PathGPT: Path-enhanced LLMs for Knowledge Graph Completion

1st Tao Huang[1,2]
[1]*Hangzhou Institute of Medicine, Chinese Academy of Sciences*
Hangzhou, China
[2]*University of Chinese Academy of Sciences*
Beijing, China
htao36@163.com

2nd Junwei Zhang*
Hangzhou Institute of Medicine, Chinese Academy of Sciences
Hangzhou, China
zhangjunwei@him.cas.cn

3rd Pengju Yan
Hangzhou Institute of Medicine, Chinese Academy of Sciences
Hangzhou, China
yanpengju@gmail.com

4th Rui Yang
School of Nursing
Zhejiang Chinese Medical University
Hangzhou, China
kelseyjhonna@163.com

5th Xiaolin Li[1,2,3*]
[1]*Hangzhou Institute of Medicine, Chinese Academy of Sciences*
Hangzhou, China
[2]*University of Chinese Academy of Sciences*
Beijing, China
[3]*School of Statistics and Data Science, School of Information and Electronic Engineering, Zhejiang AIX College, Zhejiang Gongshang University*
Hangzhou, China
xiaolinli@ieee.org

***Abstract*—Knowledge graph completion requires establishing credible reasoning chains between structure and text. However, existing methods are either confined to low-dimensional embeddings or neglect explicit graph structures, leading to hallucinations and insufficient explainability. To address this, this paper proposes PathGPT, a path-enhanced large language model framework. The core idea of PathGPT is to use key paths as evidence, textual paths as prompts, and adapters as bridges to achieve deep synergy between structural reasoning and semantic generation. Experiments on three benchmarks demonstrate that PathGPT significantly outperforms existing state-of-the-art methods in terms of accuracy and F1 score. Moreover, PathGPT possesses explainability and generalisation ability, effectively integrating structural path information and providing a new technical route for knowledge graph completion.**

***Index Terms*—Knowledge graph completion, GPT, Path-enhanced, Large language model.**

I. Introduction

Knowledge Graphs (KGs), which structure real-world knowledge as triples (head, relation, tail), are vital for semantic reasoning in AI systems [1]. They have advanced applications like question answering [2] and recommendation [3]. Despite their scale (e.g., DBpedia [4], Freebase [5]), inherent incompleteness limits their utility. Knowledge Graph Completion (KGC) offers a solution by leveraging existing graph structures and associated text to infer missing triples [6]. Predominant KGC paradigms include embedding-based methods (learning vector representations), path-based methods (reasoning over

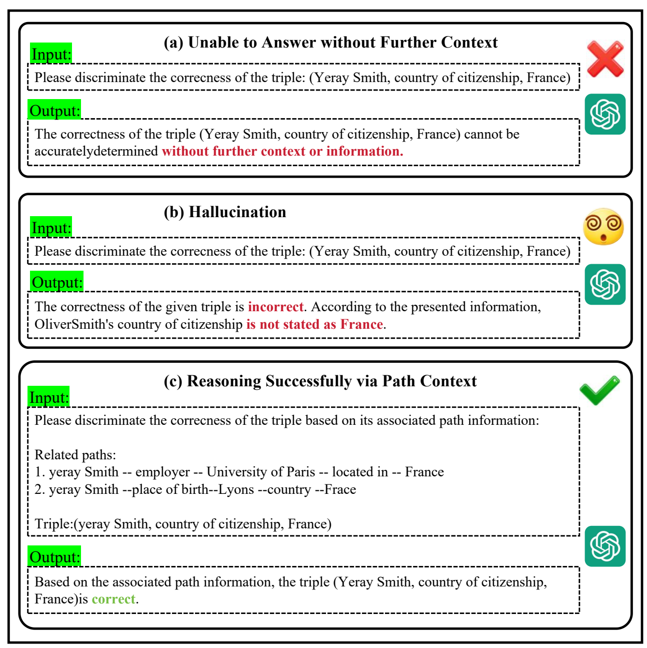

Fig. 1. A simple and intuitive example. By providing path context, LLM produces more grounded predictions and less hallucination.

This work is supported in part by funds from the National Key Research and Development Program of China (2022YFC3600902), Key Research and Development Program of Zhejiang (2025C01129) and the National Science Technology Major Project (No. 2023ZD0509706).

multi-hop paths), and language model-based methods (leveraging textual descriptions). In recent years, Large Language Models (LLMs) have demonstrated powerful reasoning abilities [7], [8]. However, LLM-based KGC often relies on isolated triples and descriptions, neglecting structural dependencies and risking factual inaccuracy. While recent work integrates KG structure into LLMs [8], two key limitations persist: (1) Modality misalignment between KG structure and LLM textual space, and (2) Path neglect, where triple-level integration fails to leverage relational paths for deeper reasoning.

To address this, we argue that explicit path integration is essential, as reasoning over knowledge graph (KG) paths enables LLMs to assess triple validity, improve answer accuracy, and reduce hallucinations. However, the exponential number of inter-entity paths poses challenges for integration within the limited input window of LLMs. This paper proposes PathGPT—a KG completion framework that introduces high-quality reasoning paths via a learnable alignment mechanism to bridge the modality gap and maximize LLMs' structural reasoning capacity. The framework employs a two-phase strategy:

- **Structural Path Sampling**: During pre-training, multi-hop paths between entity pairs are extracted. A scoring function retains the top-N most discriminative paths, filtering noise while preserving critical structural semantics.
- **Path-Aware Inference:** A lightweight adapter compresses the filtered paths into fixed-length virtual tokens in the LLM's embedding space. These tokens, along with textual path descriptions in prompt templates, are prepended to the input. Using LoRA fine-tuning [9], only the adapter and LoRA parameters are updated, keeping the LLM backbone frozen. This ensures deep structure-semantics integration while avoiding catastrophic forgetting and preserving the model's linguistic capabilities.

PathGPT integrates path information from knowledge graphs with the language generation capabilities of large language models, achieving substantial performance gains in knowledge reasoning. On UMLS, CoDeX-S, and FB15K-237N, instruction fine-tuning enables the model to surpass 18 baseline methods, including embedding-based, path-based, and pure LLM approaches. It achieves the highest accuracy and F1 scores while effectively mitigating hallucination.

In summary, this work makes two key contributions: (1) proposing a knowledge graph completion framework that effectively selects and integrates essential path information with LLMs; and (2) validating the framework's effectiveness through comprehensive experiments on three benchmark datasets.

II. Related Work

Knowledge Graph Completion (KGC) aims to infer missing facts (triples) to address the inherent incompleteness of real-world Knowledge Graphs (KGs) [10]. Among its subtasks, this study focuses on triple classification [8]. Predominant KGC methodologies include embedding-based, path-based, and Pre-trained Language Model (PLM)-based approaches. Embedding-based methods [11], [12] learn low-dimensional vector representations of entities and relations, effectively capturing local graph structures but largely overlooking rich textual semantics. In contrast, PLM-based methods [13], [14] treat triples as textual sequences, leveraging the semantic power of models like BERT [15] for classification, yet they often struggle to encode explicit, multi-hop graph topology. Path-based methods offer a distinct advantage in interpretable reasoning by leveraging relational paths between entities [16]. Early methods relied on inefficient heuristic enumeration or reinforcement learning with sparse rewards [17]. Subsequent advancements introduced more efficient and scalable algorithms for learning path representations [18]–[20], establishing path-based reasoning as a robust paradigm for KGC due to its interpretability and strong performance.

The integration of Large Language Models (LLMs) [21], [22] with KGs has emerged as a crucial research direction to enhance the factual accuracy of LLMs and mitigate hallucination [23], while also empowering KG-related tasks [24]. Recent efforts to adapt LLMs for KGC primarily follow two paradigms. The first paradigm treats triples as plain text for sequence classification via fine-tuning [7], which discards essential graph structural information. The second paradigm seeks to align KG embeddings with the LLM's semantic space using adapter modules [8], but a significant gap often remains between discrete graph embeddings and continuous textual representations. Effectively incorporating explicit, interpretable path information into LLMs is recognized as a promising solution to bridge the structural-semantic divide. However, this approach faces two primary challenges: the exponential number of potential paths between entities, which demands efficient sampling strategies [20], and the non-trivial alignment between symbolic path sequences and the LLM's latent semantic space.

III. PathGPT: The Proposed Framework

PathGPT is a two-stage framework designed to efficiently inject multi-hop path information from KGs into LLMs for enhanced triple completion. As illustrated in Figure Figure 2, the framework first samples discriminative structural paths and then integrates them into the LLM through a lightweight adapter and prefix-enhanced prompting.

A. Structural Path Sampling

We employ a pre-trained A*Net model to extract and rank key relational paths that connect a given head entity h to a tail entity t within the knowledge graph. When handling an incomplete triple represented as (h, x, t), the model performs multi-hop path reasoning over the graph structure and selects the most informative connecting paths through a learned scoring mechanism. This process captures complex relational dependencies while filtering out less relevant connections.

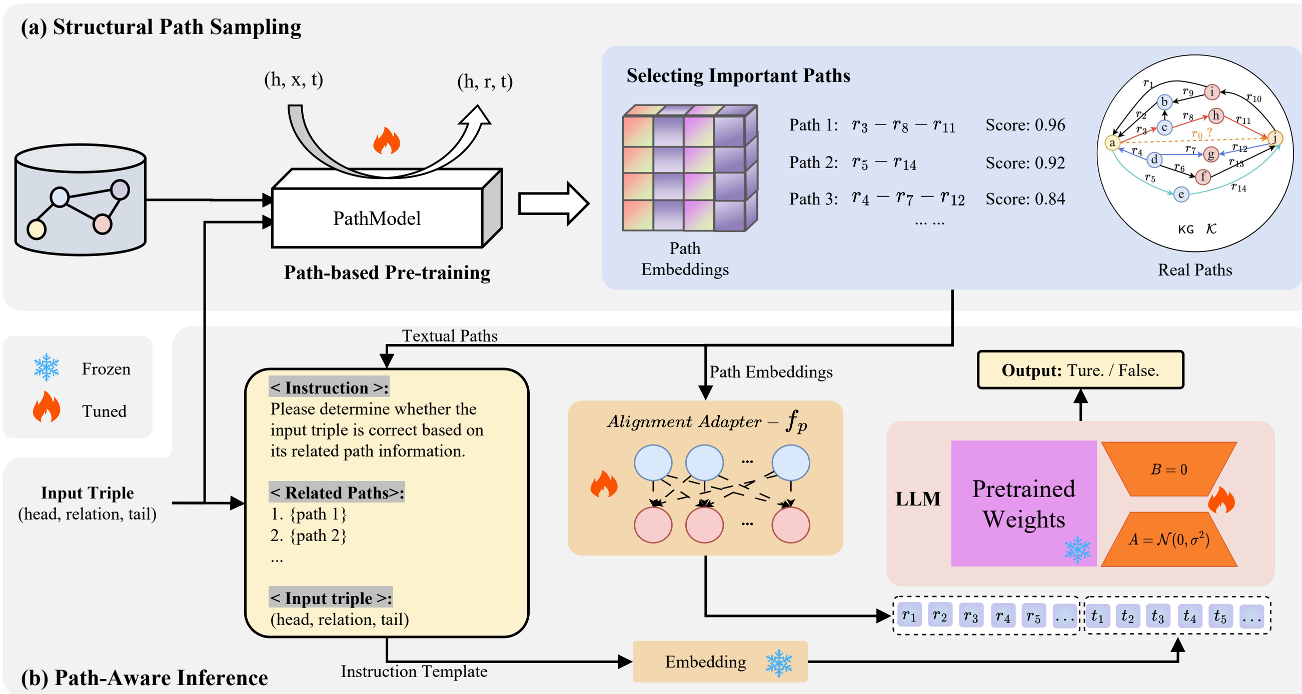

Fig. 2. An Overview of PathGPT. (a) Structural Path Sampling. By employing a pre-trained path model, multi-hop paths between two entities are extracted. Subsequently, a scoring function is utilised to filter out the most discriminative paths. (b) Path-Aware Inference. The filtered paths are embedded and mapped into the LLM word vector space. Before injecting the path tokens into the text prompt, fine-tuning is achieved through LoRA, thereby enabling a deep integration of structure and semantics.

The path model $\mathcal{A}^*$ is first pre-trained on the graph $\mathcal{G}$ to predict the relationship r:

$$\mathcal{A}^* = \arg\min_{\theta} \left(- \sum_{(h,r,t)\in\mathcal{G}} \log p(r|h,t;\theta) \right) \tag{1}$$

where θ denotes the model parameters. After pre-training, θ is fixed.

For a given (h,t) pair, we extract the top-N paths $\mathcal{P}_{\text{Top-N}}$ based on a path importance score, which aggregates the learned node priority values $s_{h,t}(v)$ along each path P:

$$\mathcal{P}_{\text{Top-N}} = \text{Top-N}\left(\mathcal{P}_{h\rightsquigarrow t}, ; \sum_{v\in P} s_{h,t}(v) \right) \tag{2}$$

From these paths, we derive corresponding path embeddings $\mathcal{E}_{\text{Top}-\text{N}} = e(P) \mid P \in \mathcal{P}_{\text{Top-N}}$ and their textual descriptions $\mathcal{T}_{\text{Top}-\text{N}} = T(P) \mid P \in \mathcal{P}_{\text{Top-N}}$. These representations encode both structural and semantic information of the most salient paths between h and t, providing critical evidence for subsequent reasoning steps.

These representations encode both structural and semantic information of the most salient paths between h and t, providing critical evidence for subsequent reasoning steps.

B. Path-Aware Inference

Building upon the extracted key paths $\mathcal{P}_{\text{Top-N}}$, this phase integrates structural path information into the LLM through a lightweight MLP adapter. The adapter maps each path embedding $e(P_i)$ to a virtual token v_{P_i} compatible with the LLM's embedding space. These tokens are prepended to the prompt embeddings as a structured prefix:

$$E_{\text{enhanced}} = [v_{P_1}, \dots, v_{P_N}, e_1, \dots, e_m] \tag{3}$$

enabling the LLM to perceive relational evidence before processing the textual query.

For efficient fine-tuning, we adopt Low-Rank Adaptation (LoRA), updating only the adapter and low-rank matrices while keeping the base LLM frozen. This strategy mitigates catastrophic forgetting and enables effective knowledge integration. Compared to existing LLM-based KGC methods, PathGPT uniquely injects structured path embeddings, enhancing relational reasoning and interpretability.

IV. Experiments

A. Datasets and Experimental Settings

We employed three publicly available knowledge graph benchmark datasets to validate our model: UMLS [13], CoDeX-S [25], and FB15K-237N [26]. UMLS is a medical knowledge graph, while CoDeX-S and FB15K-237N are encyclopaedic graphs extracted from Wikidata. These benchmark datasets provide challenging evaluations, incorporating difficult-to-distinguish negative samples. This design aims to prevent the occurrence of false negatives in the validation or test datasets [8].

TABLE I

PERFORMANCE COMPARISONS OF DIFFERENT TRIPLE CLASSIFICATION MODELS USING DIFFERENT EVALUATION METRICS.

Dataset	**UMLS** [13]				**CoDeX-S** [25]				**FB15K-237N** [26]			
Metric	**Acc**	**P**	**R**	**F1**	**Acc**	**P**	**R**	**F1**	**Acc**	**P**	**R**	**F1**
Embedding-based methods												
TransE [11]	84.49	86.53	81.69	84.04	72.07	71.91	72.42	72.17	69.71	70.80	67.11	68.91
DistMult [27]	86.38	87.06	86.53	86.79	66.79	69.67	59.46	64.16	58.66	58.98	56.84	57.90
ComplEx [12]	90.77	89.92	91.83	90.87	67.64	67.84	67.06	67.45	65.70	66.46	63.38	64.88
RotatE [28]	92.05	90.17	94.41	92.23	75.68	75.66	75.71	75.69	68.46	69.24	66.41	67.80
Path-based methods												
CURL [16]	81.69	86.43	75.19	80.42	68.57	62.96	90.21	74.16	63.86	64.69	61.03	62.80
NBFNet [18]	92.58	89.81	96.06	92.83	79.34	79.42	79.21	79.32	71.79	67.14	85.32	75.15
A*Net [20]	91.60	86.47	**98.64**	92.15	79.18	72.37	94.42	81.93	69.98	64.33	89.68	74.92
PLM-based methods												
KG-BERT [13]	77.30	70.96	92.43	80.28	77.30	70.96	92.43	80.28	56.02	53.47	97.62	67.84
PKGC [26]	-	-	-	-	-	-	-	-	79.60	-	-	79.50
LLM-based training-free methods												
Zero-shot(Alpaca) [8]	52.64	51.55	87.69	64.91	50.62	50.31	**99.83**	66.91	56.06	53.32	97.37	68.91
Zero-shot(GPT-3.5) [8]	67.58	88.04	40.71	55.67	54.68	69.13	16.94	27.21	60.15	**86.62**	24.01	37.59
ICL [8]	55.52	55.85	52.65	54.21	50.62	50.31	**99.83**	66.91	59.23	57.23	73.02	64.17
Path Prompt(GPT-3.5)	54.92	54.16	63.99	58.66	63.29	65.20	57.00	60.82	60.24	70.36	35.38	47.09
LLM-based fine-tuning methods												
KG-LLaMA [7]	85.77	87.84	83.05	85.38	79.43	78.67	80.74	79.69	74.81	67.37	96.23	79.25
KG-Alpaca [7]	86.01	94.91	76.10	84.46	80.25	79.38	81.73	80.54	69.91	62.71	**98.28**	76.56
Textual Paths (Ours)	86.83	88.83	84.26	86.49	70.76	**94.59**	44.03	60.09	75.16	73.66	78.32	75.92
KoPA (LLaMa 2) [8]	91.45	**95.66**	86.83	91.04	81.94	75.63	94.25	83.92	77.42	71.82	90.26	79.99
KoPA (Alpaca) [8]	92.58	90.85	94.70	92.70	82.74	77.91	91.41	84.11	77.65	70.81	94.09	80.81
PathGPT (Ours)	**94.32**	92.59	96.39	**94.44**	**83.89**	82.72	85.66	**84.17**	**79.51**	75.56	87.24	**80.98**

We compared models from four representative categories. For embedding-based approaches, we included TransE [11], DistMult [27], ComplEx [12], and RotatE [28]. Path-based methods were represented by CURL [16], NBFNet [18], and A*Net [20]. As for PLM-based methods, we selected KG-BERT [13] and PKGC [26]. Finally, for LLM-based methods, we considered both training-free configurations (Zero-shot [8] and ICL [8]) and fine-tuned ones (KG-LLaMA [7], KG-Alpaca [7], and KoPA [8]).

In addition, we propose two novel baseline methods: **Path Prompt** and **Textual Paths**. “Path Prompt” guides zero-shot reasoning of LLMs by constructing text contexts related to knowledge graph paths. “Textual Paths” involves transforming path information into textual form for instruction fine-tuning of LLMs.

All experiments utilized a LLaMA-based 7B parameter LLM and were conducted on eight NVIDIA A100 GPUs.

B. Main Results

We evaluated our method using the triplet classification task. All test datasets were balanced with equal numbers of true and false triplets. While accuracy and F1 score were our main focus, we also included precision and recall in the results for detailed analysis. As shown in Table IV, PathGPT consistently outperformed all baseline models. For instance, on the UMLS dataset, it improved accuracy and F1 score by 1.74%, largely due to its effective use of pre-trained A*Net embeddings. This approach also enabled strong performance on the more challenging FB15K-237N dataset.

In contrast, generic LLMs such as GPT-3.5-turbo perform poorly on zero-shot triplet classification, even when path information is provided. This limitation highlights their inability to effectively utilize structural or relational cues without task-specific supervision. Training-free methods also yield unstable and biased predictions, further underscoring the unreliability of non-adaptive approaches. While fine-tuning LLMs with path information leads to improved results, PathGPT outperforms text-based prompting approaches, suggesting its capacity to capture richer semantic information.

C. Case Study

Fig. 3 shows the important paths learned by PathGPT for two test sample in UMLS. Given the query *(anatomical abnormali, result of, health care activity)*, we can see two

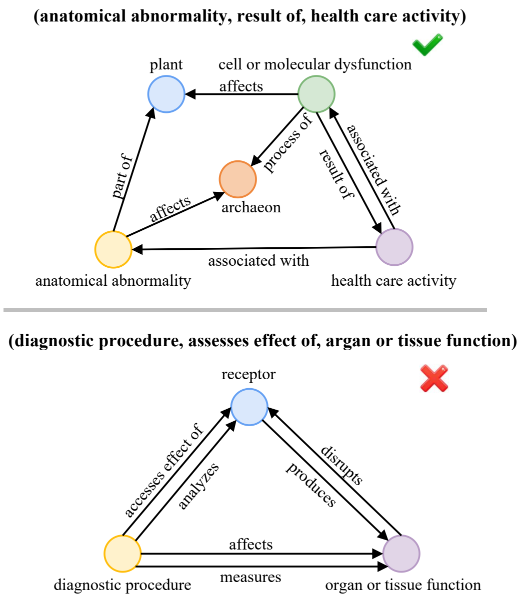

Fig. 3. Case study of different samples, including important learned paths of samples. "✓" indicates that the label of the sample is true, and "✗" indicates false.

paths *anatomical abnormali* $\xleftarrow{\textit{associated with}}$ *health care activity*, and *anatomical abnormali* $\xrightarrow{\textit{affects}}$ *archaeon* $\xleftarrow{\textit{process of}}$ *Cell or molecular dysfunction* $\xrightarrow{\textit{result of}}$ *health care activity* are consistent with human cognition. This proves that the reasoning capability of LLMs can be effectively improved by injecting path information embedding into LLMs.

For additional experimental details and a comprehensive description of the methodology, please refer to PathGPT.

V. Conclusion and Limitations

In conclusion, we propose PathGPT, a novel framework that enhances the knowledge reasoning capabilities of Large Language Models (LLMs) by integrating structured path information from knowledge graphs. The core of our approach is to deeply fuse this path information into LLMs, generating enriched prompts that guide the model towards more logically consistent and factually accurate predictions. Experimental evaluations on the triple classification task confirm that PathGPT significantly outperforms existing baseline methods. However, this work has limitations; it has not yet explored the integration of LLMs with a broader range of path-based reasoning methods. Future research will focus on developing a more unified framework capable of seamlessly combining diverse LLMs and path-based techniques, and on leveraging knowledge graphs to augment LLMs for downstream tasks—thereby enriching their factual knowledge and reducing hallucination.

References

[1] Z. Chen, Y. Zhang, Y. Fang, Y. Geng, L. Guo, X. Chen, Q. Li, W. Zhang, J. Chen, Y. Zhu, *et al.*, "Knowledge graphs meet multi-modal learning: A comprehensive survey," *arXiv preprint arXiv:2402.05391* , 2024.

[2] M. Su, Z. Li, Z. Chen, L. Bai, X. Jin, and J. Guo, "Temporal knowledge graph question answering: A survey," *arXiv preprint arXiv:2406.14191* , 2024.

[3] N. Arina, I. Hidayah, and N. A. Setiawan, "A survey on knowledge graph for enhancing the performance of llm-based recommendation systems," in *2026 18th International Conference on Knowledge and Smart Technology (KST)*, pp. 434–439, IEEE, 2026.

[4] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *international semantic web conference*, pp. 722–735, Springer, 2007.

[5] K. D. Bollacker, C. Evans, P. K. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *SIGMOD Conference*, pp. 1247–1250, ACM, 2008.

[6] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE transactions on neural networks and learning systems* **33**(2), pp. 494–514, 2021.

[7] L. Yao, J. Peng, C. Mao, and Y. Luo, "Exploring large language models for knowledge graph completion," *CoRR* **abs/2308.13916**, 2023.

[8] Y. Zhang, Z. Chen, L. Guo, Y. Xu, W. Zhang, and H. Chen, "Making large language models perform better in knowledge graph completion," in *Proceedings of the 32nd ACM international conference on multimedia*, pp. 233–242, 2024.

[9] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, *et al.*, "Lora: Low-rank adaptation of large language models.," *ICLR* **1**(2), p. 3, 2022.

[10] S. Moon and Y. Ko, "How sharp and bias-robust is a model? dual evaluation perspectives on knowledge graph completion," in *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, pp. 1211–1215, 2026.

[11] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *NIPS*, pp. 2787–2795, 2013.

[12] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *ICML*, *JMLR Workshop and Conference Proceedings* **48**, pp. 2071–2080, JMLR.org, 2016.

[13] L. Yao, C. Mao, and Y. Luo, "KG-BERT: BERT for knowledge graph completion," *CoRR* **abs/1909.03193**, 2019.

[14] B. Kim, T. Hong, Y. Ko, and J. Seo, "Multi-task learning for knowledge graph completion with pre-trained language models," in *COLING*, pp. 1737–1743, International Committee on Computational Linguistics, 2020.

[15] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT (1)*, pp. 4171–4186, Association for Computational Linguistics, 2019.

[16] D. Zhang, Z. Yuan, H. Liu, H. Xiong, *et al.*, "Learning to walk with dual agents for knowledge graph reasoning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, **36**(5), pp. 5932–5941, 2022.

[17] W. Xiong, T. Hoang, and W. Y. Wang, "Deeppath: A reinforcement learning method for knowledge graph reasoning," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, ACL, (Copenhagen, Denmark), September 2017.

[18] Z. Zhu, Z. Zhang, L. A. C. Xhonneux, and J. Tang, "Neural bellman-ford networks: A general graph neural network framework for link prediction," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 29476–29490, 2021.

[19] G. Niu, B. Li, Y. Zhang, Y. Sheng, C. Shi, J. Li, and S. Pu, "Joint semantics and data-driven path representation for knowledge graph reasoning," *Neurocomputing* **483**, pp. 249–261, 2022.

[20] Z. Zhu, X. Yuan, M. Galkin, S. Xhonneux, M. Zhang, M. Gazeau, and J. Tang, "A* net: A scalable path-based reasoning approach for knowledge graphs," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[21] OpenAI, "GPT-4 technical report," *CoRR* **abs/2303.08774**, 2023.

[22] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," *CoRR* **abs/2302.13971**, 2023.

[23] M. Park, H. Yang, J. Kim, K. Park, and H. Kim, "Prograg: Hallucination-resistant progressive retrieval and reasoning over knowledge graphs," in

Proceedings of the AAAI Conference on Artificial Intelligence, **40**(39), pp. 32674–32682, 2026.

[24] M. Besta, L. Paleari, J. H. A. Jiang, R. Gerstenberger, Y. Wu, J. G. Hannesson, P. Iff, A. Kubicek, P. Nyczyk, D. Khimey, *et al.*, "Affordable ai assistants with knowledge graph of thoughts," *arXiv preprint arXiv:2504.02670* , 2025.

[25] T. Safavi and D. Koutra, "Codex: A comprehensive knowledge graph completion benchmark," in *EMNLP (1)*, pp. 8328–8350, Association for Computational Linguistics, 2020.

[26] X. Lv, Y. Lin, Y. Cao, L. Hou, J. Li, Z. Liu, P. Li, and J. Zhou, "Do pre-trained models benefit knowledge graph completion? A reliable evaluation and a reasonable approach," in *ACL (Findings)*, pp. 3570–3581, Association for Computational Linguistics, 2022.

[27] B. Yang, W. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *ICLR (Poster)*, 2015.

[28] Z. Sun, Z. Deng, J. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," in *ICLR (Poster)*, OpenReview.net, 2019.