful graph benchmarking model **OFA** [Kong et al., 2025], **InstructGLM** [Ye et al., 2024] and **GraphText** [Zhao et al., 2023] — which utilize natural language descriptions of textual graphs — as well as **GraphAdapter** [Huang et al., 2024] and **LLaGA** [Chen et al., 2024a], which are equipped with dedicated graph encoders.

Second, we compare our approach with graph retrieval-augmented generation methods that have shown strong performance on GraphQA tasks. This comparison covers methods such as **G-Retriever** [He et al., 2024b], which employs the Prize-Collecting Steiner Tree (PCST) algorithm for sub-graph retrieval, and **GRAG** [Hu et al., 2025], which introduces a graph soft-pruning mechanism to dynamically filter out irrelevant nodes and edges during reasoning.

D Further Details about Experiments

D.1 More Results for Overall Performance

We also conduct experiments on multi-hop question answering (QA) datasets. Originally organized as text-based QA pairs, these datasets are transformed into graph-structured representations to better align with our GraphRAG framework. This transformation enables the model to leverage graph reasoning capabilities to capture complex multi-hop paths, thereby better supporting QA tasks.

From the experimental results in Table 9, we observe that graph-based RAG methods generally achieve higher accuracy than traditional Vanilla RAG approaches that rely solely on textual passage retrieval. This indicates that, for tasks requiring multi-hop reasoning, graph structures can more effectively model complex relationships between entities, thereby improving both retrieval quality and generation performance.

Table 9: Statistics and training settings (third).

Model	MultihopQA		MusiqueQA	
	Accuracy	Recall	Accuracy	Recall
Llama-2-7B	51.56	36.38	5.67	9.23
VanillaRAG	53.89	38.29	18.98	27.98
G-Retriever	44.17	45.41	8.56	13.27
GRAG	44.23	46.10	8.61	13.51
GraphPack	45.91	47.20	10.75	16.93

Furthermore, we observe that GraphPack consistently outperforms both G-Retriever and GRAG across multiple QA datasets. We attribute this advantage to GraphPack’s subgraph retrieval mechanism. Compared to other methods, GraphPack achieves a larger local receptive field, enabling it to retrieve longer reasoning paths within the graph. This capability is particularly crucial for solving problems involving indirect associations among multiple entities.

D.2 Discussion of Retrieval Strategy

We compared our approach with traditional n-hop subgraphs on node classification datasets. Specifically, we replaced the subgraphs in GraphPack with 1-hop and 2-hop subgraphs while keeping other components unchanged, as presented in the table [10](#). Simply substituting the subgraphs in GraphPack with n-hop subgraphs resulted in a decline in performance. We attribute this performance degradation to the fact that n-hop subgraphs do not incorporate semantic information, leading to excessive noise. Furthermore, in larger text graphs such as Wikics, 2-hop subgraphs can scale up to 5,000 nodes. Therefore, we believe that GraphPack’s subgraph retrieval strategy effectively balances performance while controlling computational costs.

Table 10: The impact of different subgraph strategies.

Model	Cora		Citeseer	
	Acc	F1	Acc	F1
1-Hop Subgraph	<u>74.98</u>	74.33	68.23	<u>66.54</u>
2-Hop Subgraph	74.78	<u>74.42</u>	<u>68.38</u>	66.13
GraphPack	76.40	75.45	69.95	67.59

D.3 Performance in Low-resource Zero-shot Setting

GraphPack exhibits robust zero-shot performance, prompting us to conduct a deeper investigation into its adaptability in low-resource scenarios. We focus on how well the framework generalizes to unseen domains and tasks, aiming to uncover the extent to which its structural and semantic knowledge can be transferred effectively under limited supervision.

Table 11: Performance in Low-resource Setting.

Model	WebQSP		CWQ	
	F1	Hit@1	F1	Hit@1
Llama-2-7B	25.91	42.99	20.01	24.32
Mistral-7B	<u>26.32</u>	<u>43.65</u>	<u>21.55</u>	<u>25.39</u>
GraphPack	32.67	48.14	28.78	30.52

Specifically, we performed 5-shot training on a citation graph node classification task (i.e., 5 labeled samples per class), and tested the model on a graph question-answering dataset. This setting assesses GraphPack’s ability to quickly adapt to a new domain under limited training data conditions. As shown in Table [11](#), GraphPack outperforms fine-tuned Vanilla LLMs across all metrics on the graph qa benchmark, demonstrating stronger cross-task generalization and data efficiency. These results validate GraphPack’s capability to bootstrap effective structural learning even under sparse supervision.

D.4 Details of Ablation Study

In the ablation study, we systematically evaluate the contribution of each key component in GraphPack by removing them individually and measuring the resulting performance drop. As shown in Table [12](#), all components — including node retrieval, edge retrieval, query-aware linear modulation (Query-LM), and graph-to-text reconstruction (GTR) in the knapsack optimization — play essential roles in the overall effectiveness of the framework. Specifically, the absence of edge retrieval leads to the worst performance degradation, which we attribute to the more prominent role of edges in capturing local structural information. Moreover, Query-LM brings significant performance improvements, indicating that integrating query features into the message-passing process of the graph encoder helps it identify more relevant local structures

Table 12: Performance in Low-resource Setting.

Model	WebQSP		CWQ	
	F1	Hit@1	F1	Hit@1
GraphPack	51.79	73.01	41.03	48.50
w/o Nodes Retrieval	49.73	71.28	39.62	47.02
w/o Edges Retrieval	48.97	69.55	39.21	46.54
w/o Query-LM	50.25	71.30	39.81	47.28
w/o GTR	51.48	72.56	40.40	47.97

Table 13: Prompts for tasks in the supervised fine-tuning setting.

Dataset	Prompt
Cora	This is a co-citation network focusing on artificial intelligence, nodes represent academic papers and edges represent two papers are co-cited by other papers. Here is a paper content: <paper content>. What is the most likely paper category for the target paper?
Citeseer	This is a citation network that contains research papers and their citation relationships within the field of computer science. Each node represents a research paper, and each edge denotes a citation relationship between two papers. Here is a paper content: <paper content>. What is the most likely paper category for the target paper?
WikiCS	This is a Wikipedia graph focusing on computer science. Nodes represent Wikipedia terms and edges represent two terms have hyperlink. Here is a Computer Science article: <entity content>. What is the most likely category for this Wikipedia term?
Instagram	This is a social network in which nodes represent users and edges represent follow relationships. The raw text associated with each node includes the personal introduction of this user. Here is a introduction of the user: <user content>. Which category does the user belong to?
Ogbn-arxiv	This is a citation network that contains research papers and their citation relationships within the field of computer science. Each node represents a research paper, and each edge denotes a citation relationship between two papers. Here is a paper content: <paper content>. What is the most likely paper category for the target paper?
WebQSP	You are given a knowledge graph in the form of a structured textual description and a user question. Your task is to use the information from the graph to answer the question accurately. Graph description: <graph description> Question: <user query> Based on the provided graph description, please answer the user’s question directly and concisely.
CWQ	You are given a knowledge graph in the form of a structured textual description and a user question. Your task is to use the information from the graph to answer the question accurately. Graph description: <graph description> Question: <user query> Based on the provided graph description, please answer the user’s question directly and concisely.

1120 based on the query. These findings confirm that the components of GraphPack are complementary and
1121 collectively contribute to its strong performance in knowledge-intensive, multi-hop graph reasoning
1122 tasks.

Table 14: Prompts for tasks in the zero-shot setting.

Dataset	Prompt
Cora	You are an expert in the field of graphs, equipped with strong structural analysis capabilities. This is a co-citation network focusing on artificial intelligence, nodes represent academic papers and edges represent two papers are co-cited by other papers. Here is a paper content: <paper content>. What is the most likely paper category for the target paper? Choose the most probable answer from the provided list: {<label>}. Please directly answer the category.
WikiCS	You are an expert in the field of graphs, equipped with strong structural analysis capabilities. This is a Wikipedia graph focusing on computer science. Nodes represent Wikipedia terms and edges represent two terms have hyperlink. Here is a Computer Science article: <entity content>. What is the most likely category for this Wikipedia term? Choose the most probable answer from the provided list: {<label>}. Please directly answer the category.
Instagram	You are an expert in the field of graphs, equipped with strong structural analysis capabilities. This is a social network in which nodes represent users and edges represent follow relationships. The raw text associated with each node includes the personal introduction of this user. Here is a introduction of the user: <user content>. Choose the most probable answer from the provided list: {<label>}. Which category does the user belong to? Please directly answer the category.
WebQSP	You are an expert in the field of graphs, equipped with strong structural analysis capabilities. You are given a knowledge graph in the form of a structured textual description and a user question. Your task is to use the information from the graph to answer the question accurately. Graph description: <graph description> Question: <user query> Based on the provided graph description, please answer the user’s question directly and concisely.
CWQ	You are an expert in the field of graphs, equipped with strong structural analysis capabilities. You are given a knowledge graph in the form of a structured textual description and a user question. Your task is to use the information from the graph to answer the question accurately. Graph description: <graph description> Question: <user query> Based on the provided graph description, please answer the user’s question directly and concisely.

References

- Runjin Chen, Tong Zhao, Ajay Kumar Jaiswal, Neil Shah, and Zhangyang Wang. LLaGA: Large language and graph assistant. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 7809–7823. PMLR, 21–27 Jul 2024a. URL <https://proceedings.mlr.press/v235/chen24bh.html>.
- Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang, Dawei Yin, Wenqi Fan, Hui Liu, and Jiliang Tang. Exploring the potential of large language models (llms) in learning on graphs, 2024b. URL <https://arxiv.org/abs/2307.03393>.
- Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. Label-free node classification on graphs with large language models (llms), 2024c. URL <https://arxiv.org/abs/2310.04668>.
- Keyu Duan, Qian Liu, Tat-Seng Chua, Shuicheng Yan, Wei Tsang Ooi, Qizhe Xie, and Junxian He. Simteg: A frustratingly simple approach improves textual graph learning, 2023. URL <https://arxiv.org/abs/2308.02565>.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitan, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2025. URL <https://arxiv.org/abs/2404.16130>.
- Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models, 2023. URL <https://arxiv.org/abs/2310.04560>.
- Arnaud Freville. The multidimensional 0-1 knapsack problem: An overview. *European Journal of Operational Research*, 155(1):1–21, May 2004. URL <https://ideas.repec.org/a/eee/ejores/v155y2004i1p1-21.html>.
- Bin Fu, Yunqi Qiu, Chengguang Tang, Yang Li, Haiyang Yu, and Jian Sun. A survey on complex question answering over knowledge base: Recent advances and challenges, 2020. URL <https://arxiv.org/abs/2007.13069>.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training, 2020. URL <https://arxiv.org/abs/2002.08909>.
- William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, 2017.
- Zhen Han, Yue Feng, and Mingming Sun. A graph-guided reasoning approach for open-ended commonsense question answering, 2023. URL <https://arxiv.org/abs/2303.10395>.
- Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation, 2020. URL <https://arxiv.org/abs/2002.02126>.
- Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning, 2024a. URL <https://arxiv.org/abs/2305.19523>.
- Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering, 2024b. URL <https://arxiv.org/abs/2402.07630>.
- Yufei He, Yuan Sui, Xiaoxin He, and Bryan Hooi. Unigraph: Learning a unified cross-domain foundation model for text-attributed graphs, 2025. URL <https://arxiv.org/abs/2402.13630>.

371 Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. GRAG: Graph
372 retrieval-augmented generation. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings*
373 *of the Association for Computational Linguistics: NAACL 2025*, pages 4145–4157, Albuquerque,
374 New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7.
375 URL <https://aclanthology.org/2025.findings-naacl.232/>.

376 Qian Huang, Hongyu Ren, Peng Chen, Gregor Kržmanc, Daniel Zeng, Percy Liang, and Jure
377 Leskovec. Prodigy: Enabling in-context learning over graphs, 2023a. URL <https://arxiv.org/abs/2305.12600>.

378 Xuanwen Huang, Kaiqiao Han, Dezheng Bao, Quanjin Tao, Zhisheng Zhang, Yang Yang, and Qi Zhu.
379 Prompt-based node feature extractor for few-shot learning on text-attributed graphs, 2023b. URL
380 <https://arxiv.org/abs/2309.02848>.

381 Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu.
382 Can gnn be good adapter for llms? WWW, 2024.

383 Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane
384 Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with
385 retrieval augmented language models, 2022. URL <https://arxiv.org/abs/2208.03299>.

386 Mingxuan Ju, Tong Zhao, Qianlong Wen, Wenhao Yu, Neil Shah, Yanfang Ye, and Chuxu Zhang.
387 Multi-task self-supervised graph neural networks enable stronger task generalization, 2023. URL
388 <https://arxiv.org/abs/2210.02016>.

389 Thomas Kipf and Max Welling. Semi-supervised classification with graph convolutional networks.
390 *arXiv: Learning, arXiv: Learning*, Sep 2016.

391 Lecheng Kong, Yixin Chen, and Muhan Zhang. Geodesic graph neural network for efficient graph
392 representation learning, 2023. URL <https://arxiv.org/abs/2210.02636>.

393 Lecheng Kong, Jiarui Feng, Hao Liu, Chengsong Huang, Jiaxin Huang, Yixin Chen, and Muhan
394 Zhang. Gofa: A generative one-for-all model for joint graph language modeling, 2025. URL
395 <https://arxiv.org/abs/2407.09709>.

396 Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. A survey on
397 complex knowledge base question answering: Methods, challenges and solutions. In Zhi-Hua
398 Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence,*
399 *IJCAI-21*, pages 4483–4491. International Joint Conferences on Artificial Intelligence Organization,
400 8 2021. doi: 10.24963/ijcai.2021/611. URL <https://doi.org/10.24963/ijcai.2021/611>

401 Survey Track.

402 Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman
403 Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel,
404 and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In
405 H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in*
406 *Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates,
407 Inc., 2020. URL [https://proceedings.neurips.cc/paper_files/paper/2020/file/](https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf)
408 [6b493230205f780e1bc26945df7481e5-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf).

409 Yuhan Li, Peisong Wang, Xiao Zhu, Aochuan Chen, Haiyun Jiang, Deng Cai, Victor Wai Kin Chan,
410 and Jia Li. Glbench: A comprehensive benchmark for graph with large language models, 2024.
411 URL <https://arxiv.org/abs/2407.07457>.

412 Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan
413 Zhang. One for all: Towards training one graph model for all classification tasks, 2024a. URL
414 <https://arxiv.org/abs/2310.00149>.

415 Haochen Liu, Song Wang, Yaochen Zhu, Yushun Dong, and Jundong Li. Knowledge graph-enhanced
416 large language models via path selection. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar,
417 editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6311–6321,
418 Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/
419 2024.findings-acl.376. URL <https://aclanthology.org/2024.findings-acl.376/>.

420

421 Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike
422 Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining
423 approach, 2019. URL <https://arxiv.org/abs/1907.11692>.

424 Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. Reasoning on graphs: Faithful
425 and interpretable large language model reasoning, 2024. URL <https://arxiv.org/abs/2310.01061>.

427 Qiyao Ma, Xubin Ren, and Chao Huang. XRec: Large language models for explainable recommenda-
428 tion. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association
429 for Computational Linguistics: EMNLP 2024*, pages 391–402, Miami, Florida, USA, November
430 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.22. URL
431 <https://aclanthology.org/2024.findings-emnlp.22/>.

432 Costas Mavromatis and George Karypis. Gnn-rag: Graph neural retrieval for large language model
433 reasoning, 2024. URL <https://arxiv.org/abs/2405.20139>.

434 Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang
435 Tang. Graph retrieval-augmented generation: A survey, 2024. URL <https://arxiv.org/abs/2408.08921>.

437 Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual rea-
438 soning with a general conditioning layer, 2017. URL <https://arxiv.org/abs/1709.07871>.

439 Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal,
440 Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever.
441 Learning transferable visual models from natural language supervision, 2021. URL <https://arxiv.org/abs/2103.00020>.

443 Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and
444 Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association
445 for Computational Linguistics*, 11:1316–1331, 2023. doi: 10.1162/tac1_a_00605. URL <https://aclanthology.org/2023.tac1-1.75/>.

447 Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-
448 networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings
449 of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th
450 International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–
451 3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi:
452 10.18653/v1/D19-1410. URL <https://aclanthology.org/D19-1410/>.

453 Xubin Ren, Jiabin Tang, Dawei Yin, Nitesh Chawla, and Chao Huang. A survey of large language
454 models for graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery
455 and Data Mining, KDD ’24*, page 6616–6626. ACM, August 2024. doi: 10.1145/3637528.3671460.
456 URL <http://dx.doi.org/10.1145/3637528.3671460>.

457 Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked
458 label prediction: Unified message passing model for semi-supervised classification, 2021. URL
459 <https://arxiv.org/abs/2009.03509>.

460 Alon Talmor and Jonathan Berant. The web as a knowledge-base for answering complex ques-
461 tions. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Confer-
462 ence of the North American Chapter of the Association for Computational Linguistics: Human
463 Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana,
464 June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1059. URL
465 <https://aclanthology.org/N18-1059/>.

466 Yanchao Tan, Zihao Zhou, Hang Lv, Weiming Liu, and Carl Yang. Walklm: A uniform lan-
467 guage model fine-tuning framework for attributed graph embedding. In A. Oh, T. Nau-
468 mann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural
469 Information Processing Systems*, volume 36, pages 13308–13325. Curran Associates,
470 Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/2ac879d1865475a7abc8dfc7a9c15c27-Paper-Conference.pdf.

472 Yanchao Tan, Hang Lv, Xinyi Huang, Jiawei Zhang, Shiping Wang, and Carl Yang. Musegraph:
 473 Graph-oriented instruction tuning of large language models for generic graph mining, 2024. URL <https://arxiv.org/abs/2403.04780>.
 474

475 Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang.
 476 Graphgpt: Graph instruction tuning for large language models, 2024a. URL [https://arxiv](https://arxiv.org/abs/2310.13023)
 477 [org/abs/2310.13023](https://arxiv.org/abs/2310.13023).

478 Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. Higt:
 479 Heterogeneous graph language model, 2024b. URL <https://arxiv.org/abs/2402.16024>.

480 Yixuan Tang and Yi Yang. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop
 481 queries, 2024. URL <https://arxiv.org/abs/2401.15391>.

482 Yijun Tian, Huan Song, Zichen Wang, Haozhu Wang, Ziqing Hu, Fang Wang, Nitesh V. Chawla, and
 483 Panpan Xu. Graph neural prompting with large language models. In *Proceedings of the Thirty-*
 484 *Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative*
 485 *Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in*
 486 *Artificial Intelligence*, AAAI’24/IAAI’24/EAAI’24. AAAI Press, 2024. ISBN 978-1-57735-887-9.
 487 doi: 10.1609/aaai.v38i17.29875. URL <https://doi.org/10.1609/aaai.v38i17.29875>.

488 Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop
 489 questions via single-hop question composition. *Transactions of the Association for Computational*
 490 *Linguistics*, 10:539–554, 2022. doi: 10.1162/tac1_a_00475. URL [https://aclanthology.org/](https://aclanthology.org/2022.tac1-1.31/)
 491 [2022.tac1-1.31/](https://aclanthology.org/2022.tac1-1.31/).

492 Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua
 493 Bengio. Graph attention networks, 2018. URL <https://arxiv.org/abs/1710.10903>.

494 Jianing Wang, Junda Wu, Yupeng Hou, Yao Liu, Ming Gao, and Julian McAuley. Instructgraph:
 495 Boosting large language models via graph-centric instruction tuning and preference alignment,
 496 2024. URL <https://arxiv.org/abs/2402.08785>.

497 Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin,
 498 and Chao Huang. Llmrec: Large language models with graph augmentation for recommendation,
 499 2024. URL <https://arxiv.org/abs/2311.00423>.

500 Lianghao Xia, Ben Kao, and Chao Huang. OpenGraph: Towards open graph foundation models. In
 501 Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for*
 502 *Computational Linguistics: EMNLP 2024*, pages 2365–2379, Miami, Florida, USA, November
 503 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.132.
 504 URL <https://aclanthology.org/2024.findings-emnlp.132/>.

505 Han Xie, Da Zheng, Jun Ma, Houyu Zhang, Vassilis N. Ioannidis, Xiang Song, Qing Ping, Sheng
 506 Wang, Carl Yang, Yi Xu, Belinda Zeng, and Trishul Chilimbi. Graph-aware language model
 507 pre-training on a large graph corpus can help multiple graph applications, 2023. URL <https://arxiv.org/abs/2306.02592>.
 508

509 Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Defu Lian, Sanjay Agrawal, Amit Singh,
 510 Guangzhong Sun, and Xing Xie. Graphformers: Gnn-nested transformers for representation
 511 learning on textual graph, 2023. URL <https://arxiv.org/abs/2105.02605>.

512 Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. Qa-gnn:
 513 Reasoning with language models and knowledge graphs for question answering, 2022. URL
 514 <https://arxiv.org/abs/2104.06378>.

515 Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. Language is all a graph
 516 needs, 2024. URL <https://arxiv.org/abs/2308.07134>.

517 Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. The value of
 518 semantic parse labeling for knowledge base question answering. In Katrin Erk and Noah A.
 519 Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational*
 520 *Linguistics (Volume 2: Short Papers)*, pages 201–206, Berlin, Germany, August 2016. Association
 521 for Computational Linguistics. doi: 10.18653/v1/P16-2033. URL [https://aclanthology](https://aclanthology.org/P16-2033/)
 522 [org/P16-2033/](https://aclanthology.org/P16-2033/).

- 523 Jing Zhang, Xiaokang Zhang, Jifan Yu, Jian Tang, Jie Tang, Cuiping Li, and Hong Chen. Subgraph
524 retrieval enhanced model for multi-hop knowledge base question answering. In Smaranda Muresan,
525 Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the*
526 *Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5773–5784, Dublin,
527 Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.
528 396. URL <https://aclanthology.org/2022.acl-long.396/>.
- 529 Mengmei Zhang, Mingwei Sun, Peng Wang, Shen Fan, Yanhu Mo, Xiaoxiao Xu, Hong Liu, Cheng
530 Yang, and Chuan Shi. Graphtranslator: Aligning graph model to large language model for
531 open-ended tasks, 2024. URL <https://arxiv.org/abs/2402.07197>.
- 532 Jianan Zhao, Le Zhuo, Yikang Shen, Meng Qu, Kai Liu, Michael Bronstein, Zhaocheng Zhu, and
533 Jian Tang. Graphtext: Graph reasoning in text space, 2023. URL [https://arxiv.org/abs/](https://arxiv.org/abs/2310.01089)
534 [2310.01089](https://arxiv.org/abs/2310.01089).
- 535 Yingli Zhou, Yaodong Su, Youran Sun, Shu Wang, Taotao Wang, Runyuan He, Yongwei Zhang,
536 Sicong Liang, Xilin Liu, Yuchi Ma, and Yixiang Fang. In-depth analysis of graph-based rag in a
537 unified framework, 2025. URL <https://arxiv.org/abs/2503.04338>.
- 538 Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford
539 networks: A general graph neural network framework for link prediction, 2022. URL [https:](https://arxiv.org/abs/2106.06935)
540 [//arxiv.org/abs/2106.06935](https://arxiv.org/abs/2106.06935).