Evaluating and Enhancing Large Language Models for Conversational Reasoning on Knowledge Graphs

Anonymous ACL submission

Abstract

The development of large language models (LLMs) has been catalyzed by advancements in pre-training techniques. These models have demonstrated robust reasoning capabilities through manually designed prompts. In this work, we evaluate the conversational reasoning capabilities of the current state-of-the-art LLM (GPT-4) on knowledge graphs (KGs). However, the performance of LLMs is constrained due to a lack of KG environment awareness and the difficulties in developing effective optimization mechanisms for intermediary reasoning stages. We further introduce LLM-ARK, a LLM grounded KG reasoning agent designed to deliver precise and adaptable predictions on KG paths. LLM-ARK leverages Full Textual Environment (FTE) prompt to assimilate state information within each reasoning step. We reframe the challenge of multi-hop reasoning on the KG as a sequential decision-making task. Utilizing the Proximal Policy Optimization (PPO) online policy gradient reinforcement learning algorithm, our model is optimized to learn from rich reward signals. Additionally, we conduct an evaluation of our model and GPT-4 on the OpenDialKG dataset. The experimental results reveal that LLaMA-2-7B-ARK outperforms the current state-of-the-art model by 5.28 percentage points, with a performance rate of 36.39% on the target@1 evaluation metric. Meanwhile, GPT-4 scored 14.91%, further demonstrating the effectiveness of our method.

1 Introduction

With significant progress in large language models (LLMs), researchers have recognized their superior capabilities in the field of natural language processing (NLP)(Liu et al., 2023; Shakarian et al., 2023; Lai et al., 2023). Reasoning ability stands as the most demonstrative of AI intelligence. Recently, to boost the performance of LLMs in reasoning tasks, we noted various optimization strategies adopted by researchers such as Chain of Thought (COT)(Wei et al., 2023) and decomposing subtasks(Kazemi et al., 2023). Currently, the reasoning method of LLMs have received limited attention in the conversational KG reasoning task. This research aims to address this gap in the field.

Knowledge Graphs composed of vertices or entities connected by edges or relations, gaining popularity in knowledge-based dialogue systems for its structured disposition. Conversational reasoning models are able to traverse the KG based on conversational context to introduce diverse entities and attributes to make replies more engaging, as well as to improve the logic of the model to mitigate illusions(Rawte et al., 2023; Dong et al., 2022). Previous work(Moon et al., 2019; Zhang et al., 2020; Ni et al., 2022; Tuan et al., 2022) mainly relied on supervised learning methods. To assess the capabilities of current SOTA LLM: GPT-4(OpenAI, 2023), we initially examine the proficiency of LLMs on KG reasoning, as illustrated in Figure 1, determining their potential application in the KG domain. Empirical studies reveal that, despite demonstrating reasonable performance on KG tasks, indicative of their proficiency in managing complex problems, understanding contextual relationships, and utilizing pre-training knowledge, LLMs still present issues and fall short when compared with state-of-the-art models.

There are two main challenges applying LLM-based agents. On the one hand, LLMs suffer from a limited perception of variable reasoning environments. The alignment between LLMs' knowledge and the environment can be wrong and limit functional competence due to lack of grounding(Carta et al., 2023). If properly grounded, the model's structure would be both simplified and effective. For KG reasoning tasks, as shown in Figure 1, the agent achieved better scores when provided with as much information about the environment as possible, such as dialog history, inference path history,

Figure 1: Manual prompts on the OpenDialKG dataset. Compare to GPT-4-Standard, GPT-4-Normal has more awareness of dialog context and path history, while GPT-4-OPA has more awareness of 2-hop exit path subgraphs compared to the former two. The experimental results show that the more environmental information GPT-4 perceives, the higher the knowledge graph reasoning path@1 evaluation metric score.

and all exit paths. Although LLMs are not designed to take actions, Peng et al. (2023); Carta et al. (2023); Du et al. (2023) found that it can be achieved good results in downstream decision making by simply feeding full textual representations as inputs to LLMs.

On the other hand, Yao et al. (2023) indicate that there is a lack of systematic methods for consistent model refinement. In essence, LLMs fall short in possessing essential mechanisms for optimizing intermediary reasoning processes in multi-hop reasoning tasks. This is mainly attributed to the fact that manual prompt tuning is widely used in many application scenarios. It has been observed that LLM-based agents can easily fall into infinite loops if state is not handled properly, and inevitably run into prompt length problems when the trajectory becomes longer. In addition, the design of prompt is also a challenge because an entity may have more than 100 exit edges, all of which are formatted into prompt which is impractical in a knowledge graph environment. LLMs often encounter these issues because they are not designed or trained for action-agent applications.

We introduce **LLM-ARK**, an effective framework that employs **LLM** as an **Agent** for **R**easoning on **K**nowledge Graphs. We employ LLMs as agent and express the Large model KG inference task as a reinforcement learning sequential decision-making problem, and using a Full-Textual-Environment prompt to aggregate multiscale inputs. Moreover, our agent architecture does not necessitate access to LLM parameters or gradient propagation through it. Instead, we adopt a policy gradient approach where the Actor LLM functions as part of the environment. This configuration enables the model to learn from diverse reward signals across varied tasks and environments. In summary, our contributions are as follows:

- We assess the capabilities of state-of-the-art LLM: GPT-4, on large-scale KG inference datasets and analyze the experimental results in detail to understand the causes of their inferior performance.
- To enhance the performance of the LLM agents, we introduce LLM-ARK. Our method expresses the KG dialog inference problem as a reinforcement learning sequential decision-making issue, using a Full-Textual-Environment prompt to aggregate multiscale inputs, dual-environment sensing on the state and decision side and leverage LLMs to explore on KGs.
- Furthermore, we update only the parameters of the PA-MLP that are part of the our agent using the policy gradient method, freezing the parameters of the LLM. This approach enables learning from diverse reward signals during interactions with the environment and improves the efficiency of training.

2 Related Work

2.1 KG Reasoning on Dialog Systems

Given its structured nature, Knowledge Graphs are becoming an increasingly popular external information source in knowledge-based systems. Moon et al. (2019) developed a retrieval system designed to generate responses based on a graph reasoning task. They employed a graph walker to navigate the graph, propelled by the symbol transformation conditions of the dialog context. Jung et al. (2020) utilizing graph attention techniques to navigate the conditional graph of a conversation within a KG dialogue system. The model computes an incoming attention fow to represent entities and an outgoing attention fow to select KG paths. However, this approach cannot be extended to long KG path prediction due to the exponential increase in computational complexity. Ni et al. (2022) introduced a hierarchical reinforcement learning KG inference model that aggregates multiple inputs utilizing an attention mechanism. This approach instructs the model to reason in one step and then fine-tunes it using a goal-directed reinforcement learning. Tuan et al. (2022) employed a single transformer model that walks directly over large-scale KGs, reasoning over fine-tunable KGs to generate responses. Similarly, Luo et al. (2023) initially create relational paths derived from KGs as high-confidence plans, which are later utilized to extract valid reasoning paths from KGs for confident reasoning. Sun et al. (2023) leverage KGs to augment LLMs for deep and responsible reasoning. The framework explores and infers by identifying entities relevant to a given question and retrieving relevant triples from external KGs. This iterative process generates multiple inference paths until enough information is gathered to answer the question or maximum depth is reached.

2.2 LLMs with Reinforcement Learning

Reinforcement learning and large models are divided into two main aspects of the combination, the first aspect further improves the ability of LLM to understand and follow user instructions through reinforcement learning based methods(Ouyang et al., 2022). Yao et al. (2023) employed a special RLHF technique to tailor the model to human preferences, generating beneficial, non-toxic, and safe data for training while also training reward models to evaluate LLMs. Retroformer, a significant improvement over Chain of Thought (COT), is primarily applied to reasoning tasks and uses a unique RLHF method. Shinn et al. (2023) introduce a novel framework, Reflexion, that strengthens linguistic agents through linguistic feedback, rather than updating weights. The Reflexion agent verbally reflects on task feedback signals, and then stores its reflection text in an episodic memory buffer to make better decisions in subsequent trials.

The second aspect is to further improve the applicability of LLM on real-world tasks through a reinforcement learning-based approach, since training a LLM public NLP task/dataset can only cover a small portion of the real world, and reinforcement learning can train LLM-based intelligences to explore the realization of various real-world goals. Carta et al. (2023) studied LLMs interaction with physical environments. Using an interactive textual environment designed to study a series of spatial and navigational tasks and using online reinforcement learning to improve its performance to solve goals. Huang et al. (2022) consistently integrate feedback from diverse sources into the planning language cues of the LLM, thereby enabling it to reason and replan to solve complex problems in both simulated and real-world environments. Singh et al. (2022) propose a procedural LLM hint structure that facilitates plan generation functionality in contextual environments, robot capabilities, and tasks.

3 Methods

3.1 Overview

As shown in Figure 2, our model has the following main components, FTE (Full-Textual-Environment), LLM (Large Language Model) and RL(Reinforcement Learning). FTE can be seen as state manager, using a Full-Textual-Environment prompt to aggregate multi-scale inputs, updating and maintaining state transfers between itself and the environment. At first LLMs obtain a richly informative representation of state embeddings. To capture the path embedding information of the KG, we pre-train the KG on TransE(Fan et al., 2014). Rather than directly introducing the probability distribution of the action space, our Actor feeds the probability distribution along with the path embedding, subsequently eliminating invalid paths (we utilise 'Pad' for this adaptation process) before outputting a precise and legitimate action. We formulate the large model KG inference task within an online reinforcement learning framework and continuously optimize the decision network based on the collected experience in replay buffer. Finally, we refine the adapter using the Proximal Policy Optimization (PPO) online reinforcement learning method. In this section, we will first describe the method used to evaluate GPT-4, and then present each of the modules of our model in turn.

3.2 Manual Prompt Tuning

As illustrated in Figure 1, we detail the prompting schemes, encompassing the standard prompt, normal prompt and out path aware prompt. To guide LLMs in performing specific dialogue tasks, we can formulate the standard prompt and normal

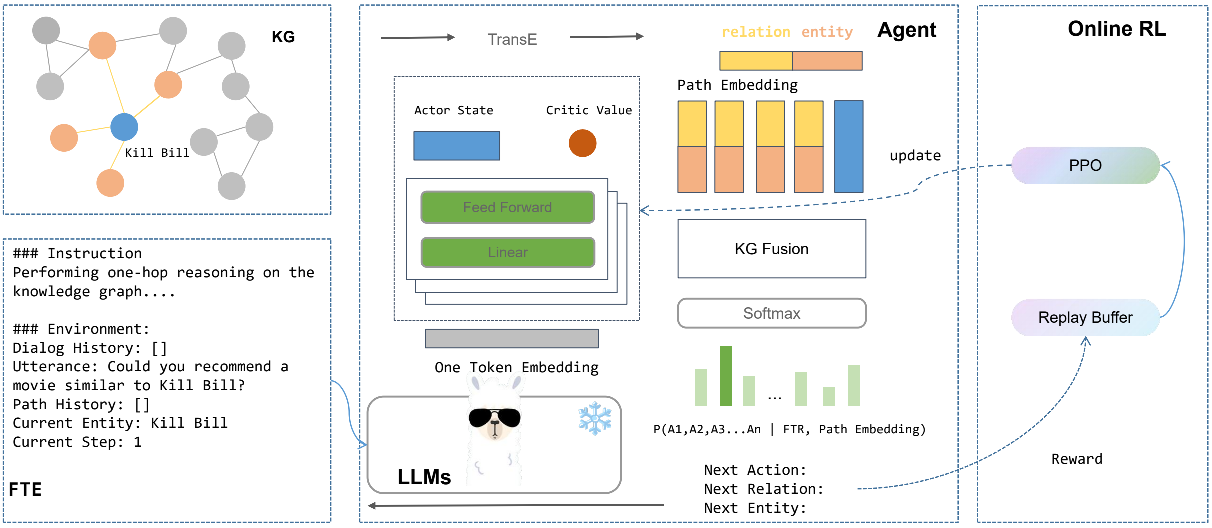

Figure 2: The overall architecture of LLM-ARK.

prompt scheme as:

$$p(r|D,C) \tag{1}$$

Given the task background D and the conversation history C, instruct the LLM to generate the response r. More complex path aware prompt aims to provide alternative options for LLMs to decide what kinds of actions should be taken in the response, instead of simply responding to the instruction. It can be formulated as:

$$p(a,r|D,C,A) \tag{2}$$

Given the task background D, conversation history C, and a set of potential dialogue acts A, the LLM is guided to select the most appropriate dialogue act a ∈A, which then generates the response r.

3.3 LLM-ARK

Knowledge Graphs are structured knowledge networks composed of vertices, interpreted as entities, associated via edges or relationships. Let $\mathcal{E}$ stand for a collection of entities and $\mathcal{R}$ for a collection of relations. We represent the external KG as $\mathcal{G}$ = {V,E,$\mathcal{R}$}, where V and E denote the vertices and edges of the graph, respectively. Note that V = $\mathcal{E}$ and E $\subseteq V \times \mathcal{R} \times$ V. Let v denote a node and e denote an edge in $\mathcal{G}$. Given dialog context $X =$ and $\mathcal{G}$, we can identify an entity in the KG (e.g., an entity name The Wondering Earth) and and represent it as $v_s, v_s \in V$. The goal is to select a proper edge e_t at the t-th timestamp for one-hop reasoning.

Graph attention-based models require significant annotation effort since all potential paths must be evaluated, which can be computationally expensive for large Knowledge Bases (KBs) with millions of entities. To overcome this challenge, our study employs a policy gradient model that efficiently traverses the KG to select relationships and ultimately achieves the target, demonstrating proficiency in multi-hop reasoning.

KG reasoning naturally reduces to a finite horizon, deterministic partially observed Markov decision process that lie on a KG $\mathcal{G}$. We formulate KG reasoning as a Markov Decision Process (MDP) described by a five-tuple (S, O, A, T, R, γ):

- State. S is an infinite set of environment states, which encode information stored in Working Memory, including task background tb, user query q, dialog history h, current entity v_c, path history ph, current step t, The normal state is represented using a six-tuple: $S = (tb, q, h, v_c, ph, t)$.

- Observation. The complete state of the environment can be observed. Formally, the observation function $O = S$.

- Action. The set of possible actions $\mathcal{A}$ from a state S consists of all the environment information. Formally $\mathcal{A}_s = \{e \in E \colon S\} \cup \{(s, \varnothing, s)\}$. This means that each state's agent can choose one of all output edges of the current entity.

- Transition. Depending on the edge selected by the agent at time step t, the environment is changed deterministically by updating the state to the new environment.

For single turn dialogue, we update current entity, path history and step. Formally, the transition function : $\delta \colon \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is defined by $\delta(S, A) = (tb, q, h, v_c^{'}, ph^{'}, t^{'})$, For multi-turn dialogue wo also need to update user query and dialogue history. $\delta(S, A) = (tb, q^{'}, h^{'}, v_c^{'}, ph^{'}, t^{'})$, where $S = (tb, q, h, v_c, ph, t)$.

- Reward. We have a final reward of +1 if the current entity is the target entity v_g and -1 otherwise. if $S_t = (tb, q^{'}, h^{'}, v_c^{'}, ph^{'}, k^{'})$ is the final state, then we have a final reward of +1 if $v_c^{'} = v_g$, else -1.

- γ denote reward discounts factor are used to compute the reward information of each intermediate process when agent reaches the goal, or the end of the maximum step t.

3.3.1 Full Textual Environment

This module tracks the agent's state that captures all essential information in the conversation so far. FTE is a text dictionary structure, the same as Prompt Engineering's normal prompt format.

3.3.2 Agent

Inference to previous work Carta et al. (2023), we use standard RL practices by adding action heads - a Multi-Layer Perceptron (MLP) on top of the LLM. Thus, we can use only pretrained operations from the LLM and leverage language modeling heads' prior, this method is robust to any action space and can thus be used on any textual environment with no change. Agent has two components: LLM and PA-MLP.

LLM We initially utilize a LLM to encode the state S into a continuous vector $s \in \mathbb{R}^{2d}$. As a rule of thumb, for BERT models the cls token is used to represent the semantics of the whole sequence, while standard transformers and GPT-like LLMs use the embedding of the last token. We used the model on huggingface hub as well as the code to get the sequence vector representation[1]. It is defined by:

$$s = llm(fte) \tag{3}$$

PA-MLP Instead of just adding an MLP with a single output for the value on top of the last layer of the first decoder block as in the conventional multicategorization task, to enhance the ability of the large model to perceive the environment, we further fused the hidden state after the MLP with the Knowledge Graph exit path information, called PA-MLP (**P**ath **A**ware MLP). Recall that each possible action represents an outgoing edge e with information about the edge relation label r_l and the target vertex/entity v_d. So the embedding for each A_t is $[r_l; v_d]$, and stacking the embeddings for all outgoing edges we get the matrix A_t. The network taking this as input is parameterized as a three-layer feed-forward network (MLP) with $\tanh$ nonlinearity, which takes the FTE representation s and the embedding for the outgoing paths embedding and outputs a probability distribution over the possible actions from which a discrete action is sampled. The dimension of the MLP output hidden state is equal to the dimension of the path embedding. Finally, formulated as:

$$\begin{aligned} \mathbf{h_t} &= \mathbf{A_t}\left(\mathbf{W_3}\left(\tanh\left(\mathbf{W_2}\left(\tanh\left(\mathbf{W_1}\left(s_t\right)\right)\right)\right)\right)\right) \\ a_t &\sim \text{Categorical}\ \left(\text{softmax}\left(\mathbf{h_t}\right)\right) \end{aligned} \tag{4}$$

3.3.3 Training

Optimizer Our model is optimized by utilizing the experience accumulated by agent during KG reasoning. More formally, for the above policy network (π_θ), we want to find the parameter θ that maximizes the reward.

$$\begin{aligned} J(\theta) = \mathbb{E}_{(e_s, \mathcal{P}, e_g) \sim D} \mathbb{E}_{A_1, \ldots, A_{T-1} \sim \pi_\theta} \\ \left[R\left(S_t\right) \mid S_1 = \left(s_1\right)\right], \end{aligned} \tag{5}$$

where we assume that there is a true underlying distribution $(e1, r, e2) \sim \mathcal{P}$. To address this optimization challenge, we adopt an online reinforcement learning policy gradient algorithm, Proximal Policy Optimization (PPO). PPO is a family of policy optimization methods that use multiple epochs of stochastic gradient ascent to perform each policy update. These methods have the stability and reliability of trust-region methods(Schulman et al., 2017). For value approximation, we include a three-layer feed-forward network with a single output for the value, given by:

$$\mathbf{V} = \mathbf{W_3}\left(\tanh\left(\mathbf{W_2}\left(\tanh\left(\mathbf{W_1}\left(s_t\right)\right)\right)\right)\right) \tag{6}$$

Significantly, the LLM remains frozen for both the actor and critic modules, with only the linear forward layer being trained.

Replay Buffer The replay buffer stores the triplets $rb = (v_c, s, logit, a, s^{'}, done)$ of the reflection prompt, indicating the current entity, the

[1]https://huggingface.co

current state, logits, the selected action, the next state, and whether the episode has ended. The reason for recording the current entity is that we need to get all exit paths of the current entity for further fusion 4.

4 Experiments and Results

4.1 Datasets

OpenDialKG is a publicly available parallel corpus of conversations and Knowledge Graphs consisting of 91,000 conversations, each supplemented by paths connecting Knowledge Graph entities and their relationships. The purpose of the corpus is to present the implicit reasoning processes of human dialog as explicit computer operations on the Knowledge Graph. Following previous work described in Moon et al. (2019), we split this dataset into a 70% training set, a 15% validation set, and a 15% test set.

4.2 Baselines

We compared our results with these baseline models: Tri-LSTM, Seq2Seq, Seq2Path, DialKG Walker(Young et al., 2018; Moon et al., 2019), DiffKG, AttnFlow, AttnIO(Jung et al., 2020) and HiTKG(Ni et al., 2022). HiTKG is a hierarchical transformer-based tool that uses diverse inputs to predict KG paths. Our team chose HiTKG as a strong baseline. To evaluate the performance of the state-of-the-art LLMs on the KG inference task, we designed three prompt methods: GPT4-Standard, GPT4-Normal and GPT4-OPA. The difference between GPT4-Standard, GPT4-Normal is that GPT4-Normal has more awareness of dialog context and path history, while GPT4-OPA has more awareness of 2-hop exit path subgraphs compared to the former two. See appendix A.4 for full prompt.

4.3 Implement Details

The training was conducted on A40. Informed by prior research from Jung et al. (2020); Ni et al. (2022), we pre-trained the knowledge graph using TransE (Fan et al., 2014) based on this GitHub repository[2]. The objective was to unearth and explore entity relationships, expand the knowledge graph for connection prediction, and enable diverse reward function design. To facilitate reproducibility, we adopt an open-source LLM, i.e., LLaMA-2-7B(Touvron et al., 2023b). To reduce GPU memory usage and increase the pace of training, all experiments - excluding LLaMA-2-7B-ARK-FP32 were carried out with BFLOAT16(Kalamkar et al., 2019) half-precision format. Since all true paths in OpenDialKG are at most 2 hops, we set the maximum path length to $t = 2$. We included "Equal" to ensure that the model stops automatically after the second hop. To ensure fairness, we randomly shuffled the exit paths of the knowledge graph. We set max patience to 5, meaning that training is terminated if there is no boost for 5 rewards on the validation set. Further information on the hyperparameters is available in the Appendix 8.

[2] https://github.com/thunlp/OpenKE

4.4 Evaluation Metrics

In line with the baselines, we utilize recall@k as the evaluation metric for both path-level (path@k) and target entity-level (target@k) correctness.

4.5 Comparative Experiments

For the KG reasoning task, we assessed path recall at different K values (1, 3, 5, 10, 25) and target entity recall at position K (1, 3, 5, 10, 25). As presented in Table 1, the result demonstrate that our proposed model LLaMA-2-7b-ARK performs better than all benchmarked baselines in target@1, 5, 10, 25 metrics. The performance gain is signifcant, especially in recalls with taget@1, 10: there is a 5.28% relative improvement in target@1 and 9.59% in target@10. Unfortunately, our model's path@k evaluation matrix socores do not outperform the current state-of-the-art (SOTA) model HiTKG because we trained using only the target arrival reward function, but we are very extensible and there is potential for improvement. As described in this paper, we evaluate the performance of GPT-4 in performing dialog inference using manual prompt constructed with different environmental information. Therefore, we also report the performance of GPT-4 with different prompts on the same dataset.

At the decoding stage of AttnIO, AttnFlow and DiffKG, KG paths are predicted by scoring entity paths and relation paths respectively, and then rerank which makes it harder to achieve optimum. While a KG triple is composed of both, our model uses PA-MLP to aggregate all the exit path information to improve the perception of the agent, which is a more reasonable modeling approach.

HiTKG is state-of-the-art KG walker, which build a multi-hierarchy attention block to aggregate the multiscale information. However, different types of input data sources are difficult to aggregate,

Model	path@k					target@k				
	path@1	path@3	path@5	path@10	path@25	target@1	target@3	target@5	target@10	target@25
Tri-LSTM	3.2	14.2	22.6	36.3	56.2	-	-	-	-	-
Seq2Seq	3.1	18.3	29.7	44.1	60.2	-	-	-	-	-
DialKG Walker	13.2	26.1	35.3	47.9	62.2	-	-	-	-	-
Seq2Path	14.92	24.95	31.1	38.68	48.15	15.65	27.04	33.86	42.52	53.28
AttnFlow	17.37	24.84	30.68	39.48	51.4	18.97	36.23	45.48	58.84	71.35
AttnIO	23.72	37.53	43.57	52.17	62.86	24.98	43.78	53.49	65.48	78.79
HiTKG	**25.99**	**38.67**	**49.18**	**59.32**	**71.27**	31.11	46.29	55.59	71.61	86.09
T5-DiffKG	-	-	-	-	-	26.80	**54.33**	61.75	-	-
GPT-4-Standard	0.007	-	-	-	-	14.91	-	-	-	-
GPT-4-Normal	0.02	-	-	-	-	13.30	-	-	-	-
GPT-4-OPA	0.09	-	-	-	-	12.19	-	-	-	-
LLaMA-2-7B-ARK	16.59	27.17	34.85	47.88	62.32	**36.39**	53.63	**65.68**	**80.20**	**89.68**

Table 1: Path-level (path@k) and target-level (target@k) performance of KG path reasoning. LLM-ARK is benchmarked against several state-of-the-art baselines models on the OpenDialKG dataset.

and how well they are aggregated directly affects the performance of the model. We unify all the multi-scale input sources into the prompt, and due to the large model has a large number of instruction comprehension ability to get a rich information encoding representation.

The GPT-4-Standard and GPT-4-Normal methods are deficient in path awareness. GPT-4-OPA exhibit improved outcomes with the addition of path awareness. The generation of GPT-4 paths is entirely dependent on the background knowledge in the dataset during the training phase, and the GPT-4 generative model itself is not designed for sequential decision-making tasks, and achieving such a score has impressed us. Although we added states to GPT-4 through prompt, this is limited by the length of the prompt, which is fundamentally due to the fact that GPT-4 is inherently memoryless. Based on these factors, optimizing GPT-4 on multi-hop inference datasets of the Knowledge Graph to further improve its performance, generating human-preferred inference paths on large-scale Knowledge Graph datasets is still a challenge.

Considering LLMs as agents that explore a knowledge graph to acquire experience can benefit benefits from the positive-negative feedback optimization mechanism of the Reinforcement Learning Policy Supervisor Algorithm. This method enhances the training of our model to perform flexible reasoning on KGs in multi-step scenarios, outperforming not only GPT-4 but also smaller models. The model's superior performance corroborates the effectiveness of our approach.

4.6 Analysis Experiment

As shown in Table 2, LLM-ARK was benchmarked against multiple ablation models on the OpenDialKG dataset. **(1)** First, to evaluate the impact of instructions on model performance, we trained the LLaMA-2-7B-ARK-UI model without instruction. The results of this model are the closest to those of LLaMA-2-7B-ARK, indicating that the presence or absence of commands has an effect on the model's results, but not a serious one. **(2)** Then, we have implemented IEEE 754 floating-point format (FP32) operations for our experiments. The results show that using the BFLOAT16 tensor for training, recall@K gives better results than FP32 without changing the hyperparameters. **(3)** Next, LLaMA-2-7B-ARK-WT that was trained by randomly initializing relation and entity embedding. The decrease in performance indicates that the absence of knowledge graph representation learning has negatively impacted the training process. **(4)** We conducted the fourth ablation experiment LLaMA-2-7B-ARK-WP and found that the performance of our export environment-aware sub-module PA-MLP decreases substantially if we do not consider the exit paths. The score for path@1 is only 0.98%, which is 15.61% lower than LLaMA-2-7B-ARK. These ablation experiments and results demonstrate the contribution and necessity of our subcomponents to the model.

4.7 Case Study

We resort to a case study, for a clear presentation of LLM-ARK's path reasoning process as shown in Table 3. Note that there are hundreds of neighbor nodes connected to each entity in the external KG. Intuitively, there could be diverse knowledge

Model	path@k					target@k				
	path@1	path@3	path@5	path@10	path@25	target@1	target@3	target@5	target@10	target@25
LLaMA-2-7B-ARK	16.59	**27.17**	**34.85**	**47.88**	62.32	**36.39**	**53.63**	**65.68**	**80.20**	89.68
LLaMA-2-7B-ARK-UI	**16.87**	27.01	34.35	47.67	**63.03**	34.70	52.02	62.66	78.09	88.65
LLaMA-2-7B-ARK-FP32	14.59	24.64	32.24	45.80	61.75	34.35	52.57	62.51	79.18	88.51
LLaMA-2-7B-ARK-WT	1.10	3.44	5.71	10.47	15.42	9.45	19.46	50.94	71.43	**94.03**
LLaMA-2-7B-ARK-WP	0.98	2.53	3.28	4.88	5.52	18.90	40.67	54.76	77.25	93.08

Table 2: Path-level (path@k) and target-level (target@k) performance of supervised KG path reasoning (metric: recall@k). LLM-ARK is benchmarked against several ablation models on the OpenDialKG dataset.

<table>
<tr><td colspan="3">### Task Background:
Performing 2-hop reasoning on the knowledge graph.
Instruction
If you don’t think it’s necessary to perform the second hop in reasoning, stop the reasoning with the ’Equal’ relation.
Given the Task Background and the Environment, directly output this path in triplet format without any other content.</td></tr>
<tr><td rowspan="3">Success</td><td>FTE</td><td>### Environment:
Dialog History: []
Utterance: Could you recommend popular books by Gail Carson Levine?
Path History: []
Current Entity: Gail Carson Levine</td></tr>
<tr><td>Ground Truth Path</td><td>["Gail Carson Levine","∼written_by","The Two Princesses of Bamarre"]</td></tr>
<tr><td>LLM-ARK Reasoning Path</td><td>[["Gail Carson Levine","∼written_by","The Two Princesses of Bamarre"], ["The Two Princesses of Bamarre","Equal","The Two Princesses of Bamarre"]]</td></tr>
<tr><td rowspan="3">Failed</td><td>FTE</td><td>### Environment:
Dialog History: ["user: Can you recommend a movie like the Shooter?",
"assistant: A movie similar to Shooter is Nothing to Lose."]
Utterance: "Ok who is in that one?"
Path History: [["Shooter","has_genre","Thriller"],["Thriller","∼has_genre","Nothing to Lose"],["Nothing to Lose","starred_actors","Michael McKean"]]
Current Entity: "Michael McKean"</td></tr>
<tr><td>Ground Truth Path</td><td>["Michael McKean","∼starred_actors","Nothing to Lose"]</td></tr>
<tr><td>LLM-ARK Reasoning Path</td><td>[["Michael McKean","∼starred_actors","Used Cars"],["Used Cars", "Equal", "Used Cars"]]</td></tr>
</table>

Table 3: Successes and failures of our model when performing inference tasks on the OpenDialKG dataset.

paths as response to the user’s question. As the success story shows, our model makes good use of FTE information and exit path information to make decisions, rather than making decisions based on relationships alone, because Gail Carson Levine’s work is not limited to The Two Princesses of Bamarre. As shown in the error case, our model still reasons about wrong paths, partly due to the dataset itself, because OpenDialKG is an open-domain conversational knowledge graph inference dataset, and similar contexts and the same starting entities in the training set choose different Groud Truth exit paths, and so it can interfere with the training of our model. It is worth mentioning that OpenDialKG is not a unique path inference; there are many potential paths to reach the target entity. To summarize, our model would have the potential for better performance on non-open-domain conversational knowledge graph inference datasets.

5 Conclusion

This paper evaluates the ability of current state-of-the-art LLM-based dialog systems in handling KG conversational reasoning tasks. To enhance LLM’s performance on this task, we introduce LLM-ARK, a full-text environment-aware Knowledge Graph inference agent optimized using online reinforcement learning. Empirical analysis demonstrates that our model outperforms GPT-4 and smaller models. The experiments also sheds light on the model’s performance can be seriously affected by the mismatch between the LLMs and the environment information. Our method can inspire subsequent researchers to pay attention to the critical role of considering various factors during model optimization in the field of LLM-based conversational KG reasoning tasks.

References

Thomas Carta, Clément Romac, Thomas Wolf, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer. 2023. Grounding large language models in interactive environments with online reinforcement learning.

Yue Dong, John Wieting, and Pat Verga. 2022. Faithful to the document or to the world? mitigating hallucinations via entity-linked knowledge in abstractive summarization.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. 2023. Guiding pretraining in reinforcement learning with large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 8657–8677. PMLR.

Miao Fan, Qiang Zhou, Emily Chang, and Thomas Fang Zheng. 2014. Transition-based knowledge graph embedding with relational mapping properties. In *Pro-*

ceedings of the 28th Pacific Asia Conference on Language, Information and Computing, pages 328–337, Phuket,Thailand. Department of Linguistics, Chulalongkorn University.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. 2022. Inner monologue: Embodied reasoning through planning with language models.

Jaehun Jung, Bokyung Son, and Sungwon Lyu. 2020. AttnIO: Knowledge Graph Exploration with In-and-Out Attention Flow for Knowledge-Grounded Dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3484–3497, Online. Association for Computational Linguistics.

Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, Jiyan Yang, Jongsoo Park, Alexander Heinecke, Evangelos Georganas, Sudarshan Srinivasan, Abhisek Kundu, Misha Smelyanskiy, Bharat Kaul, and Pradeep Dubey. 2019. A study of bfloat16 for deep learning training.

Mehran Kazemi, Najoung Kim, Deepti Bhatia, Xin Xu, and Deepak Ramachandran. 2023. Lambada: Backward chaining for automated reasoning in natural language.

Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023. Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning.

Aiwei Liu, Xuming Hu, Lijie Wen, and Philip S. Yu. 2023. A comprehensive evaluation of chatgpt's zero-shot text-to-sql capability.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2023. Reasoning on graphs: Faithful and interpretable large language model reasoning.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 845–854, Florence, Italy. Association for Computational Linguistics.

Jinjie Ni, Vlad Pandelea, Tom Young, Haicang Zhou, and Erik Cambria. 2022. Hitkg: Towards goal-oriented conversations via multi-hierarchy learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11112–11120.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback.

Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S.M Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. 2023. The troubling emergence of hallucination in large language models - an extensive definition, quantification, and prescriptive remediations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2541–2573, Singapore. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Paulo Shakarian, Abhinav Koyyalamudi, Noel Ngu, and Lakshmivihari Mareedu. 2023. An independent evaluation of chatgpt on mathematical word problems (mwp).

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. 2022. Progprompt: Generating situated robot task plans using large language models.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2023. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Yi-Lin Tuan, Sajjad Beygi, Maryam Fazel-Zarandi, Qiaozi Gao, Alessandra Cervone, and William Yang Wang. 2022. Towards large-scale interpretable knowledge graph reasoning for dialogue systems. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 383–395, Dublin, Ireland. Association for Computational Linguistics.

George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard E. Turner, Zoubin Ghahramani, and Sergey Levine. 2018. The mirage of action-dependent baselines in reinforcement learning.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh Murthy, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, Ran Xu, Phil Mui, Huan Wang, Caiming Xiong, and Silvio Savarese. 2023. Retroformer: Retrospective large language agents with policy gradient optimization.

Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting end-to-end dialogue systems with commonsense knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The surprising effectiveness of PPO in cooperative multi-agent games. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Houyu Zhang, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2020. Grounded conversation generation as guided traverses in commonsense knowledge graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2031–2043, Online. Association for Computational Linguistics.

Knowledge Graph			Dataset	
Entity	Relation	Triplets	Train Data	Test Data
100,927	1,383	1,189,192	12,345	2,646

Table 4: Detailed information about the number of knowledge graph entity-relationship triples and the number of dataset segmentation samples after processing the OpenDialKG dataset.

A Appendix

A.1 Data Format

We preprocessed the OpenDialKG raw data to fit our KG inference task. There are individual errors in the raw data, and the information of the dataset after our screening is shown in Table 4.

A.2 Tricks

In the original paper of PPO, no implementation details and techniques are mentioned other than the use of GAE to compute the dominance function. Referring to this repository[3], we employ several optimization tricks. In the actual code implementation, to encourage the diversity of paths sampled by the strategy during training, we added an entropy regularization term to our loss function. We used the operation of normalization of advantage proposed in the paper (Tucker et al., 2018). Learning rate decay can enhance the smoothness in the late training stage to some extent and improve the training effect. Here we use the linear decay of learning rate, with the number of training steps learning rate from the initial value of a linear decline to 0. Gradient clipping is a trick introduced to prevent the gradient from exploding during the training process, which also serves to stabilize the training process. Orthogonal Initialization is a neural network initialization method proposed to prevent problems such as gradient vanishing and gradient explosion at the beginning of training. Referring to the MAPPO (Yu et al., 2022), the Adam optimizer individually sets eps=1e-5, and this particular setting can improve the training performance of the algorithm to some extent.

A.3 Limitation

A.3.1 Limitations of Inference Efficiency

Efficiency is always a significant issue when building deep learning models based on LLMs. Although our research freezes the parameters of the

[3]https://github.com/Lizhi-sjtu/DRL-code-pytorch

LLM in the back-propagation stage and uniformly uses the bfloat16 computational type, the huge number of parameters of the model leads to inefficient forward propagation and large GPU memory usage when collecting experience and inference. As stated in the LLaMA paper(Touvron et al., 2023a), the efficiency of the model's inference is more crucial than its training efficiency. It is acceptable for the training process to be slower, but the inference must be faster. Improving the inference speed of the model while ensuring its effectiveness is a challenge. In addition to constructing the research model, online applications based on LLMs must also address the efficiency issue. Conversational reasoning models based on LLMs must be efficient for real-time applications. The inference efficiency is crucial for building online applications based on LLMs.

A.3.2 Limitations of Entity Embedding

Our research work has identified limitations in the semantic representation of knowledge graph entities. The attributes of knowledge graph entities should be considered during the reasoning process. However, these attributes may be lengthy descriptions that are not easily processed by our TransE knowledge graph semantic embedding model. Furthermore, while most knowledge graphs are currently represented in text form, it is equally important to consider multimodal knowledge graph reasoning in research. By constructing a reasoning model based on multimodal inputs, machines can better describe and understand the real world. For example, the soon-to-be two-dimensionalized Law in Three Body Death Forever says "Oh, it's time to go into the picture, kids, go ahead," and the user asks a question about this scenario, "can you help me find some pictures related to this galaxy?". The model may need to deduce that the two-dimensional representation portrayed in this book is the Milky Way galaxy, and then locate relevant images of the galaxy. Our model is currently unable to incorporate the combination of multi-modal, multi-attribute entities, which is a limitation of our work in this endeavor, as well as an area for future research efforts.

A.4 GPT4 Prompts

For GPT4-OPA prompt, since max length is 2, we need to recursively get all exit paths at the next level of all exit paths of the current entity, most of which are omitted due to the large number of KG subgraph triples of exit paths.

```
Standard Prompt
### Task Background
Performing 2-hop reasoning on the knowledge graph.

### Instruction
If you don't think it's necessary to perform the second hop in reasoning, stop the reasoning with the 'Equal' relation.
Given the Task Background and the Environment, directly output this path in triplet format without any other content.

### Environment
Utterance: What do you think about the Washinton Redskins? Are you a fan?
Current Entity: Washington Redskins

### Examples
.
.
.

### Response
```

Table 5: GPT4-Standard prompt only perceived user's query and Current Entity.

```
Normal Prompt
### Task Background
Performing 2-hop reasoning on the knowledge graph.

### Instruction
If you don't think it's necessary to perform the second hop in reasoning, stop the reasoning with the 'Equal' relation.
Given the Task Background and the Environment, directly output this path in triplet format without any other content.

### Environment
Dialog History: []
Utterance: What do you think about the Washinton Redskins? Are you a fan?
Path History: []
Current Entity: Washington Redskins

### Examples
.
.
.

### Response
```

Table 6: GPT4-Normal prompt has more awareness of dialog context and path history.

```
OPA(Out Paths Aware) Prompt
### Task Background
Performing 2-hop reasoning on the knowledge graph.

### Instruction
Given the Task Background and the Environment, please choose select two consecutive paths KG path from a set of Out Paths.
If you don't think it's necessary to perform the second hop in reasoning, just select the 'Equal' relation at the second hop.
Directly output these path in triplet format without any other content.

### Environment
Dialog History: []
Utterance: What do you think about the Washinton Redskins? Are you a fan?
Path History: []
Current Entity: Washington Redskins
Out Path: ['Washington Redskins,Equal,
Washington Redskins',
'Washington Redskins,∼Game,Mike Sellers',
'Washington Redskins,∼Runner-up,Super Bowl VII',
.
.
.
'Ladell Betts,Ethnicity,African American']

### Examples
.
.
.

### Response
```

Table 7: GPT4-OPA prompt has more awareness of 2-hop exit KG path subgraphs.

A.5 HyperParameters

Computing Infrastructure	Tesla A40 GPU
Search Strategy	Beam Search
Training Efficiency	6 seconds per step

Hyperparameter	Best Setting
use transe	True
out path aware	True
bf16	True
relation embedding size	200
entity embedding size	200
max out	50
number of explorations	8
replay buffer size	4096
mini batch size	1024
positive reward	1
negative reward	-1
actor learning rate	5e-5
critic learning rate	5e-5
gamma	0.95
lamda	0.95
epsilon	0.2
K epochs	10
use advantage normalization	True
use entropy coef	0.01
use learning rate decay	True
use gradient clip	True
use orthogonal init	True
set adam eps 1e-5	True
use tanh	True

Table 8: Additional implementation detail of LLM-ARK.