Walk Wisely on Graph: Knowledge Graph Reasoning with Dual Agents via Efficient Guidance-Exploration

Zijian Wang[1,2], **Bin Wang**[1*], **Haifeng Jing**[1,3], **Huayu Li**[1], **Hongbo Dou**[1]

[1]College of Computer Science and Technology & Qingdao Institute of Software, China University of Petroleum(East China), China

[2]College of Science, China University of Petroleum(East China), China

[3]School of Software & Microelectronics, Peking University, China

alococo710@gmail.com, {wangbin,lhyzj}@upc.edu.cn, 2201120005@stu.pku.edu.cn, hongboDou@163.com

Abstract

Recent years, multi-hop reasoning has been widely studied for knowledge graph (KG) reasoning due to its efficacy and interpretability. However, previous multi-hop reasoning approaches are subject to two primary shortcomings. First, agents struggle to learn effective and robust policies at the early phase due to sparse rewards. Second, these approaches often falter on specific datasets like sparse knowledge graphs, where agents are required to traverse lengthy reasoning paths. To address these problems, we propose a multi-hop reasoning model with dual agents based on hierarchical reinforcement learning (HRL), which is named **FULORA**. FULORA tackles the above reasoning challenges by e**F**ficient **GU**idance-Exp**LORA**tion between dual agents. The high-level agent walks on the simplified knowledge graph to provide stage-wise hints for the low-level agent walking on the original knowledge graph. In this framework, the low-level agent optimizes a value function that balances two objectives: (1) maximizing return, and (2) integrating efficient guidance from the high-level agent. Experiments conducted on three real-word knowledge graph datasets demonstrate that FULORA outperforms RL-based baselines, especially in the case of long-distance reasoning.

Introduction

Knowledge graphs (KGs) are designed to represent the world knowledge in a structured way. There are various downstream NLP tasks especially knowledge-driven services, such as query answering (Guu, Miller, and Liang 2015; Cui et al. 2019), relation extraction (Mintz et al. 2009; Reiplinger, Wiegand, and Klakow 2014) and dialogue generation (He et al. 2017). However, a significant proportion of KGs are severely incomplete, which constrains their efficacy in numerous tasks. Consequently, this study concentrates on automatic knowledge graph (KG) reasoning, also as known knowledge graph completion (KGC).

Over recent years, embedding-based models (Bordes et al. 2013; Lin et al. 2015) have effectively preserved KG structural information for single-hop reasoning but lack interpretability. To address this, reinforcement learning (RL) frameworks (Xiong, Hoang, and Wang 2017; Das et al. 2018) have been introduced to compose single-hop triplets into multi-hop reasoning chains. Recent advances in deep learning (DL) and RL (Wang et al. 2019; Lv et al. 2020; Nikopensius et al. 2023; Wang et al. 2025) further enhance multi-hop reasoning. For instance, AttnPath (Wang et al. 2019) employs attention mechanisms to guide agents, preventing them from stalling at the same node.

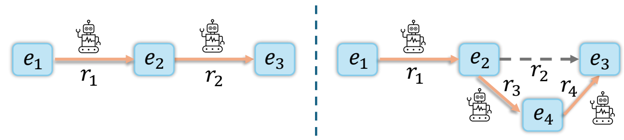

Figure 1: An illustrative example of short direct path and long indirect path. When no short direct path exists, the agent searches for a long indirect path.

Although existing multi-hop reasoning models have achieved impressive results, there is a noteworthy issue with these models that they only perform well when the agent's reasoning path length is short. The drawback that the agent relies heavily on short reasoning chains is fatal in certain datasets, such as sparse KG. Figure 1 is an illustration of short direct path and long indirect path. It's challenging for an agent to identify a short direct path due to the sparsity which makes the relation r_2 do not exist. The optimal path for agent is $e_1 \to e_2 \to e_4 \to e_3$ we called long indirect path instead of $e_1 \to e_2 \to e_3$. From this point, enhancing the agent's long-distance reasoning ability is one solution to alleviate the multi-hop models' poor performance in the sparse KG.

However, the dimension of the discrete action space at each space is large (Das et al. 2018). As the path length increases, the choices the agent faced will grow exponentially. The most relevant to our work is CURL (Zhang et al. 2022) which proposed a dual-agent framework and mutual reinforcement rewards to assign one of the agents (GIANT) searching on cluster-level paths quickly and providing stage-wise hints for another agent (DWARF). Compared to the classic multi-hop model MINERVA (Das et al. 2018), CURL does improve long-distance reasoning ability. But CURL also has two drawbacks, which make CURL's performance

[*]Bin Wang is the corresponding author.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

inconsistent: (1) Mutual reinforcement reward mechanism forces the low-level agent DWARF to adopt similar policies to the high-level agent GIANT, even if GIANT's policies is not well enough. This may result in false-negative rewards to the intermediate actions which are reasonable. (2) At the early phase, DWARF and GIANT adopt a near-random policy which makes low training efficacy. For briefness, we call the high-level agent as GIANT and the low-level agent as DWARF like (Bai et al. 2022).

In light of these challenges, we propose FULORA, a robust dual-agent framework for KG reasoning with taking full advantage of entity embedding and relation embedding in the KG. FULORA seamlessly makes the self-exploration and path-reliance trade-off of DWARF through a supervised learning method. This unique mechanism will enable DWARF to make its own decisions while receiving meaningful guidance from GIANT, instead of relying entirely on the command of GIANT. Moreover, with the aim to make better use of the entity embedding and relation embedding, FULORA introduces attention mechanism and dynamic path feedback to DWARF and GIANT respectively. Intuitively, GIANT searches a feasible path as soon as possible to guide DWARF to reduce the search space, while DWARF adopts diversified exploration policies in the constrictive search space to prevent over-dependence on GIANT. In this way, DWARF walking on the original KG has both excellent long distance reasoning ability and short direct path utilization ability.

Related Work

In line with the focus of our work, we provide a brief overview of the background and related work on knowledge graph reasoning.

Knowledge Graph Embedding

Knowledge graph embedding (KGE) methods map entities to vectors in low-dimensional embedding space, and model relations as transformations between entity embeddings (Bai et al. 2022). Once we map them to low-dimensional dense vector space, we can use modeling methods to perform calculations and reasoning (Ma et al. 2018). Prominent examples include TransE (Bordes et al. 2013), TransR (Lin et al. 2015), ConvE (Dettmers et al. 2018), TuckER (Balazevic, Allen, and Hospedales 2019) and LMKE (Wang et al. 2022), which are equipped with a scoring function that maps any triplet (e_s, r_q, e_o) to a scalar score. While KGE is effective at capturing simpler relationships in the graph, such as first-order adjacency relationships, it struggles with more complex reasoning tasks, particularly those involving multi-hop reasoning (Yao, Mao, and Luo 2019; Wang et al. 2023).

GNN-based Reasoning

Graph neural networks (GNNs) (Scarselli et al. 2008; Veličković et al. 2017; Xu et al. 2018) are a class of models used for representation learning, specifically designed to encode the structural information of graphs. In the context of link prediction, common frameworks often rely on an auto-encoder formulation, where GNNs generate node embeddings, and edges are predicted as a function of node pairs. These frameworks are inductive when node features are provided, but they become transductive when such features are not available. Another set of frameworks, including SEAL (Zhang and Chen 2018) and GraIL (Teru, Denis, and Hamilton 2020), explicitly encode the subgraph surrounding each node pair for link prediction. Recent advancements in this field include works such as NBFNet (Zhu et al. 2021) and RED-GNN (Zhang and Yao 2022). The former solves the path formulation with learned operators in the generalized Bellman-Ford algorithm while the latter makes use of dynamic programming to recursively encodes multiple r-digraphs with shared edges, and utilizes query-dependent attention mechanism to select the strongly correlated edges. However, these methods primarily focus on the structure of the knowledge graph (KG) itself, specifically the graph structure information of the central node, without addressing how to enhance the ability for long-distance reasoning.

Multi-hop Reasoning

The advancement of deep reinforcement learning (DRL) has sparked interest in applying DRL to path-finding tasks. The first significant work combining DRL and KG reasoning is DeepPath (Xiong, Hoang, and Wang 2017), which inspired subsequent models, though it requires prior knowledge of the target entity. In contrast, MINERVA (Das et al. 2018) eliminates this requirement, allowing the agent to traverse the knowledge graph until it finds the target. Recent developments have explored the use of powerful neural networks for generating walking policies. M-Walk (Shen et al. 2018) uses an RNN (Elman 1990) to record agent trajectories and optimize rewards with a Monte Carlo Tree Search (MCTS) (Coulom 2006). GRL (Wang et al. 2020) combines GAN (Goodfellow et al. 2014) and LSTM (Hochreiter 1997) to generate new trajectory sequences, enabling the agent to reason not only within the original graph but also in automatically generated sub-graphs, extending relational paths until target entities are found. HMLS (Zheng et al. 2024) improves the generalizability and effectiveness of multi-hop reasoning in few-shot scenarios by exploiting hard relations and hierarchical relation structures. While these models perform well in short-distance reasoning (path length = 3), their performance in long-distance reasoning tasks remains suboptimal. Unlike previous multi-hop reasoning methods, FULORA focuses more on the agents' long-distance reasoning ability and the significant impact of sparse rewards on training efficiency.

Preliminary

In this section, we commence with the problem definition of our work, followed by the introduction of environment representation.

Problem Definition

We formally define the research problem of this paper in the following part. The knowledge graph is defined as a directed graph $\mathcal{G} = \{\mathcal{E}, \mathcal{R}\}$, where $\mathcal{E}$ is a set of all entities and $\mathcal{R}$ is a

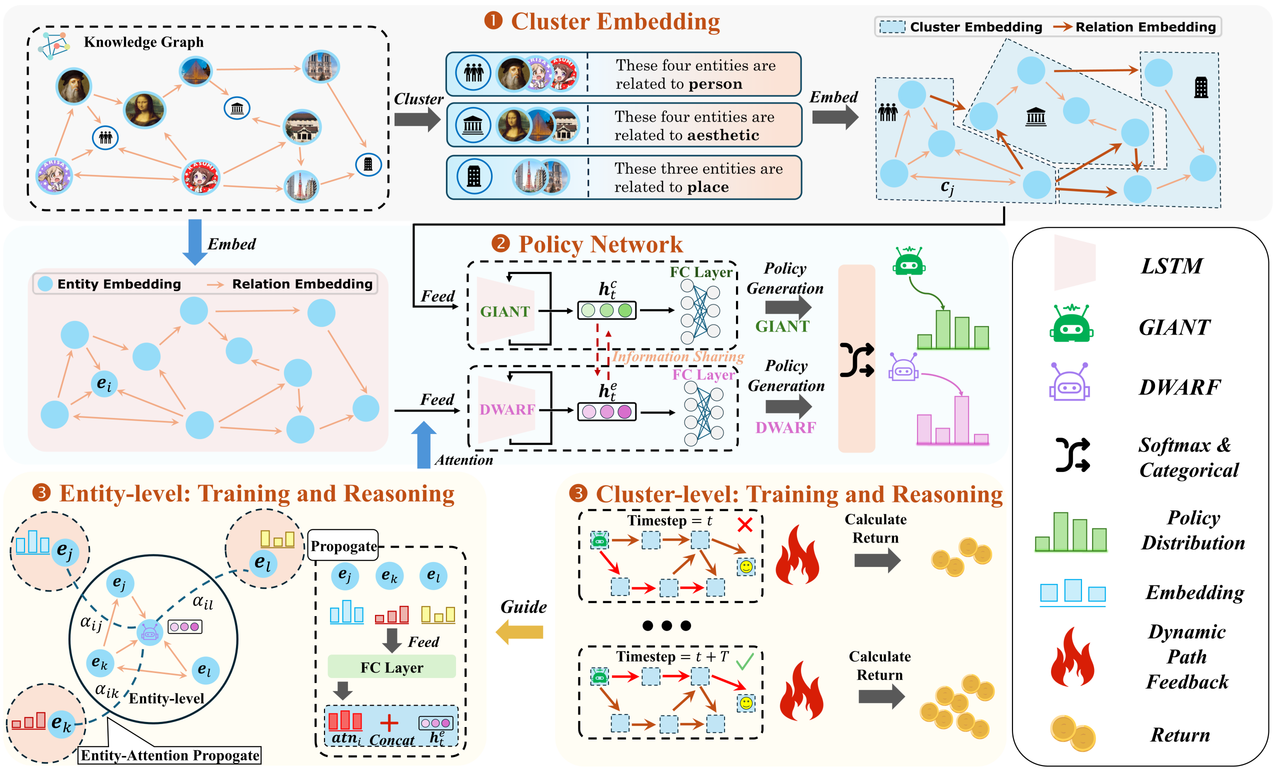

Figure 2: An overview of FULORA framework. ❶ Given a KG $\mathcal{G}$, We first pre-embed the KG using TransE and then apply K-means clustering to generate the cluster-level KG $\mathcal{G}^c$. ❷ We design separate policy networks for GIANT and DWARF, using cluster-level KG $\mathcal{G}^c$ and entity-level KG $\mathcal{G}^e$ as inputs. The hidden states of GIANT and DWARF $\mathbf{h}_t^c$ and $\mathbf{h}_t^e$ share information, facilitating communication. ❸ To enable DWARF to better leverage the KG structure, we apply the graph attention mechanism and feed the resulting attention vector into the policy network. Dynamic Path Feedback alleviates the near-random policy issue caused by sparse rewards in early training phase, allowing GIANT to provide high-quality guidance to DWARF sooner.

set of all relations. A link triplet l consists of the source entity $e_s \in \mathcal{E}$, the target entity $e_o \in \mathcal{E}$ and the relation $r \in \mathcal{R}$, i.e., $l = (e_s, r, e_o)$. In the real world, a link triplet corresponds to a fact tuple. For instance, we can represent the fact **whale is a mammal** as $(whale, is\ a, mammal)$. We follow the definition in prior graph walking models, that is, query answering (Das et al. 2018; Zhang et al. 2022). In the applications such as searching and query answering, most problems are to infer another entity when we only know the source entity e_s and the query relation r_q, which can be formed by an incomplete link triplet $l_m = (e_s, r_q, ?)$, where "?" indicates the target entity e_o is unknown and needs to be found in the KG.

Environment Representation

State. The state consists of two entities—current entity e_t and source entity e_s—and the query relation. The current entity e_t is state-dependent, reflecting the agent's reasoning condition, while the source entity e_s and query relation are shared globally. Formally, $s_t = (e_t, e_s, r_q) \in \mathcal{S}$.
Action. The agent interacts with the environment by selecting an action $a_t \in \mathcal{A}$, corresponding to an outgoing edge of the graph $\mathcal{G}$. Formally, $a_t = \{(r_{t+1}, e_{t+1}) | (e_t, r_{t+1}, e_{t+1}) \in \mathcal{G}\}$. To allow termination, the agent can also stay at the current entity by selecting a self-loop edge.
Transition. A state transition occurs when the agent takes an action. The transition function $\delta : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is defined as $\delta(s_t, a_t)$, which produces a new state. Multi-hop reasoning typically selects a neighboring entity randomly, as entities often have multiple neighbors connected by the same relation (Xiong, Hoang, and Wang 2017; Das et al. 2018).
Reward. After the agent takes an action, if the corresponding entity is the correct target, it receives a reward of 1; otherwise, it receives a reward of 0.

Methodology

In this section, we propose FULORA, an extended dual-agent framework for knowledge graph reasoning via efficient guidance-exploration. As illustrated in Figure 2, FULORA is able to improve the exploration efficiency of GIANT via dynamic feedback mechanism, so as to provide more reliable guidance for DWARF. In a corresponding manner, DWARF employs a supervised learning approach to achieve a balance between guidance and exploration. Subsequently, it aggregates messages emitted by all neighbors of the current entity via an attention mechanism. Next, we will delve into the specifics of each aforementioned components.

Embedding Generation

Consistent with (Zhang et al. 2022), we employ TransE (Bordes et al. 2013) due to its efficiency in encoding the structural proximity information of KG to generate pre-trained entity embeddings, then we divide the original KG into N clusters by utilizing K-means. In order for GIANT can easily walk on the cluster-level graph $\mathcal{G}^c$ while preserving the relation information of the original KG $\mathcal{G}$, we add a link to two clusters if there is at least one entity-level edge between them, which is detailed in Appendix E.

Policy Networks

In our model, we utilize a three-layer LSTM, enabling the agent to memorize and learn from the actions taken before. In contrast to previous models, the necessity arises to design the network separately in this case, given that two agents are walking on the KG. An agent based on LSTM encodes the recursive sequence as a continuous vector $\mathbf{h}_t \in \mathbb{R}^{2d}$. Specifically, the hidden state embedding of GIANT is $\mathbf{h}_t^c$ while the hidden state embedding of DWARF is $\mathbf{h}_t^e$. Their initial hidden state is $\mathbf{0}$. In addition, we define an information sharing vector $\mathbf{I}_t = [\mathbf{h}_t^e; \mathbf{h}_t^c]$ for GIANT and DWARF to share path information. To a certain extent, cluster-level paths are complementary to entity-level paths, as they ensure the sharing of essential path information from GIANT to DWARF.

For GIANT. We denote the current cluster embedding at time step t by $\mathbf{c}_t \in \mathbb{R}^{2d}$. The action representation $\mathbf{a}_t^c$ is given by the cluster embedding itself, i.e., $\mathbf{a}_t^c = \mathbf{c}_t \in \mathbb{R}^{2d}$ because the action corresponds to the next outgoing cluster. The history embedding is updated according to LSTM dynamics:

$$\mathbf{h}_t^c = \text{LSTM}_c(\mathbf{W}^c[\mathbf{h}_{t-1}^c; \mathbf{I}_{t-1}], \mathbf{a}_{t-1}^c), \tag{1}$$

where $\mathbf{W}^c \in \mathbb{R}^{2d \times 6d}$ is a projection matrix to maintain shape.

For DWARF. Unlike GIANT, which operates on cluster-level KGs, DWARF faces greater challenges on the original KG. Entities often have multiple aspects. For instance, a professor may have both professional relations (e.g., *worksForUniversity*) and family relations (e.g., *spouse*). Additionally, cluster-level paths are typically shorter than entity-level paths, requiring DWARF to have enhanced long-distance reasoning. Consequently, DWARF should prioritize relations and neighbors most relevant to the query. Therefore, we integrate the Graph Attention mechanism (Velickovic et al. 2017) into DWARF. We adopt the same approach as AttnPath (Wang et al. 2019) to obtain the attention vector $\mathbf{atn}_{t-1}$. DWARF's history embedding $\mathbf{h}_t^e$ can be obtained from

$$\mathbf{h}_t^e = \text{LSTM}_e(\mathbf{W}^e[\mathbf{h}_{t-1}^e; \mathbf{atn}_{t-1}; \mathbf{I}_{t-1}], \mathbf{a}_{t-1}^e), \tag{2}$$

where $\mathbf{W}^e \in \mathbb{R}^{2d \times 7d}$ is a projection matrix, while the action representation $\mathbf{a}_t^e$ is the concatenation of the relation embedding $\mathbf{r}_t \in \mathbb{R}^d$ and the end node embedding $\mathbf{e}_t \in \mathbb{R}^d$, i.e., $\mathbf{a}_t^e = [\mathbf{r}_t; \mathbf{e}_t] \in \mathbb{R}^{2d}$.

Policy Generation. To predict the next cluster for GIANT and the next entity for DWARF, we apply a two-layer feed-forward network on the concatenation of their last LSTM states and current RL state embeddings,

$$\begin{aligned} &\mathbf{d}_t^c = \text{SoftMax}(\mathbf{A}_t^c \times \mathbf{W}_2^c \text{ReLU}(\mathbf{W}_1^c[\mathbf{c}_t; \mathbf{h}_t^c])), \\ &a_t^c \sim \text{Categorical}(\mathbf{d}_t^c), \end{aligned} \tag{3}$$

$$\begin{aligned} &\mathbf{d}_t^e = \text{SoftMax}(\mathbf{A}_t^e \times \mathbf{W}_2^e \text{ReLU}(\mathbf{W}_1^e[\mathbf{e}_t; \mathbf{r}_q; \mathbf{h}_t^e])), \\ &a_t^e \sim \text{Categorical}(\mathbf{d}_t^e), \end{aligned} \tag{4}$$

where $\mathbf{W}_1^c \in \mathbb{R}^{4d \times 4d}$, $\mathbf{W}_2^c \in \mathbb{R}^{2d \times 4d}$, $\mathbf{W}_1^e \in \mathbb{R}^{4d \times 4d}$, $\mathbf{W}_2^e \in \mathbb{R}^{2d \times 4d}$ are the matrices of learnable weights to maintain dimension of history embedding. While $\mathbf{A}_t^c \in \mathbb{R}^{|A_t^c| \times 2d}$, $\mathbf{A}_t^e \in \mathbb{R}^{|A_t^e| \times 2d}$ represent the embeddings of all next possible actions for GIANT and DWARF respectively.

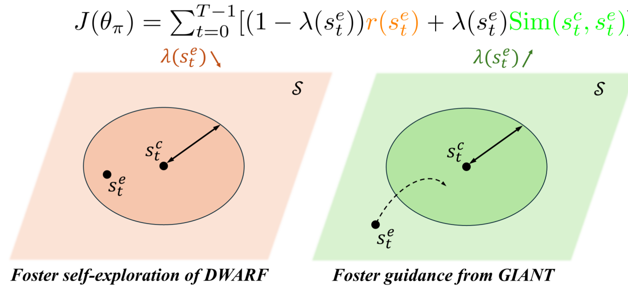

Figure 3: An illustration of Effective Guidance-Exploration. When DWARF is out of bounds, GIANT guides it to move quickly inside. Otherwise, DWARF prefers to explore for itself to find a correct target.

Efficient Guidance-Exploration

As mentioned above, to address the issue of sparse rewards and a large action space, we aim for GIANT to guide DWARF in reducing the action space. However, GIANT's guidance is not always beneficial due to two key issues:

- **Poor guidance.** Poor guidance can lead DWARF to incorrect answers.
- **Policy shift.** A policy which is suitable for GIANT may not be fully applicable to DWARF.

Considering the two issues, we propose an efficient guidance-exploration approach, which gives DWARF a constraint reward to balance guidance and exploration via supervised learning approaches.

Constraint Reward. Considering the constraint, that is, receiving high-quality guidance from GIANT, we introduce a metric for the state similarity between GIANT and DWARF, denoted as $\text{Sim}(s_t^c, s_t^e)$. It is calculated as the cosine similarity between the pre-trained embeddings of the current cluster[1] and the current entity:

$$\text{Sim}(s_t^c, s_t^e) = \frac{\mathbf{c}_t^\top \mathbf{e}_t}{||\mathbf{c}_t||_2 ||\mathbf{e}_t||_2}. \tag{5}$$

[1]Each cluster embedding is obtained by averaging all entity embeddings within it. A slight abuse of notation indicates that the $\mathbf{c}_t$ here is not the $\mathbf{c}_t$ mentioned in the policy network. In the concrete implementation, we treat the $\mathbf{c}_t$ in the policy network as a self-cascade of the $\mathbf{c}_t$ here.

We solve the optimization problem by the following formulas,

$$\max \mathbb{E}_{a_1^e,\ldots,a_T^e \sim \pi_\theta^e}\left[\sum_{t=0}^{T-1} r_e(s_t^e|s_0^e)\right]$$
$$\text{s.t. } \mathrm{Sim}(s_t^c, s_t^e) \geq \frac{\delta}{r_c(s_t^c)+\varepsilon} \tag{6}$$

where $r_e(s_t^e)$ and $r_c(s_t^c)$ are default rewards for DWARF and GIANT respectively, the agent obtains a favorable reward 1 if the corresponding entity or cluster is a correct target and unfavorable reward 0 otherwise. It is only when GIANT reaches the correct cluster that the constraint in Equation 6 is operative. In practice, we set $\varepsilon = -0.01\delta$.

Practical Algorithm. The objective in Equation 6 can be optimized with any reinforcement learning algorithm that implements generalized policy iteration. Here we use REINFORCE (Williams 1992) and the method of Lagrange multipliers as described by (Abdolmaleki et al. 2018; Grillotti et al. 2024). For all DWARF state s_t^e, we maximize the Lagrangian function, subject to $0 \leq \lambda(s_t^e) \leq 1$,

$$J(\theta_{\pi^e}) = \sum_{t=0}^{T-1}[(1-\lambda(s_t^e))r_e(s_t^e) + \lambda(s_t^e)\mathrm{Sim}(s_t^c, s_t^e)], \tag{7}$$

the Lagrange multiplier is updated to make the guidance-exploration trade-off. Figure 3 gives an illustration: When the state similarity metric between GIANT and DWARF $\mathrm{Sim}(s_t^c, s_t^e)$ is less than the threshold, the parameter θ_λ are optimized so that $\lambda(s_t^e)$ increases to encourage GIANT to provide more guidance for DWARF. Conversely, the parameter θ_λ are optimized so that $\lambda(s_t^e)$ decreases to encourage DWARF to explore in the constrictive space when the state similarity metric between GIANT and DWARF $\mathrm{Sim}(s_t^c, s_t^e)$ is greater than the threshold. In practice, we utilize a cross-entropy loss to optimize θ_λ:

$$J(\theta_\lambda) = \sum_{t=0}^{T-1}[-(1-y)\log(1-\lambda(s_t^e)) - y\log(\lambda(s_t^e))]$$
$$\text{where } y = \begin{cases} 0 & \text{if } \mathrm{Sim}(s_t^c, s_t^e) \geq \dfrac{\delta}{r_c(s_t^c)+\varepsilon} \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

Dynamic Path Feedback

A further challenge is to enhance the search efficiency of GIANT in order to provide DWARF with the requisite guidance as expeditiously as possible, given that the search is limited to a fixed number of step T. In the default reward, GIANT will receive 1 reward only when it reaches the correct cluster, and the rest will be 0, which makes GIANT adopt random policy at the early phase. This phenomenon is not conducive to stable learning outcomes for two agents. On the basis of the theory of reward shaping (Harutyunyan et al. 2015), we rewrite the reward function of GIANT, named dynamic path feedback. In particular, we write it in the form of an objective function $J(\theta_{\pi^c})$,

$$J(\theta_{\pi^c}) = \sum_{t=0}^{T-1}[r_c(s_t^c) - \alpha\Delta(s_t^c, s_{t+1}^c)],$$
$$\Delta(s_t^c, s_{t+1}^c) = \mathrm{Sim}(s_t^c, s_{\text{target}}^c) - \mathrm{Sim}(s_{t+1}^c, s_{\text{target}}^c) \tag{9}$$

s_{t+1}^c comes from the next state generated by the policy network. In contrast to the default reward, dynamic path feedback uses the reward function to score the GIANT's path in a rollout rather than simply identifying whether it has reached the correct target cluster. Even if GIANT does not reach the correct target cluster in a rollout, it will evaluate the quality of the path, thereby accelerating the learning process. In Appendix D, we prove that GIANT learns optimal policy in dynamic path feedback is consistent with default rewards circumstance:

Theorem 1 (*Consistency of optimal policy*) *Given two MDPs that differ only in reward function, denoted as $M = (\mathcal{S}, \mathcal{A}, \delta, \mathcal{R})$ and $M' = (\mathcal{S}, \mathcal{A}, \delta, \mathcal{R}_D)$ respectively, where $\mathcal{R} = r_c(s_t^c)$ is the default reward while $\mathcal{R}_D = r_c(s_t^c) - \alpha\Delta(s_t^c, s_{t+1}^c)$ is the dynamic path feedback reward. Their optimal policies are consistent, that is,*

$$\pi_M^*(s_t^c) = \pi_{M'}^*(s_t^c).$$

Dataset	#Ent	#Rel	#Fact	#Que	#Mean	#Med
NELL-995	75,492	200	154,213	3,992	4.07	1
WN18RR	40,945	11	86,835	3,134	2.19	2
FB15K-237	14,505	237	272,115	20,466	19.74	14

Table 1: The statistics of some benchmark KG datasets. #Mean is the averaged outgoing degree of every entity that can indicate the sparsity level while #Med is the corresponding median.

Experiments

In this section, we evaluate the efficacy of FULORA on three real-world KG datasets: NELL-995 (Xiong, Hoang, and Wang 2017), WN18RR (Dettmers et al. 2018) and FB15K-237 (Toutanova et al. 2015). The datasets statistics are listed in Table 1.

As can be seen from the statistical indicators, these three datasets represent standard KG, sparse KG and Dense KG respectively. In our selection of the baseline, we are not only evaluating it against the state-of-the-art multi-hop reasoning methods, but also against with other embedding-based KG reasoning methods and GNN-based reasoning methods. FULORA and all baselines are implemented under the Pytorch framework and run on the NVIDIA 3080Ti GPU. All the results are the average of the results in five experiments. See Appendix A.1 for a brief introduction of each baseline.

Short-distance Reasoning Ability

For each triplet (e_s, r_q, e_o) in the test set, we convert it to a query $(e_s, r_q, ?)$ and use embedding-based or multi-hop

Model	NELL-995				WN18RR				FB15K-237			
	MRR	@1	@3	@10	MRR	@1	@3	@10	MRR	@1	@3	@10
TransE (Bordes et al. 2013)	51.4	45.6	67.8	75.1	35.9	28.9	46.4	53.4	36.1	24.8	40.1	45.0
DistMult (Yang et al. 2015)	68.0	61.0	73.3	79.5	43.3	41.0	44.1	47.5	37.0	27.5	41.7	56.8
ComplEx (Trouillon et al. 2016)	68.4	61.2	76.1	82.1	41.5	38.2	43.3	48.0	39.4	30.3	43.4	57.2
LMKE (Wang et al. 2022)	<u>74.6</u>	71.7	84.7	89.5	<u>53.6</u>	45.1	66.0	79.4	<u>41.2</u>	31.8	46.2	56.9
NBFNet (Zhu et al. 2021)	70.9	68.0	76.8	80.4	<u>48.1</u>	44.9	49.8	58.3	<u>40.5</u>	30.8	45.8	55.2
RED-GNN (Zhang and Yao 2022)	<u>71.2</u>	67.8	-	86.2	46.9	42.5	-	53.0	39.8	29.9	-	54.4
MINERVA (Das et al. 2018)	67.5	58.8	74.6	81.3	44.8	41.3	45.6	51.3	27.1	19.2	30.7	42.6
AttnPath (Wang et al. 2019)	69.3	62.7	73.9	80.1	42.9	40.7	44.3	52.9	31.9	24.1	40.4	43.8
SQUIRE (Bai et al. 2022)	71.1	68.2	81.4	87.2	48.2	45.0	51.4	59.7	35.0	26.5	41.7	50.3
CURL (Zhang et al. 2022)	70.8	66.7	78.6	84.3	46.0	42.9	47.1	52.3	30.6	23.9	38.1	50.9
HMLS (Zheng et al. 2024)	71.8	69.0	80.9	88.9	48.5	43.9	52.9	60.4	**37.4**	28.0	42.1	52.5
FULORA (Ours)	**72.5**	69.4	79.7	89.2	**49.1**	45.6	50.7	59.2	36.4	27.1	41.9	51.3

Table 2: Link prediction results with a **path length of 3** on the NELL, WordNet, and Freebase datasets (embedding-based and GNN-based reasoning do not inherently consider path length, so we exclude triples reachable by shorter paths for fair comparison). All metrics are multiplied by 100. The best score of embedding-based reasoning models, GNN-based models are <u>underlined</u> while multi-hop reasoning models are in **bold**. Compared to other indicators, we specifically highlight the MRR to emphasize its significance.

models with a beam search width of 50 to rank the tail entities. Following (Bordes et al. 2013), we evaluate using two metrics: (1) mean reciprocal rank (MRR) and (2) Hits@K, the proportion of correct tail entities ranked in the top K.

As shown in Table 2, we present the performance of FULORA and all baselines on NELL-995, WN18RR, and FB15K-237. On both the standard KG (NELL-995) and sparse KG (WN18RR), FULORA not only outperforms CURL, SQUIRE and HMLS currently regarded as the most powerful multi-hop reasoning models, but also significantly surpasses NBFNet and RED-GNN, both of which are recognized as advanced GNN-based reasoning algorithms. FULORA also achieves comparable performance to the best embedding-based model, LMKE. On FB15K-237, where 1-to-M relations dominate, multi-hop models often struggle with high-degree nodes, hindering correct entity retrieval. In contrast, FULORA's attention mechanism focuses on the most relevant neighbors. Averaging across three datasets, FULORA improves MRR and Hits@1, 3, and 10 by 3.5%, 2.9%, 2.8%, and 4.1% over CURL. **Notably, Table 2 presents results based on a path length of 3, which does not capture FULORA's exceptional performance in long-distance reasoning.** Subsequent experiments focus on the performance of FULORA's components, GIANT and DWARF, in long-distance reasoning, with comparisons to other advanced models in Table 2. In fact prediction, FULORA outperforms other multi-hop baselines, as detailed in Appendix B.1.

Long-distance Reasoning Ability

Recall the motivation of FULORA, we tend to address the issue of multi-hop reasoning models being unable to infer correct answer due to the lack of short direct path by improving long-distance reasoning ability. This issue is significant on standard KG and sparse KG, so we conduct the following experiments on NELL-995 and WN18RR. For effective evaluation, we compare our model with MINERVA and CURL in NELL-995 and WN18RR, where we remove the most frequently-visited short paths found by the bi-directional search (Xiong, Hoang, and Wang 2017; Zhang et al. 2022) inside KGs.

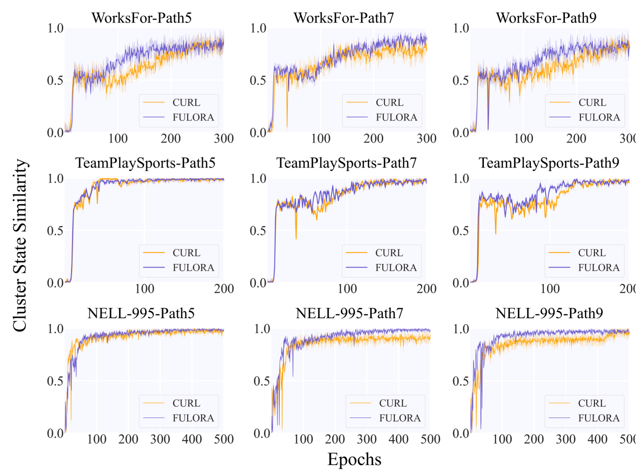

Figure 4: Learning curves comparing the performance of ours against CURL from NELL-995 relation tasks and all tasks. Averaged over 5 seeds with the shaded area showing standard deviation. Our proposed model is significantly better than CURL in both score and stability.

Cluster State Similarity. One of the most significant contributions of GIANT is the implementation of dynamic path feedback, which has been shown to enhance the learning efficiency. To visually demonstrate the efficacy of dynamic path feedback, we initially focus on GIANT's

NELL-995/WN18RR/FB15K-237	MRR	@1	@10
FULORA + TransE (Bordes et al. 2013)	72.5/49.1/36.4	69.4/45.6/27.1	89.2/57.2/51.3
FULORA + LMKE (Wang et al. 2022)	76.8/54.4/38.6	73.0/46.3/30.2	93.7/72.4/51.2

Table 3: Effects of various pre-embedding methods on the performance of FULORA.

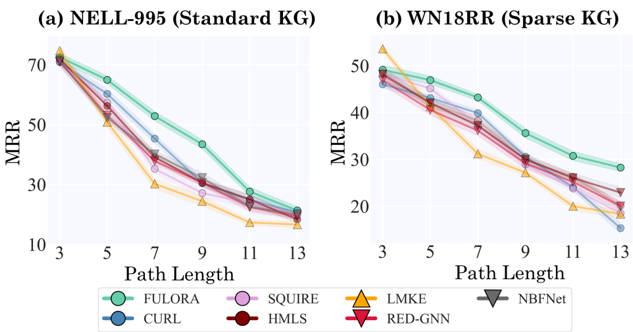

Figure 5: The long-distance performance: FULORA significantly outpeforms CURL, SQUIRE, HMLS, LMKE, RED-GNN, NBFNet on NELL-995 (standrad KG) and WN18RR (Sparse KG).

reasoning ability on cluster-level, which directly impacts DWARF's reasoning results. Here, we record the cluster state similarity (CSS) $\text{Sim}(s_T^c, s_{\text{target}}^c)$ under each epoch. As shown in Figure 4, our model outperforms CURL in scores and stability in most cases. See Appendix B.3 for a comparison of the remaining tasks.

Entity Reasoning Accuracy. we now demonstrate that FULORA's efficient guidance-exploration enhances DWARF's performance in long-distance reasoning compared to other well-performed baselines (Appendix B.4.2 provides a further analysis). Figure 5 shows the MRR for varying path lengths on NELL-995 and WN18RR. FULORA excels in long-distance reasoning accuracy on WN18RR (sparse KG). This highlights our motivation: FULORA mitigates the poor performance of current KG reasoning methods in the absence of short direct paths by enhancing long-distance reasoning. Overall, FULORA demonstrates more robust long-distance reasoning performance with minimal degradation in long-path settings. This is due to the dynamic path feedback and efficient guidance-exploration, which enable better information sharing between GIANT and DWARF, ensuring DWARF's excellent self-exploration ability.

Ablation Study

Sensitivity Analysis. We utilize δ and α to control the degree of efficient guidance-exploration and dynamic path feedback. To comprehensively explore the efficacy of the two mechanisms, we conduct link prediction on NELL-995 and WN18RR by varying the values of δ and α as $\{0.20, 0.30, 0.40, 0.50\}$ and $\{0.05, 0.10, 0.15, 0.20\}$, respectively. The averaged results are shown in Figure 6. In accordance with our previous analysis, it is evident that

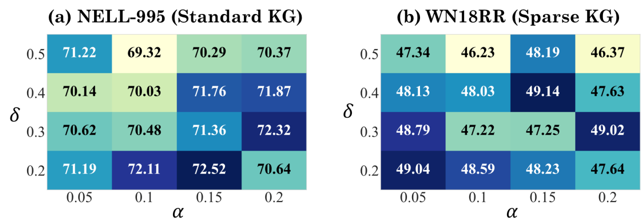

Figure 6: Ablation analysis w.r.t δ and α on NELL-995 and WN18RR, respectively. Here we report MRR results with path length=3 for both datasets.

DWARF cannot rely excessively on GIANT for guidance. Furthermore, the level of path feedback that GIANT receives should not be unduly strong. In addition, in Appendix B.4, we have carefully discussed the influence of three main components of FULORA on KG reasoning efficiency.

Pre-embedding Methods. In our approach, we use TransE (Bordes et al. 2013) for pre-training the embeddings. One straightforward idea is to improve FULORA's performance by using a more advanced KG embedding model such as LMKE (Wang et al. 2022). Table 3 presents the performance of TransE and LMKE as pre-embedding methods. The more advanced LMKE indeed improves FULORA's performance. However, we use TransE in our experiments to emphasize that the performance improvement is due to FULORA's efficient guidance-exploration mechanism, not the strength of the KG embedding method.

Conclusion

We present FULORA, an efficient guidance-exploration model built on dual-agent KG reasoning framework to enhance the agent's long-distance reasoning ability on standard KG and sparse KG. The key insight behind our approach is balancing the self-exploration of DWARF and the guidance from GIANT. Specifically, on the one hand, we leverage the attention mechanism to make DWARF pay attention to the neighbouring entities that are close to the query. On the other hand, we propose that dynamic path feedback enables GIANT to have better learning efficiency, thus providing DWARF with high-quality guidance, making the DWARF to have a favourable global vision while having excellent local reasoning ability. Experiments on three real-world datasets demonstrate that FULORA outperforms state-of-the-art multi-hop reasoning methods. Further analysis reveals that FULORA's long-distance reasoning ability on standard and sparse KGs significantly outperforms current KG reasoning methods.

References

Abdolmaleki, A.; Springenberg, J. T.; Tassa, Y.; Munos, R.; Heess, N.; and Riedmiller, M. A. 2018. Maximum a Posteriori Policy Optimisation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Bai, Y.; Lv, X.; Li, J.; Hou, L.; Qu, Y.; Dai, Z.; and Xiong, F. 2022. SQUIRE: A Sequence-to-sequence Framework for Multi-hop Knowledge Graph Reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 1649–1662.

Balazevic, I.; Allen, C.; and Hospedales, T. M. 2019. TuckER: Tensor Factorization for Knowledge Graph Completion. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 5184–5193. Association for Computational Linguistics.

Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; and Yakhnenko, O. 2013. Translating Embeddings for Modeling Multi-relational Data. In Burges, C. J. C.; Bottou, L.; Ghahramani, Z.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, 2787–2795.

Coulom, R. 2006. Efficient selectivity and backup operators in Monte-Carlo tree search. In *International conference on computers and games*, 72–83. Springer.

Cui, W.; Xiao, Y.; Wang, H.; Song, Y.; Hwang, S.; and Wang, W. 2019. KBQA: Learning Question Answering over QA Corpora and Knowledge Bases. *CoRR*.

Das, R.; Dhuliawala, S.; Zaheer, M.; Vilnis, L.; Durugkar, I.; Krishnamurthy, A.; Smola, A.; and McCallum, A. 2018. Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases using Reinforcement Learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018. Convolutional 2D Knowledge Graph Embeddings. In McIlraith, S. A.; and Weinberger, K. Q., eds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 1811–1818. AAAI Press.

Elman, J. L. 1990. Finding structure in time. *Cognitive science*, 14(2): 179–211.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Grillotti, L.; Faldor, M.; León, B. G.; and Cully, A. 2024. Quality-Diversity Actor-Critic: Learning High-Performing and Diverse Behaviors via Value and Successor Features Critics. *CoRR*.

Guu, K.; Miller, J.; and Liang, P. 2015. Traversing knowledge graphs in vector space. *arXiv preprint arXiv:1506.01094*.

Harutyunyan, A.; Devlin, S.; Vrancx, P.; and Nowé, A. 2015. Expressing Arbitrary Reward Functions as Potential-Based Advice. In Bonet, B.; and Koenig, S., eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, 2652–2658. AAAI Press.

He, H.; Balakrishnan, A.; Eric, M.; and Liang, P. 2017. Learning Symmetric Collaborative Dialogue Agents with Dynamic Knowledge Graph Embeddings. In Barzilay, R.; and Kan, M., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 1766–1776. Association for Computational Linguistics.

Hochreiter, S. 1997. Long Short-term Memory. *Neural Computation MIT-Press*.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In Bonet, B.; and Koenig, S., eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, 2181–2187. AAAI Press.

Lv, X.; Han, X.; Hou, L.; Li, J.; Liu, Z.; Zhang, W.; Zhang, Y.; Kong, H.; and Wu, S. 2020. Dynamic Anticipation and Completion for Multi-Hop Reasoning over Sparse Knowledge Graph. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 5694–5703. Association for Computational Linguistics.

Ma, T.; Shao, W.; Hao, Y.; and Cao, J. 2018. Graph classification based on graph set reconstruction and graph kernel feature reduction. *Neurocomputing*, 296: 33–45.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In Su, K.; Su, J.; and Wiebe, J., eds., *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, 1003–1011. The Association for Computer Linguistics.

Nikopensius, G.; Mayank, M.; Phukan, O. C.; and Sharma, R. 2023. Reinforcement Learning-based Knowledge Graph Reasoning for Explainable Fact-checking. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2023, Kusadasi, Turkey, November 6-9, 2023*, 164–170. ACM.

Reiplinger, M.; Wiegand, M.; and Klakow, D. 2014. Relation Extraction for the Food Domain without Labeled Training Data - Is Distant Supervision the Best Solution? In Przepiórkowski, A.; and Ogrodniczuk, M., eds., *Advances in Natural Language Processing - 9th International Conference on NLP, PolTAL 2014, Warsaw, Poland, September 17-19, 2014. Proceedings*, volume 8686 of *Lecture Notes in Computer Science*, 345–357. Springer.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1): 61–80.

Shen, Y.; Chen, J.; Huang, P.; Guo, Y.; and Gao, J. 2018. M-Walk: Learning to Walk over Graphs using Monte Carlo Tree Search. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 6787–6798.

Teru, K.; Denis, E.; and Hamilton, W. 2020. Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning*, 9448–9457. PMLR.

Toutanova, K.; Chen, D.; Pantel, P.; Poon, H.; Choudhury, P.; and Gamon, M. 2015. Representing Text for Joint Embedding of Text and Knowledge Bases. In Màrquez, L.; Callison-Burch, C.; Su, J.; Pighin, D.; and Marton, Y., eds., *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 1499–1509. The Association for Computational Linguistics.

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2016. Complex Embeddings for Simple Link Prediction. In Balcan, M.; and Weinberger, K. Q., eds., *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, 2071–2080. JMLR.org.

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

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2017. Graph Attention Networks. *CoRR*, abs/1710.10903.

Wang, H.; Li, S.; Pan, R.; and Mao, M. 2019. Incorporating Graph Attention Mechanism into Knowledge Graph Reasoning Based on Deep Reinforcement Learning. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 2623–2631. Association for Computational Linguistics.

Wang, Q.; Ji, Y.; Hao, Y.; and Cao, J. 2020. GRL: Knowledge graph completion with GAN-based reinforcement learning. *Knowl. Based Syst.*, 209: 106421.

Wang, X.; He, Q.; Liang, J.; and Xiao, Y. 2022. Language Models as Knowledge Embeddings. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, 2291–2297. ijcai.org.

Wang, Y.; Ouyang, X.; Guo, D.; and Zhu, X. 2023. MEGA: Meta-graph augmented pre-training model for knowledge graph completion. *ACM Transactions on Knowledge Discovery from Data*, 18(1): 1–24.

Wang, Z.; Wang, B.; Dou, H.; and Liu, Z. 2025. Windows deep transformer Q-networks: an extended variance reduction architecture for partially observable reinforcement learning. *Applied Intelligence*, 55(1): 35.

Williams, R. J. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Mach. Learn.*, 229–256.

Xiong, W.; Hoang, T.; and Wang, W. Y. 2017. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. In Palmer, M.; Hwa, R.; and Riedel, S., eds., *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 564–573. Association for Computational Linguistics.

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

Yang, B.; Yih, W.; He, X.; Gao, J.; and Deng, L. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yao, L.; Mao, C.; and Luo, Y. 2019. KG-BERT: BERT for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Zhang, D.; Yuan, Z.; Liu, H.; Lin, X.; and Xiong, H. 2022. Learning to Walk with Dual Agents for Knowledge Graph Reasoning. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, 5932–5941. AAAI Press.

Zhang, M.; and Chen, Y. 2018. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31.

Zhang, Y.; and Yao, Q. 2022. Knowledge graph reasoning with relational digraph. In *Proceedings of the ACM web conference 2022*, 912–924.

Zheng, S.; Chen, W.; Wang, W.; Zhao, P.; Yin, H.; and Zhao, L. 2024. Multi-Hop Knowledge Graph Reasoning in Few-Shot Scenarios. *IEEE Trans. Knowl. Data Eng.*, 36(4): 1713–1727.

Zhu, Z.; Zhang, Z.; Xhonneux, L.-P.; and Tang, J. 2021. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems*, 34: 29476–29490.