Mixture of Demonstrations for Textual Graph Understanding and Question Answering

Yukun Wu, Lihui Liu
yukun.wu.mail@gmail.com,hw6926@wayne.edu
Independent Researcher, Wayne State University
Detroit, USA

Abstract

Textual graph-based retrieval-augmented generation (GraphRAG) has emerged as a powerful paradigm for enhancing large language models (LLMs) in domain-specific question answering. While existing approaches primarily focus on zero-shot GraphRAG, selecting high-quality demonstrations is crucial for improving reasoning and answer accuracy. Furthermore, recent studies have shown that retrieved subgraphs often contain irrelevant information, which can degrade reasoning performance. In this paper, we propose MixDemo, a novel GraphRAG framework enhanced with a Mixture-of-Experts (MoE) mechanism for selecting the most informative demonstrations under diverse question contexts. To further reduce noise in the retrieved subgraphs, we introduce a query-specific graph encoder that selectively attends to information most relevant to the query. Extensive experiments across multiple textual graph benchmarks show that MixDemo significantly outperforms existing methods.

CCS Concepts

• **Computing methodologies** → **Reasoning about belief and knowledge**; • **Information systems** → **Data mining**.

Keywords

Knowledge graph question answering

ACM Reference Format:
Yukun Wu, Lihui Liu. 2023. Mixture of Demonstrations for Textual Graph Understanding and Question Answering. In *Proceedings of the ACM Web Conference 2023 (WWW '23), May 1–5, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3543507.3583316

1 Introduction

Large language models (LLMs) have achieved remarkable success in recent years. Yet, most LLMs are trained on open-domain data before fixed cut-off dates [41], which inevitably limits their performance when facing domain-specific questions due to outdated or missing knowledge. To address this issue, Retrieval-Augmented Generation (RAG) [4, 43] has emerged as a promising solution, where relevant knowledge is retrieved and incorporated to help LLMs generate accurate responses. Most existing GraphRAG methods [4, 6] rely on a simple yet strong assumption, that the generated response should include all relevant facts retrieved from the graph. However, this assumption overlooks a critical problem: the retrieved textual graph might contain unnecessary noise or irrelevant information. Consider Figure 1(a), where a user asks "What is a good source of nutrients for a mushroom?" While the textual graph contains correct 'a cut peony', it also contains useless information 'a flying eagle' which may mislead the model. How to mitigate low-quality information from the retrieved textual graph to effectively guide answer generation is a problem. Furthermore, existing GraphRAG methods mainly utilize zero shot learning. Selecting high-quality demonstrations is crucial for improving reasoning and answer accuracy.

In this paper, we propose a GraphRAG framework that focuses on enhancing both textual graph understanding and QA performance by learning to select high-quality demonstrations and query specific information learning. Instead of assuming all retrieved subgraph content is useful, our approach adaptively identifies the most informative and context-relevant node-level and edge-level evidence. Specifically, we design a specialized, query-specific GraphEncoder to model the complex interactions between nodes within the retrieved subgraphs. This encoder generates a dense graph prompt embedding that captures relational patterns and serves as a bridge between structured knowledge and the LLM's input space. Additionally, we incorporate a Mixture-of-Experts (MoE) [3] paradigm to select the most informative demonstrations to enhance in-context learning. We evaluate our method on the GraphQA benchmark. Extensive experiments show that our model significantly outperforms existing baselines.

2 Problem Definition

We study the task of answering queries over a *textual graph* using large language models (LLMs). A textual graph $G = (V, E)$ contains natural-language content on both nodes and edges. Given a query q, the goal is to generate an answer a_{gen} by retrieving a relevant subgraph $S \subseteq G$ and using it to guide LLM-based reasoning.

We adopt *in-context learning* (ICL), where a set of demonstration examples $\mathcal{D} = \{(q_i, a_i)\}$ is prepended to the query to prompt the LLM. At test time, a subset $\text{Sub}(D)$ is selected from $\mathcal{D}$ to construct the prompt. The LLM then predicts:

$$\hat{a} = \arg\max_{a} P_{\text{LM}}(a \mid \text{Sub}(D), q).$$

Final Problem Definition. Our goal is to answer a query q on a textual graph G by: (1) retrieving a relevant subgraph S, and (2) prompting an LLM via ICL with demonstrations (S_i, q_i, a_i) to generate an accurate answer.

WWW '23, Austin, TX, USA

ACM ISBN 978-1-4503-9416-1/23/04
https://doi.org/10.1145/3543507.3583316

3 Proposed Method

Our method tackles two key challenges in textual graph question answering: noisy subgraphs and limited examples. Given a query q and textual graph $G = (V, E)$, we retrieve a subgraph S using GRetriever [6]. To reduce noise in S, we apply a query-aware graph attention network that emphasizes relevant nodes and edges. To further enhance reasoning, we use few-shot in-context learning with selected (q_i, S_i, a_i) examples, enabling the model to learn from both textual and structural patterns. The framework is shown in Figure 1.

3.1 Subgraph Retrieval

Given a query q, we use a pretrained language model (Sentence-BERT [42]) $\mathrm{LM}(\cdot)$ to encode q, nodes, and edges in the textual graph $G = (V, E)$:

$$\mathbf{z}_q = \mathrm{LM}(q), \mathbf{z}_{v_i} = \mathrm{LM}(v_i), \mathbf{z}_{e_{i,j}} = \mathrm{LM}(e_{i,j}) \tag{1}$$

We compute cosine similarities between $\mathbf{z}_q$ and all node/edge embeddings and retrieve the top-k nodes V_k and edges E_k with highest similarity. A connected subgraph is then constructed using the Prize-Collecting Steiner Tree (PCST) algorithm [6]. Each retrieved item is assigned a prize based on its rank, and PCST selects a subgraph S that maximizes total prize while minimizing edge cost:

$$S = \underset{S \subseteq G}{\mathrm{argmax}} \Big(\sum_{v_i \in V_k} \mathrm{prize}(v_i) + \sum_{e_{i,j} \in E_k} \mathrm{prize}(e_{i,j}) - c \cdot |E_S| \Big).$$

3.2 Demonstration Retrieval

Building on our subgraph retrieval method, we introduce an approach to select informative few-shot examples for improved language model reasoning. Given a query q, we convert its retrieved subgraph S into text using $\mathrm{textualize}(\cdot)$, which flattens all node and edge attributes. The textualized graph is concatenated with q to form the prompt x: $x = \mathrm{textualize}(S)||q$.

While nearest-neighbor retrieval in embedding space is common, it may miss globally relevant examples. Inspired by recent work [46], we instead cluster demonstrations by semantic similarity and select a representative from each cluster for more diverse and complementary examples. Specifically, We apply K-means clustering to partition the example pool $\mathcal{D} = (S_i, q_i, a_i)_{i=1}^n$ into C clusters $C_1, C_2, \ldots, C_C$, treating each cluster as an expert. Clustering is performed on the Sentence-BERT embeddings of the augmented prompts $x_i = \mathrm{textualize}(S_i)||q_i$, following prior work [6].

To adaptively determine the optimal number of clusters C, we minimize a regularized objective that balances within-cluster variance and model complexity:

$$C^* = \min_C \sum_{k=1}^{C} \sum_{x_i \in C_k} |f(x_i) - \boldsymbol{\mu}_k|_2^2 + \lambda C, \tag{2}$$

where $f(\cdot)$ denotes the embedding function, $\boldsymbol{\mu}_k$ is the centroid of cluster C_k, and λ controls the regularization strength.

At inference time, a test query q is augmented into $x_q = \mathrm{textualize}(S_q)||q$, embedded via $f(x_q)$, and assigned to its closest expert based on cosine similarity with cluster centroids:

$$c(q) = \arg \max_{i=1,\ldots,C^*} \cos(f(x_q), \boldsymbol{\mu}_i). \tag{3}$$

The selected expert then provides a set of representative demonstrations, which are combined with the input to generate the model's final prediction.

3.3 Noise Mitigation

Building on our retrieval and demonstration selection framework (Section 3.2), we now describe how the final answer is generated using the retrieved subgraphs. Specifically, we encode each subgraph S with a *GraphEncoder* that transforms its structural and semantic information into a graph-prompt representation. This encoding serves two purposes: (1) preserving relational patterns critical to the query, and (2) filtering irrelevant information through query-sensitive attention, thereby constructing an optimized input for the language model.

Prior methods like G-Retriever use GCNs [10] or GATs [44], but these suffer from over-smoothing [2], making node embeddings indistinguishable, an issue in our setting where retrieved subgraphs mix relevant and noisy content. Effective encoding thus requires *query-aware* representations that selectively highlight important nodes. For instance, given a query, the encoder should emphasize `a cut peony` and ignore irrelevant nodes like `a flying eagle`, even if structurally nearby. To achieve this, we design a query-conditioned GNN where both message passing and node interactions are modulated by the input query q. Specifically, we redefine the edge weight $\zeta_{e_{i,j}}^{(l)}$ using a query-aware attention mechanism, allowing the model to focus on the most informative edges. At each layer l, the attention weight $\zeta_{e_{i,j}}^{(l)}$ is computed by:

$$\alpha_{v_i}^{(l)} = \texttt{LINEAR}\left(\texttt{CONCAT}(z_{v_i}^{(l)}, q)\right) \tag{4}$$

$$\beta_{v_j}^{(l)} = \texttt{LINEAR}\left(\texttt{CONCAT}(z_{v_j}^{(l)}, q)\right) \tag{5}$$

$$\gamma_{e_{i,j}} = \texttt{LINEAR}\left(\texttt{CONCAT}(z_{e_{i,j}}, q)\right) \tag{6}$$

$$\zeta_{e_{i,j}}^{(l)} = \texttt{tanh}\left(\alpha_{v_i}^{(l)} + \gamma_{e_{i,j}} - \beta_{v_j}^{(l)}\right) \tag{7}$$

where $\alpha_{v_i}^{(l)}$, $\beta_{v_j}^{(l)}$, and $\gamma_{e_{i,j}}$ are learned query-conditioned node/edge embeddings. $\zeta_{e_{i,j}}^{(l)}$ serves as the attention weight for message passing along edge $e_{i,j}$. Similarly, messages are generated using query-conditioned features:

$$\mathrm{msg}_{e_{i,j}}^{(l)} = \texttt{LINEAR}\left(\texttt{CONCAT}(z_{v_i}^{(l)}, z_{v_j}^{(l)}, z_{e_{i,j}}, q)\right) \tag{8}$$

and aggregated via attention weights:

$$z_{v_j}^{(l+1)} = \frac{1}{d_{v_j}} \sum_{v_i \in \mathcal{N}(v_j)} \zeta_{e_{i,j}}^{(l)} \cdot \mathrm{msg}_{e_{i,j}}^{(l)} \tag{9}$$

Unlike standard GCNs that apply static, query-agnostic filters, our GNN conditions node interactions and message passing on the query q. Specifically, we redefine edge weights $\zeta_{e_{i,j}}^{(l)}$ using a query-aware attention mechanism, allowing the model to emphasize task-relevant nodes and edges.

After L layers, each node $v_j \in S$ has an embedding $z_{v_j}^{(L)}$, which we mean-pool to obtain the subgraph representation: $z_S = \texttt{POOL}(z_{v_j}^{(L)})$.

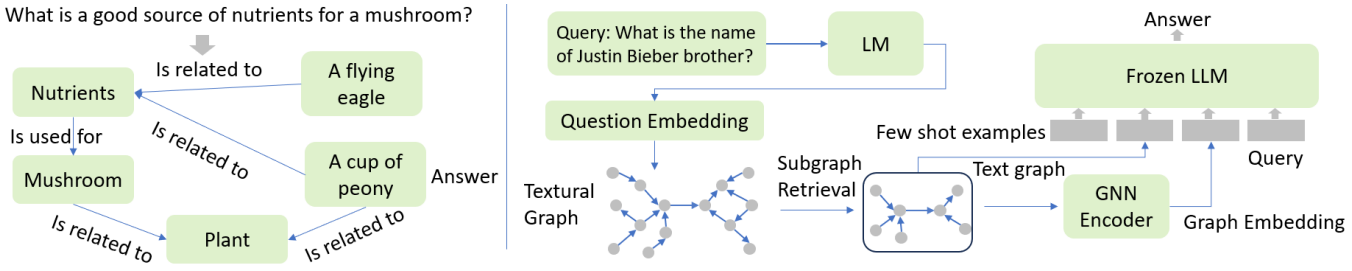


Figure 1: (a) An example of a retrieved subgraph. (b) Overview of MixDemo.

This is projected into the LLM embedding space using an MLP: $p_{\text{graph}} = \texttt{MLP}(z_S)$, serving as the graph prompt.

To incorporate multiple demonstration subgraphs $\{z_d\}_{d \in D_R}$, we use query-based relevance weighting:

$$\lambda(q, z_d) = \frac{e^{s(q,z_d)}}{\sum_{d'} e^{s(q,z_{d'})}}, z_{\text{final}} = \sum_{d=0}^{N} \lambda(q, z_d) z_d, \tag{10}$$

where $z_0 \equiv z_{\text{current}}$.

Generating responses. We prepend task-specific instructions and tokenize all inputs: $q = \texttt{tokenize}(q)$, $p_{\text{demo}} = \texttt{tokenize}(p_{\text{demo}})$, $p_{\text{text-graph}} = \texttt{tokenize}(p_{\text{text-graph}})$, then feed the combined sequence into the frozen LLM:

$$a_{\text{gen}} = \texttt{LLM}(\texttt{CONCAT}(p_{\text{demo}}, p_{\text{graph}}, p_{\text{text-graph}}, q)), \tag{11}$$

yielding the final answer a_{gen}.

4 Experiments

Table 1: Statistics of datasets. FB means FreeBase.

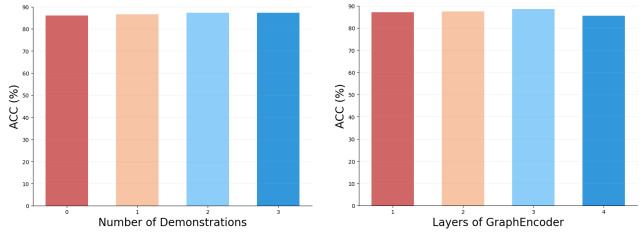
Dataset	ExplaGraphs	SceneGraphs	WebQSP
#Graphs	2,766	100,000	4,737
Average #Nodes	5.17	19.13	1370.89
Average #Edges	4.25	68.44	4252.37
Node Attribute	concepts	Object attributes	Entities in FB
Edge Attribute	relations	Spatial relations	Relations in FB
Task	reasoning	Scene graph QA	KGQA
Evaluation metrics	Accuracy	Accuracy	Hit@1

Datasets. We evaluate on the GraphQA benchmark [6], which includes ExplaGraphs, SceneGraphs, and WebQSP. Dataset stats are in Table 1 in the Appendix. **Metrics.** Following GRetriever, we use accuracy for ExplaGraphs and SceneGraphs, and Hit@1 for WebQSP, which allows multiple correct answers. **Baselines.** We compare against inference-only methods (e.g., Zero-shot [11], CoT-BAG [45], KAPING [1]) and prompt-tuning methods (e.g., Prompt Tuning, GraphToken [39], G-Retriever [6]).

4.1 Effectiveness of MixDemo

The results are summarized in Table 2, which compares MixDemo against all baseline methods. Overall, MixDemo consistently achieves superior performance across all datasets. For example, it outperforms the strongest baseline, G-Retriever, by approximately 1.1% on ExplaGraphs and 1.5% on SceneGraphs. These improvements highlight the effectiveness of the proposed approach. Additionally, we note that naively textizing the retrieved subgraph information and using it as direct input for LLMs often yields poor results, in most cases, performance is significantly degraded. This demonstrates the importance of properly encoding subgraph structural information and integrating it into LLMs.

4.2 Ablation Study



(a) Study on Demo Number **(b) Study on GNN's Layers**

Figure 2: Ablation study.

We first evaluate how the number of few-shot examples affects MixDemo's performance under zero-shot, 1-shot, 2-shot, and 3-shot settings. Since subgraphs in SceneGraphs and WebQSP exceed the LLM's input limit, we perform this study only on ExplaGraphs. As shown in Figure 2a, 2-shot learning yields the best performance, with 2-shot and 3-shot results being nearly identical.

In the hyperparameter study, we assess how the number of GraphEncoder layers impacts performance. As shown in Figure 2b, using three layers achieves the highest accuracy. Adding more layers offers no further improvement and can cause overfitting. These results highlight the importance of tuning encoder depth for optimal reasoning in MixDemo.

5 Related work

Recent work has highlighted Retrieval-Augmented Generation (RAG) [5] as a powerful solution to mitigate key limitations of large language models (LLMs), particularly their tendency toward hallucinations in domain-specific or knowledge-intensive tasks. Current RAG methodologies can be broadly grouped into three paradigms. The simplest form, naive RAG [37], operates through a basic pipeline of indexing, retrieval, and generation. Building upon this foundation, advanced RAG systems incorporate optimizations during

Table 2: Performance comparison for different methods (%).

Dataset (Metrics)	ExplaGraphs (ACC)	SceneGraphs (ACC)	WebQSP (Hit@1)
Zero-shot	56.50	39.74	41.06
Zero-CoT (Kojima et al., 2022)	57.04	52.60	51.30
CoT-BAG (Wang et al., 2024)	57.94	56.80	39.60
KAPING (Baek et al., 2023)	62.27	43.75	52.64
Graph-based Inference	33.93	42.17	47.22
Frozen LLM + Prompt Tuning (PT)	58.98	63.72	54.11
GraphToken (Perozzi et al., 2024)	85.08	49.03	57.05
G-Retriever	86.19	80.86	70.02
MixDemo	87.31	82.32	71.36

pre-retrieval, leveraging techniques like query transformation, expansion, and rewriting [38, 48], while post-retrieval enhancements often involve reranking strategies [40].

The Mixture of Experts (MoE) framework [8] has established itself as a fundamental paradigm in machine learning for developing adaptive systems and knowledge graph reasoning [7, 12–16, 16–27, 27, 27, 28, 28–36, 47]. Initial work focused on traditional machine learning implementations [9], with subsequent breakthroughs emerging through its integration with deep neural networks [10]. More recently, researchers have explored applying MoE approaches to in-context learning scenarios [46], demonstrating their potential to enhance large language model (LLM) performance.

6 Conclusion

We present MixDemo, a GraphRAG framework which leverages a Mixture-of-Experts demonstration selector and a query-aware graph encoder. By dynamically selecting contextually relevant demonstrations and filtering noisy subgraph information, our approach significantly improves answer accuracy and reasoning robustness across textual graph benchmarks. Experimental results validate that MixDemo outperforms state-of-the-art baselines, demonstrating the importance of adaptive retrieval and noise reduction in GraphRAG systems.

References

[1] Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, Bhavana Dalvi Mishra, Greg Durrett, Peter Jansen, Danilo Neves Ribeiro, and Jason Wei (Eds.). Association for Computational Linguistics, Toronto, Canada.

[2] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 3438–3445.

[3] Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models. arXiv:2401.06066 [cs.CL] https://arxiv.org/abs/2401.06066

[4] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. arXiv:2404.16130 [cs.CL] https://arxiv.org/abs/2404.16130

[5] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise Zero-Shot Dense Retrieval without Relevance Labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 1762–1777. doi:10.18653/v1/2023.acl-long.99

[6] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 132876–132907. https://proceedings.neurips.cc/paper_files/paper/2024/file/efaf1c9726648c8ba363a5c927440529-Paper-Conference.pdf

[7] Blaine Hill, Lihui Liu, and Hanghang Tong. 2024. Ginkgo-P: General Illustrations of Knowledge Graphs for Openness as a Platform. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 1066–1069.

[8] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive Mixtures of Local Experts. *Neural Computation* 3, 1 (1991), 79–87. doi:10.1162/neco.1991.3.1.79

[9] Michael Jordan, Zoubin Ghahramani, and Lawrence Saul. 1996. Hidden Markov Decision Trees. In *Advances in Neural Information Processing Systems*, M.C. Mozer, M. Jordan, and T. Petsche (Eds.), Vol. 9. MIT Press. https://proceedings.neurips.cc/paper_files/paper/1996/file/6c8dba7d0df1c4a79dd07646be9a26c8-Paper.pdf

[10] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907 [cs.LG] https://arxiv.org/abs/1609.02907

[11] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large Language Models are Zero-Shot Reasoners. arXiv:2205.11916 [cs.CL] https://arxiv.org/abs/2205.11916

[12] Lihui Liu. 2024. *Knowledge graph reasoning and its applications: A pathway towards neural symbolic AI.* Ph. D. Dissertation. University of Illinois at Urbana-Champaign.

[13] Lihui Liu. 2025. Graph-O1: Monte Carlo Tree Search with Reinforcement Learning for Text-Attributed Graph Reasoning. *arXiv preprint arXiv:2512.17912* (2025).

[14] Lihui Liu. 2025. HyperKGR: Knowledge Graph Reasoning in Hyperbolic Space with Graph Neural Network Encoding Symbolic Path. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. 25188–25199.

[15] Lihui Liu. 2025. Monte Carlo Tree Search for Graph Reasoning in Large Language Model Agents. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. 4966–4970.

[16] Lihui Liu, Yuzhong Chen, Mahashweta Das, Hao Yang, and Hanghang Tong. 2023. Knowledge Graph Question Answering with Ambiguous Query. In *Proceedings of the ACM Web Conference 2023*. 2477–2486.

[17] Lihui Liu, Jiayuan Ding, Subhabrata Mukherjee, and Carl J Yang. 2025. MIXRAG: Mixture-of-Experts Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. *arXiv preprint arXiv:2509.21391* (2025).

[18] Lihui Liu, Boxin Du, Yi Ren Fung, Heng Ji, Jiejun Xu, and Hanghang Tong. 2021. KompaRe: A Knowledge Graph Comparative Reasoning System. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3308–3318.

[19] Lihui Liu, Boxin Du, Heng Ji, ChengXiang Zhai, and Hanghang Tong. 2021. Neural-Answering Logical Queries on Knowledge Graphs. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1087–1097.

[20] Lihui Liu, Boxin Du, Hanghang Tong, et al. 2019. G-finder: Approximate attributed subgraph matching. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 513–522.

[21] Lihui Liu, Boxin Du, Jiejun Xu, Yinglong Xia, and Hanghang Tong. 2022. Joint Knowledge Graph Completion and Question Answering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1098–1108.

[22] Lihui Liu, Blaine Hill, Boxin Du, Fei Wang, and Hanghang Tong. 2024. Conversational Question Answering with Language Models Generated Reformulations over Knowledge Graph. In *Findings of the Association for Computational Linguistics ACL 2024*. 839–850.

[23] Lihui Liu, Houxiang Ji, Jiejun Xu, and Hanghang Tong. 2022. Comparative Reasoning for Knowledge Graph Fact Checking. In *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 2309–2312.

[24] Lihui Liu, Jinha Kim, and Vidit Bansal. 2024. Can Contrastive Learning Refine Embeddings. *arXiv preprint arXiv:2404.08701* (2024).

[25] Lihui Liu and Kai Shu. 2025. Unifying knowledge in agentic llms: Concepts, methods, and recent advancements. *ACM SIGKDD Explorations Newsletter* 27, 2 (2025), 88–96.

[26] Lihui Liu and Hanghang Tong. [n. d.]. Neural Symbolic Knowledge Graph Reasoning. ([n. d.]).

[27] Lihui Liu and Hanghang Tong. 2026. Accurate Query Answering with LLMs Over Incomplete KG. In *Neural Symbolic Knowledge Graph Reasoning: A Pathway Towards Neural Symbolic AI*. Springer, 73–87.

[28] Lihui Liu and Hanghang Tong. 2026. Ambiguous Query Answering with Neural Symbolic Reasoning Over Incomplete KG. In *Neural Symbolic Knowledge Graph Reasoning: A Pathway Towards Neural Symbolic AI*. Springer, 89–106.

[29] Lihui Liu and Hanghang Tong. 2026. Dynamic Query Answering with Neural Symbolic Reasoning Over Incomplete KG. In *Neural Symbolic Knowledge Graph Reasoning: A Pathway Towards Neural Symbolic AI*. Springer, 121–136.

[30] Lihui Liu and Hanghang Tong. 2026. *Neural Symbolic Knowledge Graph Reasoning: A Pathway Towards Neural Symbolic AI*. Springer Nature.

[31] Lihui Liu and Hanghang Tong. 2026. Symbolic Reasoning for Inconsistency Detection Over Complete KG. In *Neural Symbolic Knowledge Graph Reasoning: A Pathway Towards Neural Symbolic AI*. Springer, 37–54.

[32] Lihui Liu, Zihao Wang, Jiaxin Bai, Yangqiu Song, and Hanghang Tong. 2024. New frontiers of knowledge graph reasoning: Recent advances and future trends. In *Companion Proceedings of the ACM Web Conference 2024*. 1294–1297.

[33] Lihui Liu, Zihao Wang, Ruizhong Qiu, Yikun Ban, Eunice Chan, Yangqiu Song, Jingrui He, and Hanghang Tong. 2024. Logic query of thoughts: Guiding large language models to answer complex logic queries with knowledge graphs. *arXiv preprint arXiv:2404.04264* (2024).

[34] Lihui Liu, Zihao Wang, and Hanghang Tong. 2025. Neural-symbolic reasoning over knowledge graphs: A survey from a query perspective. *ACM SIGKDD Explorations Newsletter* 27, 1 (2025), 124–136.

[35] Lihui Liu, Zihao Wang, Dawei Zhou, Ruijie Wang, Yuchen Yan, Bo Xiong, Sihong He, and Hanghang Tong. 2025. Few-Shot Knowledge Graph Completion via Transfer Knowledge from Similar Tasks. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. 4960–4965.

[36] Lihui Liu, Ruining Zhao, Boxin Du, Yi Ren Fung, Heng Ji, Jiejun Xu, and Hanghang Tong. 2022. Knowledge Graph Comparative Reasoning for Fact Checking: Problem Definition and Algorithms. *IEEE Data Eng. Bull.* 45, 4 (2022), 19–38.

[37] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query Rewriting in Retrieval-Augmented Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore.

[38] Wenjun Peng, Guiyang Li, Yue Jiang, Zilong Wang, Dan Ou, Xiaoyi Zeng, Derong Xu, Tong Xu, and Enhong Chen. 2024. Large Language Model based Long-tail Query Rewriting in Taobao Search. arXiv:2311.03758 [cs.IR] https://arxiv.org/abs/2311.03758

[39] Bryan Perozzi, Bahare Fatemi, Dustin Zelle, Anton Tsitsulin, Mehran Kazemi, Rami Al-Rfou, and Jonathan Halcrow. 2024. Let Your Graph Do the Talking: Encoding Structured Data for LLMs. arXiv:2402.05862 [cs.LG] https://arxiv.org/abs/2402.05862

[40] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico.

[41] A Radford, J Wu, and R Child. 2018. Language Models are Unsupervised Multitask Learners. (2018).

[42] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China.

[43] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652* (2023).

[44] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. arXiv:1710.10903 [stat.ML] https://arxiv.org/abs/1710.10903

[45] Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. 2024. Can Language Models Solve Graph Problems in Natural Language? arXiv:2305.10037 [cs.CL] https://arxiv.org/abs/2305.10037

[46] Ruochen Wang, Sohyun An, Minhao Cheng, Tianyi Zhou, Sung Ju Hwang, and Cho-Jui Hsieh. 2024. One Prompt is not Enough: Automated Construction of a Mixture-of-Expert Prompts. In *International Conference on Machine Learning*.

[47] Shanglin Wu, Lihui Liu, Jinho D Choi, and Kai Shu. 2025. Improving Factuality in LLMs via Inference-Time Knowledge Graph Construction. *arXiv preprint arXiv:2509.03540* (2025).

[48] Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2024. Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models. arXiv:2310.06117 [cs.LG] https://arxiv.org/abs/2310.06117