Advancing Multi-Agent RAG Systems with Minimalist Reinforcement Learning

Yihong Wu†
Université de Montréal
Montréal, QC, Canada
yihong.wu@umontreal.ca

Liheng Ma†
McGill University & Mila
Montréal, QC, Canada
liheng.ma@mail.mcgill.ca

Muzhi Li†
The Chinese University of Hong Kong
Hong Kong, China
mzli@cse.cuhk.edu.hk

Jiaming Zhou
Huawei Noah's Ark Lab
Montréal, QC, Canada

Lei Ding
University of Manitoba
Winnipeg, MB, Canada

Jianye Hao
Tianjin University
Tianjin, China

Ho-fung Leung
Independent Researcher
Hong Kong, China

Irwin King
The Chinese University of Hong Kong
Hong Kong, China

Yingxue Zhang
Huawei Noah's Ark Lab
Montréal, QC, Canada

Jian-Yun Nie
Université de Montréal
Montréal, QC, Canada

ABSTRACT

Large Language Models (LLMs) equipped with modern Retrieval-Augmented Generation (RAG) systems often employ multi-turn interaction pipelines to interface with search engines for complex reasoning tasks. However, such multi-turn interactions inevitably produce long intermediate contexts, as context length grows exponentially with exploration depth. This leads to a well-known limitation of LLMs: their difficulty in effectively leveraging information from long contexts. This problem is further amplified in RAG systems that depend on in-context learning, where few-shot demonstrations must also be included in the prompt, compounding the context-length bottleneck. To address these challenges, we propose **Mujica-MyGo**, a unified framework for efficient multi-turn reasoning in RAG. Inspired by the divide-and-conquer principle, we introduce **Mujica** (Multi-hop Joint Intelligence for Complex Question Answering), a multi-agent RAG workflow that decomposes multi-turn interactions into cooperative sub-interactions, thereby mitigating long-context issues. To eliminate the dependency on in-context learning, we further develop **MyGO** (Minimalist Policy Gradient Optimization), a lightweight and efficient reinforcement learning algorithm that enables effective post-training of LLMs within complex RAG pipelines. We provide theoretical guarantees for MyGO's convergence to the optimal policy. Empirical evaluations across diverse question-answering benchmarks—covering both text corpora and knowledge graphs—show that **Mujica-MyGO** achieves superior performance. Proofs, implementation details, and prompt templates are available in the extended version [1].

KEYWORDS

Reinforcement Learning, Multi-Agent, RAG, QA, LLM

ACM Reference Format:
Yihong Wu†, Liheng Ma†, Muzhi Li†, Jiaming Zhou, Lei Ding, Jianye Hao, Ho-fung Leung, Irwin King, Yingxue Zhang, and Jian-Yun Nie. 2026. Advancing Multi-Agent RAG Systems with Minimalist Reinforcement Learning. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 10 pages. https://doi.org/10.65109/QCQC1144

1 INTRODUCTION

The advent of Large Language Models (LLMs) [1, 41] has redefined expectations for Artificial Intelligence, offering new opportunities for many domains. Despite their versatility, LLMs suffer from hallucination [16] and knowledge cutoffs [9]. While injecting knowledge through continuous learning is a possible solution, it is computationally expensive and risks catastrophic forgetting [21]. To address this issue, Retrieval-Augmented Generation (RAG) [14, 24], a technique enabling LLMs to incorporate external knowledge via prompts, has drawn significant attention among researchers.

Most advanced RAG systems [7, 20, 29, 35, 49] employ a multi-turn pipeline to effectively address complex questions. This process typically involves decomposing the initial question into sub-questions, retrieving relevant information for each, and synthesizing a final answer. Such an approach allows for the generation of more fine-grained queries, enabling search engines to return more relevant passages and ultimately enhancing overall performance.

Despite its effectiveness, this multi-turn interaction introduces a significant challenge: the long-context problem. For instance, to answer the query, "*Which U.S. presidents previously served as*

†qually contributed.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/QCQC1144

[1]Available at https://arxiv.org/abs/2505.17086

governors?", a RAG system must first identify all U.S. presidents and then retrieve biographical information to determine which individuals held gubernatorial positions before their presidency. Each president has an extensive historical record, and aggregating such information results in a large volume of retrieved text within a single LLM context window. Given that LLMs struggle to capture information accurately in long contexts [33], this naive aggregation can lead to performance degradation.

Moreover, this issue is compounded by the reliance of In-Context Learning (ICL) [11] to guide the LLM's behavior in most RAG workflows. This technique requires providing few-shot demonstrations within the prompt to instruct the model on the desired task. The performance of ICL is highly dependent on the precision and comprehensiveness of these examples, especially their ability to cover corner cases. Consequently, these detailed demonstrations further inflate the context length, exacerbating the long-context problem.

To address the aggregation problem, we propose **Mujica** (Multi-hop Joint Intelligence for Complex Question Answering), a multi-agent RAG system designed for complex queries. Mujica consists of two core modules: a **Planner** and a **Worker**. The Planner initiates the process by analyzing the original query, decomposing it into sub-questions, and delegating them to the Worker. Based on the responses from the Worker, the Planner iteratively formulates new queries until it has sufficient information to finalize an answer or reaches the iteration limit. The Worker, in turn, receives sub-questions from the Planner, retrieves relevant passages from search engines, and returns concise summaries. This planner-worker architecture enables a clear separation between high-level reasoning and low-level retrieval. By maintaining a clean, summary-based history in the Planner's context, Mujica effectively mitigates the long-context problem.

To eliminate the need for cumbersome few-shot demonstrations in in-context learning (ICL), post-training is a crucial step for a complex RAG system like Mujica. However, the highly customized nature of such systems means that unified datasets for Supervised Fine-Tuning (SFT) are generally unavailable. This scarcity of trajectory data naturally frames the post-training process as a Reinforcement Learning (RL) problem, where the RAG system constitutes the policy and its performance on a task determines the reward. Nevertheless, optimizing this policy via RL is non-trivial due to the intricate interactions between the agents and the search engine. To tackle this challenge, we introduce Minimalist policy Gradient Optimization (**MyGO**), a simple and efficient RL algorithm specifically designed for such complex RAG systems.

We designed MyGO to be minimalist, deliberately avoiding components like reward weighting, importance sampling, or token clipping that are common in other algorithms. This approach stands in contrast to methods like Proximal Policy Optimization (PPO) [48], which requires an auxiliary value network to stabilize training, and Group Relative Policy Optimization (GRPO) [51], which must generate multiple trajectories per query for reward normalization, exacerbating latency. The key innovation in MyGO lies in its sampling strategy. By sampling trajectories from an asymptotically approximate optimal policy, we can directly use Maximum Likelihood Estimation (MLE) for policy updates. This insight allows MyGO to be implemented easily within standard SFT frameworks, lending it high stability during training and simplifying hyperparameter tuning. We provide a detailed theoretical justification for the validity of this approach and show through empirical experiments on various datasets that Mujica-MyGO effectively improves RAG performance across different LLMs.

2 RAG AGENT

2.1 Problem Definition

This work addresses Open-Domain Multi-Hop Question Answering (MHQA), a key challenge and benchmark for RAG. QA systems aim to provide correct answers to given questions. Compared with single-hop QA, MHQA necessitates reasoning across multiple sources (hops) [39], which renders MHQA a more challenging task. We focus on the open-domain setting, where relevant information must be retrieved from large-scale corpora (knowledge graphs or text collections).

The inherent complexity of MHQA, characterized by multi-stage planning requirements and iterative evidence synthesis, necessitates framing the solution mechanism as an autonomous agent. We define an agent as an entity capable of planning, taking actions based on the environment, and engaging in sequential reasoning [32]. We observe that LLMs equipped with RAG perform functions analogous to such an agent. Specifically, LLMs determine information requirements and search strategies (planning), formulate search queries (acting), and synthesize final answers from gathered evidence (reasoning). Recognizing this functional correspondence, we position the LLM itself as the core agent within our system.

2.2 Mujica

As previously noted, repeated retrieval from RAG systems generates extended contexts that are challenging for LLMs to effectively process. To mitigate this, a straightforward approach is to adopt a divide-and-conquer approach by partitioning the context into smaller segments. Accordingly, we introduce **M**ulti-hop **J**oint **I**ntelligence for **C**omplex Question **A**nswering (**Mujica**) – a multi-agent RAG framework that decomposes multi-turn interactions into coordinated sub-interactions.

Figure 1 shows the overall pipeline of the proposed agentic RAG framework. Given a complex question q, our RAG agent aims to decompose it into a series of simple subquestions $\{q_1, q_2, \cdots, q_n\}$, and progressively answers each subquestion based on their internal dependencies until obtaining the final answer a. Let $\mathcal{I}$ denote the instruction prompts, $C = \{C_1, C_2, \cdots, C_n\}$ denote the supporting contexts of each subquestion q_i, and a_i an answer to it. Inspired by [29], we can formulate the answer generation process as the following objective function:

$$\begin{aligned} f_{\mathrm{QA}}(q, I, C) &= P(a|q, I, C) \\ &= \prod_{i=1}^{n} P(q_i|a_{<i}, q_{<i}, q, \mathcal{I}) \cdot \prod_{i=1}^{n} P(a_i|q_i, C_i, \mathcal{I}), \end{aligned} \tag{1}$$

where $q_{<i}$ and $a_{<i}$ denote all preceding subquestions of subquestion q_i and their corresponding answers. Based on the equation above, the QA framework can naturally be decoupled into two specialized roles:

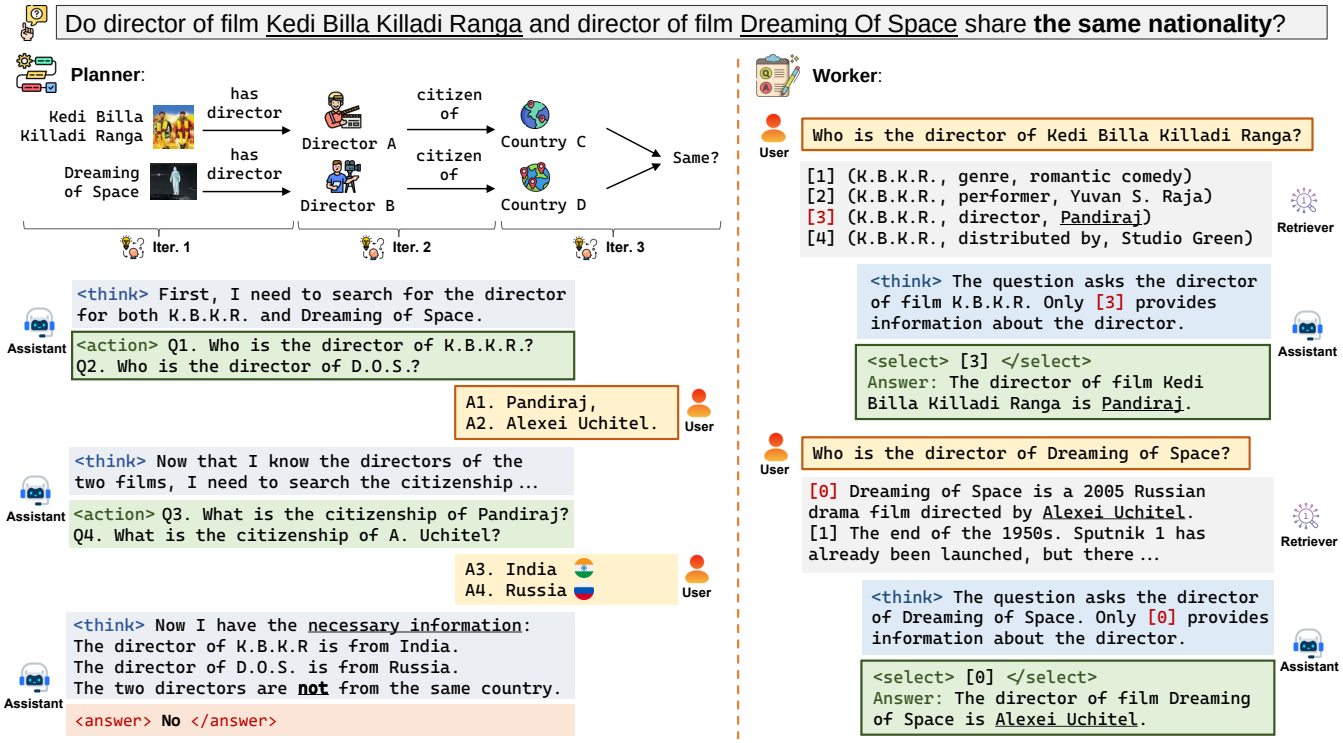

Figure 1: The end-to-end architecture of the proposed Mujica framework.

1) *Planner*: Being responsible for planning how to answer a complex question, the planner determines the series of subquestions to be addressed, identifies what specific information needs to be retrieved, and adjusts the plan based on the answer of each subquestion.
2) *Worker*: Acting as a mini-RAG system, the worker directly interacts with the retriever and tackles specific sub-questions identified by the planner.

Planner and *worker* interact with each other through a conversational process, each serving as both user and assistant for one another. The planner and worker may operate as independent agents (distinct LLMs) or as virtual agents backed by a shared LLM but conditioned on distinct instruction prompts. Considering the substantial size of LLMs, we opt for the latter design for better computational efficiency.

Planner: Planning Subquestions as a Directed Acyclic Graph. Recent work (e.g., Search-o1 [29], PoG [7]) has demonstrated that performing chain-of-thought reasoning through Eq. 1 is capable of deriving correct answers. However, in real-world scenarios, the subquestions posed are not naturally shaped as a sequence, since not all subquestions depend on answers to preceding ones. As shown in Fig. 2, there might be two conditionally independent subquestions, $S_{2,1}$ and $S_{2,2}$, which are dependent in the later subquestion $S_{4,1}$. The dependency relations form a directed acyclic graph (DAG). Directly applying Eq. 1 to such a reasoning process can be both inefficient and suboptimal. Therefore, to effectively handle DAG dependency graphs, we allow our Mujica planner to ask subquestions in multiple iterations sequentially, where the answers of subquestions at an iteration can become dependent in later iterations. In each iteration, Mujica will simultaneously ask multiple conditionally independent subquestions.

More specifically, following the previous work [67], in each iteration, the planner will execute two steps: ***think*** and ***action***. In the ***think*** step, at the i-th iteration, the planner summarizes the information obtained in the previous iterations and judges whether the supporting information it has gathered is sufficient to answer the complex question q. If all necessary information is available, the planner generates its conclusion for the entire complex question. Otherwise, the planner will excute the ***action*** step - it formulates the subquestions and outsources them to a ***Worker*** to retrieve the corresponding information from the environment. After gathering the information from the ***Worker***, the ***Planner*** will move to the next iteration, and this process continues until obtaining the final answer.

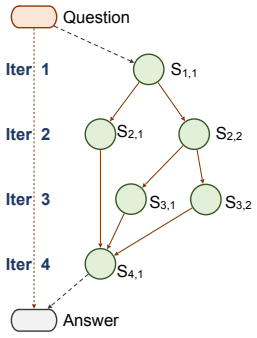

Figure 2: Modeling Complex Question Answering Process as a Directed Acyclic Graph.

Worker: Interacting with the External Environments. As aforementioned, the ***Worker*** is responsible for handling interactions with the external environments, and answers questions assigned by the planner. Specifically, we utilize the same LLM-agent with a mini-RAG system. For each simple question q_i, the worker invokes an external retriever to fetch the top k most relevant supporting contexts from a KG or a document corpus. [2] Then, the ***Worker*** employs the LLM to review the target of the question, examine each retrieved context and accordingly anwser the subquestions outsourced by the ***Planner***. Specifically, the LLM is instructed to explicitly illustrate its thoughts, and select relevant contexts by providing their indexes.

3 RL FOR RAG AGENT

3.1 Reinforcement Learning Setting

Our RAG agent is implemented within a conversational framework. A conversation is modeled as a finite sequence of interactions between the environment and the agent, denoted as $(e_0, a_0, e_1, a_1, \ldots, e_T, a_T)$. The initial environment message e_0 comprises the user's question and any necessary system prompts. For each turn $t \in [0, T]$, a_t is the agent's action, and e_{t+1} is the subsequent environment message. We define the state s_t at turn t as the history of all preceding interactions, $s_t = (e_0, a_0, e_1, a_1, \ldots, a_{t-1}, e_t)$. Then, a trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$ to the final answer of the question can be denoted as the sequence of states and actions in each step. This formulation adheres to a Markov Decision Process (MDP). Since the golden reasoning path to the ground truth answer is not available in real-world scenarios and most datasets, we do not have supervision signals for each action a_t to conduct supervised finetuning. The only supervision signal available to estimate the reward $r(\tau) \in \mathbb{R}$ for the entire trajectory τ is the final answer, which naturally corresponds to the terminal reward in an RL setting.

We consider a policy-based method - the RAG agent adopts a policy π_θ to generate reasoning trajectories, where the parameters θ are optimized using RL techniques. The objective of the RL is to find the best policy that maximizes the expected cumulative reward:

$$\mathcal{J}(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[r(\tau)] \,, \tag{2}$$

If trajectories sampled from π_θ are differentiable w.r.t. θ, the gradient of the optimization objective can be estimated using the Vanilla Policy Gradient (VPO) [60]:

$$\nabla_\theta \mathcal{J}(\theta) = \mathop{\mathbb{E}}_{\tau\sim\pi_\theta}\left[r(\tau) \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \right] . \tag{3}$$

3.2 Reinforcement Learning for LLMs

Even though the VPO (Eq. 3) estimator is unbiased, it often exhibits high gradient variance, which potentially renders the training process unstable. To address this issue, many previous approaches [46, 48] propose to replace the raw trajectory reward $r(\tau)$ with the Generalized Advantage Estimation (GAE) [47]. These RL techniques, particularly Proximal Policy Optimization (PPO) [48] and its variants, have been successfully applied in the post-training process for LLM [41, 44]. Recently, Group Relative Policy Optimization (GRPO) [51] has been introduced as a simplified alternative to PPO, reducing the training costs while maintaining comparable performance. The objective function of PPO and GRPO, as adapted for our QA context, can be written as follows:

$$\begin{aligned}\mathcal{J}(\theta) = \mathop{\mathbb{E}}_{\tau\sim\pi_{\theta_k}} \left[A(\tau) \sum_{t=0}^{T} \min\left(\mathbf{r}_t,\ \text{clip}\left(\mathbf{r}_t, 1-\epsilon, 1+\epsilon\right)\right) \right] \\ - \beta \mathbb{D}_{\text{KL}}(\pi_\theta || \pi_{\text{ref}}) \,,\end{aligned} \tag{4}$$

where ϵ, β are hyperparameters; $\pi_{\text{ref}}, \pi_{\theta_k}$ denote the reference policy and the behavior policy, respectively; $\mathbf{r}_t = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}$ is the importance sampling rescaling factor; and $A(\tau)$ is the advantage of applying trajectory τ. Specifically, in PPO, the advantage function $A(\tau)$ is estimated by $r(\tau) - V(\tau)$, relying on a learnable value function $V(\tau)$. In GRPO, $A(\tau)$ is instead estimated by $\frac{r(\tau)-\bar{r}}{\text{std}(r)}$, where $\bar{r}$ and $\text{std}(r)$ are the mean and standard deviation of $r(\tau)$ for each τ obtained from the same sampling group, respectively.

3.3 MyGO

Despite its wide adoption, a PPO-style objective has two shortcomings: hyperparameter sensitivity and training inefficiency. The ϵ hyperparameter is highly sensitive - a small ϵ may ignore most available trajectories during training, leading to training inefficiency; while a large ϵ might lose control over the importance sampling scale, resulting in a less stable training process. The advantage estimation renormalizes the original rewards relative to either the value functions or the group average, which weakens the supervision signals from trajectories with large rewards that are helpful for training [34]. It is worth mentioning that *training inefficiency and instability are significant challenges when training LLMs, which is both computationally expensive and time-consuming*. On the other hand, compared with previous RL scenarios in robot control and video games [4], the LLM setting poses a new characteristic: fast simulation, deterministic environment, and trajectory-level reward [30]. In this scenario, *sampling desired trajectories from the environment is relatively more efficient.*

To this end, we propose our new RL algorithm, **MyGO**: **M**inimalist Polic**y** **G**radient **O**ptimization, which shifts the burden from model training to trajectory sampling, in which we can fully leverage the fast and relatively inexpensive simulation of LLMs [22, 74]. Unlike previous methods, MyGO disentangles the RL training process into two phases: sampling and training. During sampling, MyGO directly samples trajectories from the asymptotically optimal distribution and consequently allows the use of Maximum Likelihood Estimation (MLE) to optimize the policy function. Notably, optimizing a model via MLE is simple, stable, and well established as a regular training technique widely used in unsupervised pre-training [43] and supervised fine-tuning [41].

Sampling. We first characterize the target optimal distribution π^*. Alternative to the original objective $\mathcal{J}$, we consider the expected reward with entropy regularization $\mathcal{J}' = \mathbb{E}[r(\tau)] + \alpha\mathbf{H}(\pi)$ with $\alpha \in \mathbb{R}^+$, which, as a Boltzmann distribution [75], has a closed-form solution for the corresponding optimal policy, denoted as π'^*:

$$\begin{aligned}\pi'^*(\tau) &= \frac{\exp(r(\tau)/\alpha)}{Z(\alpha)} \,, \\ \text{where } Z(\alpha) &= \int_{\mathcal{T}} \exp(r(\tau')/\alpha) d\tau' \,,\end{aligned} \tag{5}$$

[2] We employ bge-large-en-v1.5 sentence transformer as the external retriever.

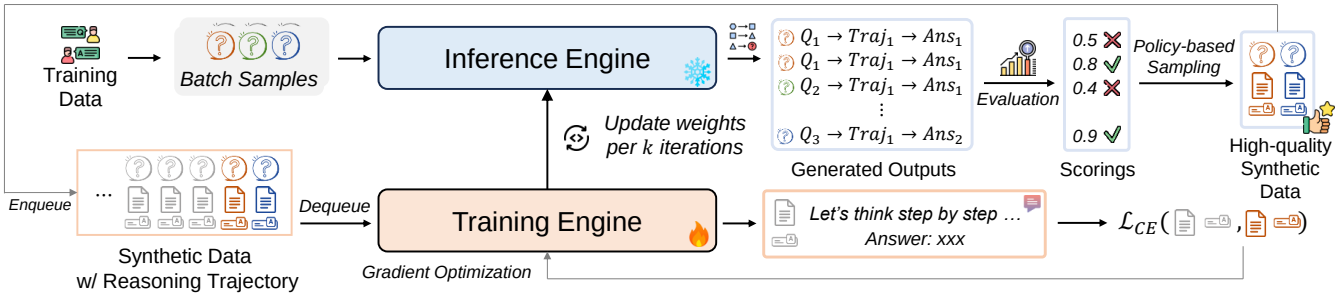

Figure 3: The Proposed Minimalist Policy Gradient Optimization Framework.

where α is the temperature and $Z(\alpha)$ is the partition function that normalizes the distribution over the space of all possible trajectories $\mathcal{T}$. With a sufficiently small $\alpha \to 0$, we have $\mathcal{J}' \to \mathcal{J}$, thus $\pi'^* \to \pi^*$. As $\alpha \to 0$, π'^* becomes sharp and highly skewed, concentrating its probability mass on trajectories yielding high rewards. Therefore, given a sufficiently large threshold K close to $\sup_{\tau\in\mathcal{T}} r(\tau)$, sampling trajectories τ with $r(\tau) > K$ is asymptotically equivalent to sampling from the π'^* and thus effectively approximating to sampling from the optimal policy π^* of $\mathcal{J}$.

In practice, we relax the choice of K by setting an update role of $K := \max(K', (\bar{r}_n/(r_{\text{sup}}+1)) \cdot r_{\text{sup}})$, where r_{sup} denotes $\sup_{\tau\in\mathcal{T}} r(\tau)$, $\bar{r}_n$ is the empirical average reward of a batch of n sampled trajectories, and K' is the previous threshold value which is initialized as a moderate value depending on the choice of dataset. As the policy π_θ improved, K will be asymptotically close to $\sup_{\tau\in\mathcal{T}} r(\tau)$, thus making the sampling distribution close to π'^*.

We highlight that this approach is practically feasible for the reward landscapes for tasks of the LLM agent, which differ significantly from those in complex robotic control problems. For LLM agents, trajectories frequently culminate in outcomes that are clearly classifiable as successful or unsuccessful. Empirically, we observe that well-prompted LLMs can readily generate a significant proportion of such successful trajectories, which makes directly sampling from π^* feasible.

Let $\pi^{>K}$ denote the distribution that consists of trajectories with $r > K$. To clarify, we provide two propositions as follows, where the proofs can be found in the extended version:

Proposition 3.1. *Given $\delta \in \mathbb{R}$, $\alpha \in \mathbb{R}$, there exists $K \in \mathbb{R}^+$ such that the following inequality holds:*

$$\mathbb{D}_{KL}(\pi^{>K} || \pi^*) < \delta$$

Proposition 3.2. *The following property holds when α approaches 0:*

$$\underset{\pi^{>K}}{\text{Var}}[r] \propto 1/Z^{>K}(\alpha)^2\ ,\ Z^{>K}(\alpha) = \int_K^{\sup(r)} \exp(r/\alpha)dr$$

Proposition 3.1 states that if K is sufficiently large, then $\pi^{>K}$ can always approximate π^*. Proposition 3.2 says the variance of $\pi^{>K}$ is small, considering that Z is usually a large value. These two propositions guarantee the proximity and stability of our proposed sampling strategy.

Learning. Once a sufficiently large dataset of optimal trajectories $\mathcal{D}$ has been collected, the optimization of policy π_θ can be simplified to maximizing the log-likelihood $\mathcal{L}$ of these trajectories. The learning process can be conducted in an online or an offline manner. This yields the following MLE gradient:

$$\begin{aligned}\nabla_\theta \mathcal{L} &= \frac{1}{|\mathcal{D}|} \sum_{\tau_i \in \mathcal{D}} \nabla_\theta \log \pi_\theta(\tau_i) \\ &= \frac{1}{|\mathcal{D}|} \sum_{\tau_i \in \mathcal{D}} \sum_{j=0}^{|\tau_i|} \nabla_\theta \mathbb{1}(o_j^i) \log \pi_\theta(o_j^i | o_{<j}^i)\ , \end{aligned} \tag{6}$$

where o_j^i denotes the j-th token from τ_i and $o_{<j}^i$ denotes the sequence of tokens preceding o_j, $\mathbb{1}(o_j)$ is a characteristic function indicating whether o_j is a token from the agent. The last equality holds because the action likelihood in the language setting could be further elaborated with token likelihood, i.e, $\log \pi_\theta(a_t|s_t) = \sum_j \log \pi_\theta(o_j|o_{<j})$.

This MLE objective, Eq. 6, is equivalent to a standard cross-entropy loss function, widely used in both unsupervised pre-training [43] and supervised fine-tuning [41] of language models. Compared to PPO-style algorithms, Eq. 4, MLE does not incorporate an advantage function, policy ratio clipping, and importance sampling. Similarly, unlike methods such as VPO, Eq. 3, MLE directly uses the curated data $\mathcal{D}$ for training and does not explicitly use reward values in the loss computation. We empirically observed that MLE leads to more stable training. We explain this increased stability by several factors: (i) the dataset $\mathcal{D}$ consists of trajectories sampled from a more stationary (approximated optimal) distribution, reducing variance compared to on-policy RL where the data distribution shifts rapidly; (ii) the direct use of a log-likelihood objective (cross-entropy) is often well-behaved and less sensitive to reward scaling issues that can affect RL algorithms; and (iii) the avoidance of importance sampling ratios, which can introduce instability if they become too large or small.

Compared with previous works, like RAFT [10] and STaR [69], which try to simplify RL in LLM post-training, we use a progressive threshold to select data. More importantly, we provide a theoretical justification that elucidates why this approach, yet relatively simple, yields strong performance, offering novel insights into its efficacy.

4 EXPERIMENTS

4.1 Environment Design

Datasets and Evaluation Metrics. To evaluate the effectiveness of our proposed agent workflow (Mujica) and training method (MyGO) on complex question answering tasks, we utilize four representative benchmark datasets: 2Wiki-MultihopQA (2Wiki) [15], QALD-10 [57], HotpotQA [66], and MuSiQue [55]. 2Wiki and QALD-10 are KBQA benchmarks; HotpotQA and MuSiQue are text-based QA dataset. We use 2Wiki and HotpotQA for training, QALD-10 and MuSiQue for zero-shot evaluation. We assess QA performance using Exact Match (EM) and F1 score. For each dataset, these metrics are computed using the official evaluation scripts provided by the respective dataset creators.

Environment Setting. To evaluate our agent's capabilities across different data modalities, we derive two experimental environments from the 2Wik dataset: 2Wiki-KG and 2Wiki-Text. This setup allows us to evaluate the agent's performance on both structured (KGs) and unstructured (text) data. We use the F1 score as the reward function for all environments.

The **2Wiki-KG** environment is centered around the Wikidata KG [58]. A knowledge graph (KG) is composed of factual statements represented as triples, such as (*Donald Trump*, *position held*, *President of the United States*). When the agent visits an entity, it can access all triples for which that entity is the head. The agent's task involves traversing the KG by interpreting the semantic relationships encoded in these triples. For example, to answer the question "*Who is the mother of the director of film Polish-Russian War (Film)?*", the agent must identify and follow the correct reasoning path, such as: *Polish-Russian War, director, Xawery Żuławski, mother, Małgorzata Braunek*. To emphasize the reasoning aspect, we follow prior work [37, 53] by providing the agent with the gold topic entities extracted from the dataset as the entry.

We establish the **2Wiki-Text** environment to further evaluate our agent's ability to interact with text-based search engines. The original 2Wiki dataset provides ten supporting passages for each question, which contain the necessary information to derive the answer and the "distractor setting" is designed to test multi-hop reasoning abilities without retrieval. In our 2Wiki-Text environment, we adapt this by aggregating all passages to form a large corpus. This approach, following prior work [2, 56], simulates an open-domain retrieval task. The agent's objective is to answer questions by issuing search queries and then synthesizing information from the retrieved passages.

Beyond the 2Wiki environments, we extend our text-based evaluation using the HotpotQA dataset, from which we construct two additional settings: **Hotpot** and **Hotpot-Kimi**. In **Hotpot**, we follow the vanilla setting that uses the Wikipedia dump provided by the dataset [66], which only includes the title and the introductory paragraph. However, we found that the retriever [19] is the bottleneck for QA performance in the HotpotQA dataset. To isolate and more directly assess the agent's reasoning capabilities independent of retrieval imperfections, we introduce the **Hotpot-Kimi** environment. Similar to but different from the distractor settings [15, 66], in the Hotpot-Kimi environment,the agent is directly provided with the ten gold supporting passages associated with each question, thereby guaranteeing access to all necessary evidence. However, its primary purpose here is to establish an idealized retrieval condition. Therefore, the gold passages are not directly fed into the QA-agent like distractor setting. Detailed examples and prompts for each environment are provided in the extended version.

4.2 Main Results

4.2.1 ***Effectieness of MyGO.*** We evaluate our RAG agent under three distinct settings: Few-Shot, WarmUp, and MyGO. In the **Few-Shot** setting, the agent is provided with a small number of illustrative examples via in-context learning within the prompt. Following common practice, we then perform a **WarmUp** phase. This phase is to adapt the base model beyond reliance on few-shot exemplars. For this purpose, we fine-tune the model on 1k samples with EM of 1. The **MyGO** training then commences, using this warmed-up model as its initialization. We conduct our main experiments with the `Qwen2.5-7B-Instruct` (Qwen) LLM. The feasibility of using `Llama-3.1-8B-Instruct` (Llama) as the backbone LLM will also be discussed.

Table 1: Comparative Performance of Models on 2Wiki-KG and 2Wiki-Text Environments.

Type	Model	2Wiki-KG		2Wiki-Text	
		EM	F1	EM	F1
Few-Shot	GPT-4.1	78.50	84.24	30.70	52.14
Few-Shot		57.60	62.24	23.35	33.52
WarmUp	Qwen2.5-7B	74.93	80.74	50.18	56.25
MyGO		77.63	84.15	53.17	59.62
Few-Shot		77.70	82.09	37.83	46.53
WarmUp	Llama3.1-8B	82.00	85.90	58.03	64.81
MyGO		85.93	91.61	58.88	65.88

Table 1 presents the performance of these models on the 2Wiki dataset. Agents trained with MyGO consistently demonstrate strong performance. In the 2Wiki-KG environment, MyGO-Qwen achieves performance on par with a few-shot prompted GPT-4.1, while the Llama model significantly surpasses this baseline. In the 2Wiki-Text environment, both MyGO-trained Qwen and Llama models substantially outperform the GPT-4.1 few-shot baseline. Despite the same underlying questions, overall performance in the 2Wiki-Text environment is notably lower than in 2Wiki-KG. This disparity can be attributed to the differing data modalities and interaction paradigms. Interacting with KG may more closely resemble navigating discrete choices (e.g., selecting entities or relations), potentially reducing the burden on complex retrieval or synthesis from noisy search results. Conversely, text-based interaction requires robust open-retrieval over the corpus where explicit relationships between retrieved passages are often absent, demanding more sophisticated synthesis capabilities. Nevertheless, the consistent improvements observed affirm MyGO's efficacy across these varied settings.

Table 2 details the results on the HotpotQA dataset. Notably, even the WarmUp models exhibit strong performance, outperforming

Table 2: Comparative Performance of Qwen on Hotpot and Hotpot-Kimi Environment.

Type	Model	Hotpot		Hotpot-Kimi	
		EM	F1	EM	F1
Few-Shot	GPT-4.1	29.80	48.09	33.50	53.35
Few-Shot		15.77	27.96	15.60	32.70
WarmUp	Qwen2.5-7B	40.55	52.35	52.51	66.04
MyGO		41.54	53.79	54.07	68.48

the few-shot GPT-4.1 baseline. However, the performance gain from MyGO over the WarmUp models is less pronounced in the Hotpot environment. We attribute this observation to two limitations inherent in this setup. Firstly, the efficacy of the dense retriever can be a bottleneck; essential evidence may not be surfaced for some queries. Secondly, the HotpotQA evaluation protocol has no answer aliases. This is problematic because LLMs frequently generate answers that are semantically equivalent to the ground truth but lexically different, which are penalized by strict string-matching metrics. To mitigate the impact of imperfect retrieval, we further evaluate using the Hotpot-Kimi environment, which is designed to simulate near-perfect retrieval by providing gold supporting passages. In this idealized Hotpot-Kimi setting, MyGO demonstrates substantial and consistent performance improvements over the WarmUp models, underscoring its effectiveness when reasoning capabilities can be more directly assessed.

4.2.2 ***Effectiveness of Mujica.*** In order to evaluate the contribution of our two critical design choices in the proposed Mujica framework, we replace the planner–worker split with vanilla iterative RAG, and substitute DAG-like reasoning with chain-like reasoning. For the sake of fair comparison, all subsequent experiments are conducted with a pre-trained Qwen 2.5-7B-Instruct model, which eliminates confounding effects introduced by model training.

From Table 4 we conclude that our proposed DAG-like reasoning paradigm not only enables independent sub-questions to be processed in parallel, but also outperforms the chain-style baseline. We attribute the performance gain to the paradigm's ability to effectively decompose complex questions and resolve sub-problems through optimal reasoning paths. The structured decomposition allows the model to better capture the underlying logic of multi-hop questions, leading to more accurate and robust reasoning outcomes. In addition, our planner–worker architecture surpasses the vanilla iterative RAG setup. We attribute the better performance to Mujica's ability to prevent irrelevant information from interfering with the model's reasoning and planning process.

4.2.3 ***Mujica-MyGO v.s. Baseline Methods.*** In order to comprehensively evaluate the performance of our proposed solution, we compare Mujica-MyGO with a series of RAG systems. Apart from direct inference with pretrained LLM [64] and vanilla RAG [25], we consider 7 iterative RAG solutions: RAFT [72], RAFe [38], HippoRAG [13], Iter-RetGen [50], IRCoT [56], IterDRAG [68], RAG-star [17] and 4 RAG agents: Search-o1 [29], RAG-Gym [25] and ReSearch [8]. Considering the differences in backbone LLMs and retrieval methods employed by each RAG system, it is inappropriate to compare the performance of different RAG systems solely based on their evaluation metrics.

Table 3 presents the experimental results on three representative multi-hop question answering datasets, which shows that *Mujica-MyGO is an effective and efficient multi-hop QA solution.* Despite leveraging a 7B Qwen2.5 LLM as its backbone, Mujica-MyGO achieves state-of-the-art performance on theMuSiQue and 2Wiki datasets and delivers competitive results on HotpotQA. One may notice that some baseline methods outperform Mujica-MyGO on the HotpotQA dataset. However, these approaches typically rely on larger-scale backbone LLMs and/or more sophisticated retrieval methods. In general, larger backbone models (e.g., GPT and QwQ) tend to yield better results. In addition, incorporating advanced retrievers (e.g., HopRetriever [28], Dragon-plus [31]) or expanding the knowledge corpus via search engines (e.g., Google, Bing) can further enhance performance. Nevertheless, it is important to emphasize that improving retrieval is not the focus of this paper. In order to evaluate the effectiveness of the proposed method, we opt to adopt the official corpus released with the HotpotQA dataset [66]. Across model variants, we find that even without costly reinforcement learning, a small amount of SFT data is sufficient for Mujica to surpass baseline methods based on question decomposition and iterative retrieval, the method to consistently surpass, demonstrating its robustness. However, every approach has its trade-offs. As a complex question-answering pipeline, Mujica imposes substantial demands on the LLM's instruction-following and reasoning capabilities, making it difficult for small-scale pre-trained LLMs to achieve desirable results in few-shot settings.

4.3 Additional Studies - Out-of-Domain Eval.

To assess generalization capabilities, we conduct out-of-distribution (OOD) evaluations where models are RL post-trained on one dataset and evaluated on another one. Specifically, we examine cross-dataset transfer from HotpotQA to Musique (text-based), and from 2WikiKG to Qald-10 (KG-based). As shown in Table 5, MyGO demonstrates strong OOD performance. As presented in Table 5, models trained with MyGO demonstrate robust OOD generalization. Specifically, applying MyGO on the source datasets leads to significant performance improvements on the respective target OOD datasets when compared to the few-shot baseline. These findings indicate that MyGO does not overfit to the training data; rather, it appears to foster the acquisition of more generalizable reasoning and decision-making strategies. This capacity for effective generalization to unseen datasets aligns with a primary motivation for employing RL in the development of intelligent agents.

5 RELATED WORK

MHQA with textual corpus. Retrieval-based methods dominate the landscape of textual MHQA since supporting contexts from external corpora can effectively compensate for deficiencies in the inherent knowledge of language models and mitigate hallucinations. Traditional approaches primarily utilized sentence transformers [6] for dense retrieval or BM25 [45] for keyword matching. Recently,

Table 3: End-to-end textual multi-hop question answering results.

Method	Backbone	Retriever	HotpotQA		MuSiQue		2Wiki-Text	
			EM	F1	EM	F1	EM	F1
Direct Inference	Qwen 2.5-7B	-	18.3	-	3.1	-	25.0	-
CoT	Qwen 2.5-7B	-	9.2	-	2.2	-	11.1	-
RAG	Qwen 2.5-7B	E5	29.9	-	5.8	-	23.5	-
RAFT	Llama 3.1-8B	Dragon-Plus	41.0	51.6	13.8	24.0	39.4	45.8
RaFe	GPT 4o-mini	Google Search	40.6	55.4	11.8	23.8	36.2	39.3
HippoRAG	GPT 3.5-turbo	ColBERTv2	45.7	**59.2**	21.9	33.3	47.7	62.7
Iter-RetGen	Qwen 2.5-7B	FlashRAG	34.4	-	8.7	-	27.9	-
IRCoT	Qwen 2.5-7B	BM25	30.3	-	7.0	-	21.6	-
IterDRAG	Gemini 1.5	Gecko 1B	38.4	49.8	22.6	35.0	44.3	54.6
RAG-Star	Llama 3.1-8B	bge-large-en-v1.5	42.0	54.4	13.0	22.2	34.0	42.0
RAG Agents								
Search-o1	Qwen 2.5-7B	E5	18.7	-	5.8	-	17.6	-
Search-o1	QwQ 32B	Bing Web Search	45.2	57.3	16.6	28.2	58.0	**71.4**
Search-R1	Qwen 2.5-7B	E5	37.0	-	14.6	-	41.4	-
RAG-Gym	Llama 3.1-8B	bge-base + BM25	44.1	56.8	-	-	50.2	57.9
ReSearch	Qwen 2.5-7B	FlashRAG	43.5	-	22.3	-	47.6	-
Settings			*In-domain*		*Out-of-domain*		*In-domain*	
Mujica (few-shot)	Qwen 2.5-7B	bge-large-en-v1.5	15.77	27.96	5.34	17.81	23.35	33.52
Mujica w/ warm up	Qwen 2.5-7B	bge-large-en-v1.5	40.55	52.35	N/A	N/A	50.18	56.25
Mujica-MyGO	Qwen 2.5-7B	bge-large-en-v1.5	41.54	53.79	**26.11**	**35.92**	**53.17**	59.62

Table 4: Experimental results for design choices of the proposed Mujica framework.

Settings	2Wiki-KG		2Wiki-Text	
	EM	F1	EM	F1
Mujica (Qwen2.5-7B)	57.60	62.24	23.35	33.52
- w/o. planner-worker split	54.85	59.63	23.22	33.40
- w/o. DAG-like reasoning	53.75	59.08	21.73	32.36

Table 5: Out-of-distribution Evaluation on Qwen 2.5-7B: HotpotQA→Musique and 2Wiki-KG→QALD-10.

Type	Model	Hotpot → MuSiQue		2Wiki → QALD	
		EM	F1	EM	F1
Few-Shot	Qwen	5.34	17.81	27.92	37.11
MyGO	2.5-7B	26.11	35.92	39.85	49.73

extensive research has focused on optimizing the QA pipeline, particularly through the decomposition of complex questions [65, 67], query rewriting and refinement [5, 38], and the iterative retrieval of relevant evidences in multiple steps [3, 18, 50, 56].

Knowledge-based question answering. Early approaches treat KBQA as a semantic parsing task, which generates and executes SPARQL queries [23, 42, 59]. These methods usually suffer from syntax errors and inaccurate entity labels, resulting in limited accuracy. Recently, researchers have shifted towards leveraging RAG and the advanced semantic understanding capabilities of LLMs to generate answers [61]. LLM-based KBQA approaches can be generally categorized into two types: iterative exploration and subgraph reasoning. Iterative exploration methods [7, 12, 26, 37, 52–54, 63, 70, 71, 73] progressively retrieve and verify KG elements through multi-step reasoning, balancing interpretability with computational overhead. These approaches rely heavily on LLMs' inherent reasoning capabilities due to the lack of supervised intermediate steps in most datasets. In contrast, Subgraph reasoning methods [27, 35, 36, 40, 62] extract relevant KG substructures in a single retrieval step for LLM-based answering, enabling efficient fine-tuning but suffering from the incompleteness of retrieved subgraphs and underlying knowledge bases.

6 CONCLUSION

In this work, we introduce Mujica, an interactive multi-agent framework for RAG systems, alongside MyGO, a novel reinforcement learning method designed to simplify its optimization. Mujica employs a planner-worker architecture to systematically decompose and resolve complex questions, while MyGO enhances training efficiency through an intelligent sampling strategy. Our experiments demonstrate that the proposed framework not only achieves significant performance improvement across multiple QA datasets but also exhibits strong inductive reasoning. These results underscore its scalability and potential to generalize effectively in real-world scenarios.

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Cecilia Aguerrebere, Ishwar Bhati, Mark Hildebrand, Mariano Tepper, and Ted Willke. 2023. Similarity search in the blink of an eye with compressed indices. *Proceedings of the VLDB Endowment* 16, 11 (2023), 3433–3446.

[3] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=hSyW5go0v8

[4] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. arXiv:arXiv:1606.01540

[5] Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. RQ-RAG: Learning to Refine Queries for Retrieval Augmented Generation. In *First Conference on Language Modeling*. https://openreview.net/forum?id=tzE7VqsaJ4

[6] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. arXiv:2402.03216 [cs.CL] https://arxiv.org/abs/2402.03216

[7] Liyi Chen, Panrong Tong, Zhongming Jin, Ying Sun, Jieping Ye, and Hui Xiong. 2024. Plan-on-Graph: Self-Correcting Adaptive Planning of Large Language Model on Knowledge Graphs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=CwCUEr6wO5

[8] Mingyang Chen, Tianpeng Li, Haoze Sun, Yijie Zhou, Chenzheng Zhu, Haofen Wang, Jeff Z. Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and Weipeng Chen. 2025. ReSearch: Learning to Reason with Search for LLMs via Reinforcement Learning. arXiv:2503.19470 [cs.AI] https://arxiv.org/abs/2503.19470

[9] Jeffrey Cheng, Marc Marone, Orion Weller, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2024. Dated data: Tracing knowledge cutoffs in large language models. *arXiv preprint arXiv:2403.12958* (2024).

[10] Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, SHUM KaShun, and Tong Zhang. 2023. RAFT: Reward rAnked FineTuning for Generative Foundation Model Alignment. *Transactions on Machine Learning Research* (2023).

[11] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A Survey on In-context Learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 1107–1128. https://doi.org/10.18653/v1/2024.emnlp-main.64

[12] Siyuan Fang, Kaijing Ma, Tianyu Zheng, Xinrun Du, Ningxuan Lu, Ge Zhang, and Qingkun Tang. 2024. KARPA: A Training-free Method of Adapting Knowledge Graph as References for Large Language Model's Reasoning Path Aggregation. arXiv:2412.20995 [cs.CL] https://arxiv.org/abs/2412.20995

[13] Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[14] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*. PMLR, 3929–3938.

[15] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, Donia Scott, Nuria Bel, and Chengqing Zong (Eds.). International Committee on Computational Linguistics, Barcelona, Spain (Online), 6609–6625. https://doi.org/10.18653/v1/2020.coling-main.580

[16] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* 43, 2 (2025), 1–55.

[17] Jinhao Jiang, Jiayi Chen, Junyi Li, Ruiyang Ren, Shijie Wang, Xin Zhao, Yang Song, and Tao Zhang. 2025. RAG-Star: Enhancing Deliberative Reasoning with Retrieval Augmented Verification and Refinement. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Luis Chiruzzo, Alan Ritter, and Lu Wang (Eds.). Association for Computational Linguistics, Albuquerque, New Mexico, 7064–7074. https://aclanthology.org/2025.naacl-long.361/

[18] Zhouyu Jiang, Mengshu Sun, Lei Liang, and Zhiqiang Zhang. 2025. Retrieve, Summarize, Plan: Advancing Multi-hop Question Answering with an Iterative Approach. arXiv:2407.13101 [cs.CL] https://arxiv.org/abs/2407.13101

[19] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering.. In *EMNLP (1)*. 6769–6781.

[20] Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-Search-Predict: Composing Retrieval and Language Models for Knowledge-Intensive NLP. *arXiv preprint arXiv:2212.14024* (2022).

[21] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.

[22] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*. 611–626.

[23] Yunshi Lan and Jing Jiang. 2020. Query Graph Generation for Answering Multi-hop Complex Questions from Knowledge Bases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 969–974. https://doi.org/10.18653/v1/2020.acl-main.91

[24] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* 33 (2020), 9459–9474.

[25] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 793, 16 pages.

[26] Kun Li, Tianhua Zhang, Xixin Wu, Hongyin Luo, James Glass, and Helen Meng. 2024. Decoding on Graphs: Faithful and Sound Reasoning on Knowledge Graphs through Generation of Well-Formed Chains. arXiv:2410.18415 [cs.CL] https://arxiv.org/abs/2410.18415

[27] Mufei Li, Siqi Miao, and Pan Li. 2025. Simple is Effective: The Roles of Graphs and Large Language Models in Knowledge-Graph-Based Retrieval-Augmented Generation. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=JvkuZZ04O7

[28] Shaobo Li, Xiaoguang Li, Lifeng Shang, Xin Jiang, Qun Liu, Chengjie Sun, Zhenzhou Ji, and Bingquan Liu. 2021. HopRetriever: Retrieve Hops over Wikipedia to Answer Complex Questions. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 15 (May 2021), 13279–13287. https://doi.org/10.1609/aaai.v35i15.17568

[29] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic Search-Enhanced Large Reasoning Models. arXiv:2501.05366 [cs.AI] https://arxiv.org/abs/2501.05366

[30] Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. 2024. ReMax: a simple, effective, and efficient reinforcement learning method for aligning large language models. In *Proceedings of the 41st International Conference on Machine Learning*. 29128–29163.

[31] Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to Train Your Dragon: Diverse Augmentation Towards Generalizable Dense Retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 6385–6400. https://doi.org/10.18653/v1/2023.findings-emnlp.423

[32] Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, Sirui Hong, Hongzhang Liu, Shaokun Zhang, Kaitao Song, Kunlun Zhu, et al. 2025. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv preprint arXiv:2504.01990* (2025).

[33] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics* 12 (2024), 157–173.

[34] Zhuang Liu and Kaiming He. 2024. A Decade's Battle on Dataset Bias: Are We There Yet? *arXiv preprint arXiv:2403.08632* (2024).

[35] LINHAO LUO, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on Graphs: Faithful and Interpretable Large Language Model Reasoning. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=ZGNWW7xZ6Q

[36] LINHAO LUO, Zicheng Zhao, Gholamreza Haffari, Chen Gong, and Shirui Pan. 2025. Graph-constrained Reasoning: Faithful Reasoning on Knowledge Graphs with Large Language Models. https://openreview.net/forum?id=6embY8aclt

[37] Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. 2025. Think-on-Graph 2.0: Deep and Faithful Large Language Model Reasoning with Knowledge-guided Retrieval Augmented Generation. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=oFBu7qaZpS

[38] Shengyu Mao, Yong Jiang, Boli Chen, Xiao Li, Peng Wang, Xinyu Wang, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2024. RaFe: Ranking Feedback Improves Query Rewriting for RAG. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 884–901. https://doi.org/10.18653/v1/2024.findings-emnlp.49

[39] Vaibhav Mavi, Anubhav Jangra, Adam Jatowt, et al. 2024. Multi-hop question answering. *Foundations and Trends® in Information Retrieval* 17, 5 (2024), 457–586.

[40] Costas Mavromatis and George Karypis. 2024. GNN-RAG: Graph Neural Retrieval for Large Language Model Reasoning. arXiv:2405.20139 [cs.CL] https://arxiv.org/abs/2405.20139

[41] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[42] Sukannya Purkayastha, Saswati Dana, Dinesh Garg, Dinesh Khandelwal, and G.P Shrivatsa Bhargav. 2022. A Deep Neural Approach to KGQA via SPARQL Silhouette Generation. In *2022 International Joint Conference on Neural Networks (IJCNN)*. 1–8. https://doi.org/10.1109/IJCNN55064.2022.9892263

[43] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. [n.d.]. Improving language understanding by generative pre-training. ([n. d.]).

[44] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2023), 53728–53741.

[45] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (April 2009), 333–389. https://doi.org/10.1561/1500000019

[46] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*. PMLR, 1889–1897.

[47] John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. 2016. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. http://arxiv.org/abs/1506.02438

[48] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[49] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 9248–9274.

[50] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 9248–9274. https://doi.org/10.18653/v1/2023.findings-emnlp.620

[51] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300* (2024).

[52] Yuan Sui, Yufei He, Nian Liu, Xiaoxin He, Kun Wang, and Bryan Hooi. 2025. FiDeLiS: Faithful Reasoning in Large Language Model for Knowledge Graph Question Answering. arXiv:2405.13873 [cs.AI] https://arxiv.org/abs/2405.13873

[53] Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-Graph: Deep and Responsible Reasoning of Large Language Model on Knowledge Graph. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=nnVO1PvbTv

[54] Xingyu Tan, Xiaoyang Wang, Qing Liu, Xiwei Xu, Xin Yuan, and Wenjie Zhang. 2025. Paths-over-Graph: Knowledge Graph Empowered Large Language Model Reasoning. arXiv:2410.14211 [cs.CL] https://arxiv.org/abs/2410.14211

[55] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop Questions via Single-hop Question Composition. *Transactions of the Association for Computational Linguistics* (2022).

[56] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 10014–10037. https://doi.org/10.18653/v1/2023.acl-long.557

[57] Ricardo Usbeck, Xi Yan, Aleksandr Perevalov, Longquan Jiang, Julius Schulz, Angelie Kraft, Cedric Möller, Junbo Huang, Jan Reineke, Axel-Cyrille Ngonga Ngomo, Muhammad Saleem, and Andreas Both. 2024. QALD-10 – The 10th challenge on question answering over linked data: Shifting from DBpedia to Wikidata as a KG for KGQA. *Semantic Web* 15, 6 (2024), 2193–2207. https://doi.org/10.3233/SW-233471 arXiv:https://journals.sagepub.com/doi/pdf/10.3233/SW-233471

[58] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A Free Collaborative Knowledgebase. *Commun. ACM* 57, 10 (sep 2014), 78–85. https://doi.org/10.1145/2629489

[59] Keheng Wang, Feiyu Duan, Sirui Wang, Peiguang Li, Yunsen Xian, Chuantao Yin, Wenge Rong, and Zhang Xiong. 2023. Knowledge-Driven CoT: Exploring Faithful Reasoning in LLMs for Knowledge-intensive Question Answering. arXiv:2308.13259 [cs.CL] https://arxiv.org/abs/2308.13259

[60] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8 (1992), 229–256.

[61] Yike Wu, Nan Hu, Sheng Bi, Guilin Qi, Jie Ren, Anhuan Xie, and Wei Song. 2023. Retrieve-Rewrite-Answer: A KG-to-Text Enhanced LLMs Framework for Knowledge Graph Question Answering. arXiv:2309.11206 [cs.CL] https://arxiv.org/abs/2309.11206

[62] Mufan Xu, Kehai Chen, Xuefeng Bai, Muyun Yang, Tiejun Zhao, and Min Zhang. 2025. LLM-based Discriminative Reasoning for Knowledge Graph Question Answering. arXiv:2412.12643 [cs.CL] https://arxiv.org/abs/2412.12643

[63] Yao Xu, Shizhu He, Jiabei Chen, Zihao Wang, Yangqiu Song, Hanghang Tong, Guang Liu, Jun Zhao, and Kang Liu. 2024. Generate-on-Graph: Treat LLM as both Agent and KG for Incomplete Knowledge Graph Question Answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 18410–18430. https://doi.org/10.18653/v1/2024.emnlp-main.1023

[64] An Yang, Baosong Yang, Binyuan Hui, and Others. 2024. Qwen2 Technical Report. arXiv:2407.10671 [cs.CL] https://arxiv.org/abs/2407.10671

[65] Diji Yang, Jinmeng Rao, Kezhen Chen, Xiaoyuan Guo, Yawen Zhang, Jie Yang, and Yi Zhang. 2024. IM-RAG: Multi-Round Retrieval-Augmented Generation Through Learning Inner Monologues. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 730–740. https://doi.org/10.1145/3626772.3657760

[66] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600* (2018).

[67] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=WE_vluYUL-X

[68] Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. 2025. Inference Scaling for Long-Context Retrieval Augmented Generation. arXiv:2410.04343 [cs.CL] https://arxiv.org/abs/2410.04343

[69] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems* 35 (2022), 15476–15488.

[70] Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Liu Yong, and Shen Huang. 2024. End-to-End Beam Retrieval for Multi-Hop Question Answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 1718–1731. https://doi.org/10.18653/v1/2024.naacl-long.96

[71] Minghan Zhang, Zhen Yang, Hongsheng Wu, Yongxing Lin, Jie Chen, Zhen Duan, and Shu Zhao. 2025. ToG-I: Progressively Instructed Knowledge Graph-based Large Language Model Reasoning. https://openreview.net/forum?id=oW3XIIHaOn

[72] Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. RAFT: Adapting Language Model to Domain Specific RAG. arXiv:2403.10131 [cs.CL] https://arxiv.org/abs/2403.10131

[73] Ruilin Zhao, Feng Zhao, Long Wang, Xianzhi Wang, and Guandong Xu. 2024. KG-CoT: Chain-of-Thought Prompting of Large Language Models over Knowledge Graphs for Knowledge-Aware Question Answering. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Kate Larson (Ed.). International Joint Conferences on Artificial Intelligence Organization, 6642–6650. https://doi.org/10.24963/ijcai.2024/734 Main Track.

[74] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Livia Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. 2024. Sglang: Efficient execution of structured language model programs. *Advances in Neural Information Processing Systems* 37 (2024), 62557–62583.

[75] Brian D Ziebart. 2010. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University.