

Reason-Align-Respond: Aligning LLM Reasoning With Knowledge Graphs for KGQA

Xiangqing Shen¹, Fanfan Wang², Zinong Yang², Bing Wang³, Wenli Du³, Chengqing Zong⁴, Fellow, IEEE, and Rui Xia¹

Abstract—Large language models (LLMs) have demonstrated remarkable capabilities in complex reasoning tasks, yet they often suffer from hallucinations and lack reliable factual grounding. Meanwhile, knowledge graphs (KGs) provide structured factual knowledge, but lack the flexible reasoning abilities of LLMs. In this paper, we present Reason-Align-Respond (RAR), a novel framework that systematically integrates LLM reasoning with knowledge graphs for knowledge graph question answering (KGQA). Our approach consists of three key components: a Reasoner that generates human-like natural language reasoning chains, an Aligner that maps these chains to valid KG paths, and a Responder that synthesizes the final answer. We formulate this process as a latent variable mixture model and optimize it using the Expectation-Maximization algorithm, which iteratively refines the reasoning chains and knowledge paths. Extensive experiments on multiple benchmarks demonstrate the effectiveness of RAR, achieving state-of-the-art performance with Hit scores of 93.3% and 91.0% on WebQSP and CWQ respectively. Human evaluation confirms that RAR generates high-quality, interpretable reasoning chains well-aligned with KG paths while maintaining computational efficiency during inference.

Index Terms—Knowledge graph question answering, knowledge graph reasoning, large language model, latent variable model, expectation maximization.

I. INTRODUCTION

LARGE language models (LLMs) have demonstrated remarkable capabilities in handling various complex tasks [1]. However, their reasoning processes often lack reliable knowledge support, resulting in a lack of interpretability and the occurrence of hallucinations, which can easily lead to factual errors and logical inconsistencies. Knowledge graphs (KGs) [2], which represent and store factual knowledge in a structured form, consist of entities, relationships, and attributes that form

Received 17 August 2025; revised 24 January 2026; accepted 8 February 2026. Date of publication 17 February 2026; date of current version 5 June 2026. This work was supported by the Natural Science Foundation of China under Grant 62476134. Recommended for acceptance by G. Kim. (Corresponding author: Rui Xia.)

Xiangqing Shen and Rui Xia are with the School of Intelligence Science and Technology, Nanjing University, Suzhou 215163, China (e-mail: xqshen@nju.edu.cn; rxia@nju.edu.cn).

Fanfan Wang and Zinong Yang are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: ffwang@njut.edu.cn; znyang@njut.edu.cn).

Bing Wang and Wenli Du are with the School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China (e-mail: wangb07@ecust.edu.cn; wldu@ecust.edu.cn).

Chengqing Zong is with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: cqzong@nlpr.ia.ac.cn).

Digital Object Identifier 10.1109/TPAMI.2026.3665645

0162-8828 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See <https://www.ieee.org/publications/rights/index.html> for more information.

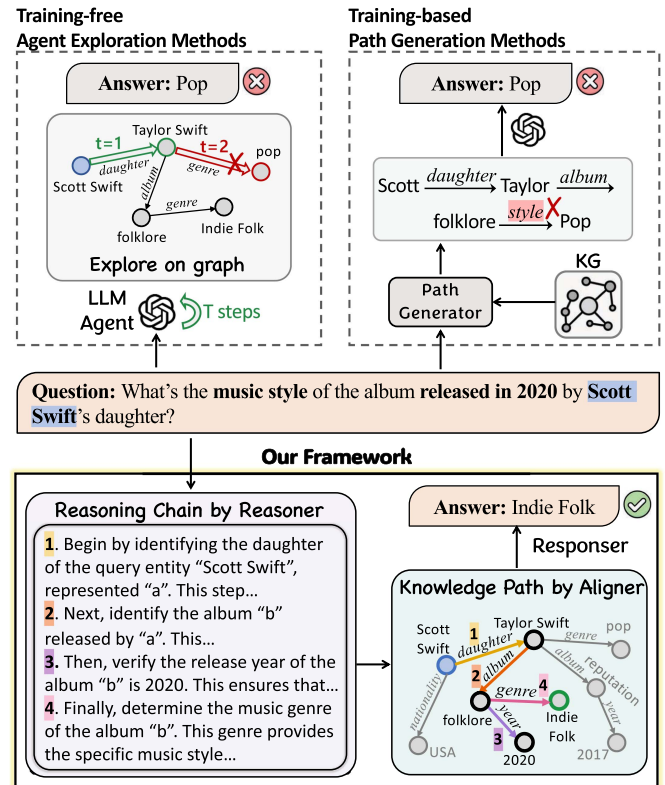


Fig. 1. The comparison between our Reason-Align-Respond framework and the existing methods for LLM-based KGQA.

a semantic network of interconnected information [3]. Over the past decade, knowledge graph question answering (KGQA) has emerged as a popular research area, focusing on utilizing KGs to answer user queries. Now, in the era of LLMs, KGs—with their good interpretability and strong knowledge representation abilities—still serve as a reliable source of knowledge to assist LLMs in question answering. The integration of KGs and LLMs in QA systems has gained increasing attention, aiming to alleviate the hallucination problem of LLMs and enhance model interpretability.

Existing LLM-based KGQA studies are primarily divided into two main categories: Training-free Agent Exploration [4] and Training-based Path Generation [5], as illustrated in Fig. 1. The former approach leverages an LLM as an agent that explores the knowledge graph from multiple query-relevant seed entities (initial nodes appearing in the question), pruning paths

to discover the final answer. The latter approach first retrieves question-relevant paths from the KG and then trains a specialized model to generate new knowledge paths for incoming queries. Both types of methods have contributed significantly to advancing the field of LLM-based KGQA. However, due to the semantic gap between natural language reasoning and structured queries over knowledge graphs, it is often challenging for the explored or generated knowledge paths to maintain logical coherence or semantic relevance with respect to the original question. As shown in Fig. 1, given the question “What’s the music style of the album released in 2020 by Scott Swift’s daughter?”, the agent exploration methods attempt to determine Taylor Swift’s genre directly, failing to first retrieve the specific album entity mandated by the temporal constraint; the path generation methods incorrectly use the predicate “style” for the query. Although “style” is semantically similar, its use deviates from the knowledge graph’s canonical “genre” predicate, which results in a semantically imprecise query.

As illustrated in Fig. 1, effective problem-solving typically involves global planning and the formulation of an overall strategy before retrieving relevant factual evidence from the knowledge graph to derive the correct answer. Recent agent-based methods [6] have made notable strides in approximating this holistic reasoning process through prompt-driven decomposition and self-correction. However, these approaches predominantly operate as inference-time strategies, relying on the inherent capabilities of frozen LLMs. There remains a significant opportunity to further establish a principled probabilistic framework that can explicitly optimize these reasoning chains, ensuring they are not just simulated via prompting but fundamentally aligned with knowledge constraints during training. To this end, we propose integrating such step-by-step reasoning chains with structured knowledge path constraints—modeled as latent variables—to systematically reduce incorrect exploration and mitigate hallucinations. This probabilistic synergy ensures that the reasoning process is grounded in a rigorous mathematical formulation, ultimately leading to more accurate and trustworthy question answering.

On the other hand, the recent deep reasoning LLMs like OpenAI-o1 [7] and DeepSeek-R1 [8], generate holistic reasoning chains before answering challenging questions, demonstrating stronger logical reasoning abilities. However, the reasoning chains are often verbose, may contain logical fallacies, and can even amplify hallucinations [9]. While our work focuses on KGQA rather than directly aiming to improve deep reasoning LLMs, we believe that if the reasoning process is provided with knowledge from KGs as constraints, and grounded in such explicit, structured knowledge, these issues may potentially be alleviated.

The aforementioned two aspects are complementary. For one thing, allowing LLMs to first perform human-like reasoning in KGQA can establish a structured, goal-oriented framework that helps reduce incorrect or noisy KG path exploration. For another, while free reasoning by LLMs tends to increase hallucinations, incorporating knowledge paths from KGs provides constraints that help ensure alignment with the knowledge graph, thereby reducing hallucinations. To this end, in this work, we introduce

Reason-Align-Respond (RAR), a new framework that systematically integrates three modules—Reasoner, Aligner, and Responder—to align LLM reasoning with knowledge graphs for KGQA. First, the Reasoner conducts step-by-step reasoning for the question, simulating a human-like thinking process to generate a reasoning chain that guides the LLM’s exploration on the KG. Next, the Aligner decodes a knowledge path on the KG based on the above reasoning chain. Each decoding step ensures correspondence to an actual knowledge triple in the graph. Finally, the Responder leverages both the reasoning chain and the knowledge path to generate the final answer. Each of these three modules is implemented as a fine-tuned LLM and jointly optimized for effective collaboration, making the reasoning chain explicitly grounded in the KG and driving the Responder to generate reliable and interpretable answers.

Specifically, we treat the reasoning chain z_r and the knowledge path z_p as latent variables, use a latent variable mixture model to formalize the distribution of the answer a given the question q and KG $\mathcal{G}$, and employ an Expectation-Maximization (EM) based learning algorithm that iteratively updates the model parameters and infers the latent variables. The E-step estimates the posterior probability of latent variables (z_r, z_p) given observations ($q, \mathcal{G}, a$) and samples high-quality reasoning chains and aligned knowledge paths accordingly. The M-step maximizes the evidence lower bound (ELBO) of the log-likelihood to update model parameters. Through iterative optimization, the model progressively refines the reasoning chain and knowledge path, and in turn promotes the generation of high-quality responses, ultimately forming a stable closed loop.

We conduct extensive evaluation across multiple benchmarks, including WebQSP, CWQ, GrailQA, CommonsenseQA (CSQA), and MedQA, utilizing Freebase, ConceptNet and a medical KG as knowledge graphs. On WebQSP and CWQ, compared with 19 baseline methods, our approach demonstrates superior effectiveness and achieves state-of-the-art KGQA results. Specifically, it achieves Hit scores of 93.3% on WebQSP and 91.0% on CWQ, with corresponding F1 scores of 87.7% and 84.8%, significantly surpassing the existing methods. On GrailQA, RAR achieves Hit score of 85.5% in Overall, demonstrating strong performance under I.I.D., compositional, and zero-shot settings. Additionally, RAR demonstrates strong zero-shot generalizability on CSQA and MedQA. Human evaluation of reasoning chains shows that RAR generates high-quality, interpretable chains aligned with KG paths, ensuring accurate and reliable reasoning. Notably, RAR achieves strong performance while maintaining computational efficiency during inference. These comprehensive results demonstrate that our approach successfully bridges the gap between LLM reasoning and structured knowledge, offering a promising direction for developing more reliable and interpretable question answering systems.

II. RELATED WORK

Recent advances in large language models (LLMs) have catalyzed significant research aimed at enhancing deep reasoning capabilities. A prominent line of work investigates prompting-based methods that elicit intermediate reasoning steps during

inference. Notably, Chain-of-Thought (CoT) prompting [10] encourages models to articulate step-by-step solutions, improving their performance on complex reasoning tasks. Building upon CoT, the self-consistency approach [11] generates multiple reasoning traces and selects the most consistent answer, thereby increasing reliability. Tree-of-Thought [12] further extends this paradigm by exploring multiple solution paths in a tree structure, which facilitates the discovery of optimal answers through structured exploration. Beyond prompting, researchers have explored fine-tuning LLMs on reasoning-centric datasets [13], [14], including the application of reinforcement learning techniques [7], [8], which encourage models to perform more sophisticated, multi-step reasoning before producing a final response. Collectively, these methods represent significant progress towards improving the reasoning abilities of LLMs.

Despite these advances, LLMs still face notable shortcomings in real-world scenarios. They often struggle to provide verifiable factual evidence, lack access to up-to-date or domain-specific knowledge, and are prone to hallucinations—generating plausible-sounding but incorrect or unverifiable information. These limitations hinder their reliability, especially in tasks that demand factual grounding and interpretability.

To address these issues, a growing body of work seeks to integrate LLMs with knowledge graphs (KGs)—structured repositories that represent real-world entities and their relationships as interconnected nodes and edges [15]. By leveraging KGs, models can ground their reasoning in explicit evidence, reduce hallucinations, and improve performance on knowledge-intensive tasks [3], [16], [17], [18], [19], [20]. Existing approaches to KG-augmented question answering (KGQA) can be broadly categorized into two main paradigms, each with distinct advantages and limitations.

A. Agent Exploration Methods

Methods in this category, such as the pioneering Think-on-Graph (ToG) [4] and other works [21], [22], [23], [24], conceptualize the LLM as an agent that iteratively navigates the KG. ToG stands out as the seminal work in this direction, being the first to successfully integrate an agent with a knowledge graph. It demonstrated that an LLM possesses the inherent capability to autonomously traverse the structured pathways of a KG, proving that LLMs can go beyond their embedded knowledge and interact with complex external symbolic systems. Recent works have addressed the limitations of greedy search in early agentic methods by incorporating advanced planning and search strategies. For instance, Plan-on-Graph [6] introduces a self-correcting adaptive planning framework where the agent employs a reflection mechanism to backtrack when reasoning errors are detected. KBQA-o1 [25] integrates Monte Carlo Tree Search (MCTS) to simulate outcomes and self-evolve its policy, overcoming local optima. Additionally, Paths-over-Graph [26] proposes a training-free dynamic exploration strategy that utilizes structural pruning. In the general framework of these methods, the model issues textual sub-goals or API calls, receives symbolic feedback, and determines subsequent actions, thereby constructing an answer path through sequential decision-making. While this

approach yields interpretable reasoning trajectories, it typically depends on intricate prompt or workflow engineering and necessitates multiple queries to large LLMs, resulting in instability, high latency, and increased computational cost. In contrast, RAR replaces such ad-hoc prompt chains with a principled Expectation-Maximization (EM) procedure that jointly trains a lightweight, natural language-based Reasoner and a KG-aware Aligner. This enables more accurate resolution of multi-hop and conjunctive queries, while significantly improving inference efficiency.

B. Path Generation Methods

Path generation methods, such as the pioneering Reasoning-on-Graphs (RoG) [5] and other works [27], [28], [29], [30], frame KG question answering as a sequence prediction task. As a foundational approach in this direction, RoG innovatively fine-tunes a large language model to learn the structured information inherent in the knowledge graph, enabling it to generate a linear chain of KG relations. This method significantly reduces the reliance on massive-scale language models and hallucinations by directly leveraging the KG's structure. GCR [31] further enhances factual faithfulness by more effectively grounding the generated answers in KG paths, thereby further minimizing hallucinations. However, representing complex queries with a single relation string can be limiting; for instance, queries such as “presidents who were also actors” require the simultaneous satisfaction of multiple constraints, which linear paths struggle to express. Additionally, supervision based solely on shortest paths can introduce noise or fail to capture the semantic intent of the query. RAR addresses these limitations by maintaining a human-interpretable chain of natural language sub-goals, each grounded to specific KG triples. The EM algorithm iteratively discovers latent reasoning chains that are both alignable and answer-bearing, enabling robust handling of conjunctive and compositional queries and resulting in notable quality improvements.

C. Graph-Enhanced RAG From Unstructured Corpora

Beyond the aforementioned KGQA paradigms that operate on pre-defined static KGs, a parallel and influential line of research focuses on constructing KGs from raw text to enhance Retrieval-Augmented Generation (RAG) [32], [33], [34], [35], [36], [37]. Unlike traditional RAG systems that retrieve flat text chunks, these methods leverage the topological structure of self-constructed graphs to address global or abstract queries. Notably, GraphRAG [33] introduces hierarchical community detection to summarize “global sensemaking” themes across an entire corpus. To improve efficiency, LightRAG [34] proposes a dual-level retrieval mechanism that indexes both low-level entities and high-level conceptual keys, while HippoRAG [35] draws inspiration from neurobiology to enable single-step multi-hop retrieval via Personalized PageRank. While our work utilizes a static expert KG, it aligns with these approaches on a core methodological concern: the necessity of *grounding* LLM generation in explicit *symbolic structures* to *control hallucinations*. However, a key distinction lies in the utility of the graph: whereas

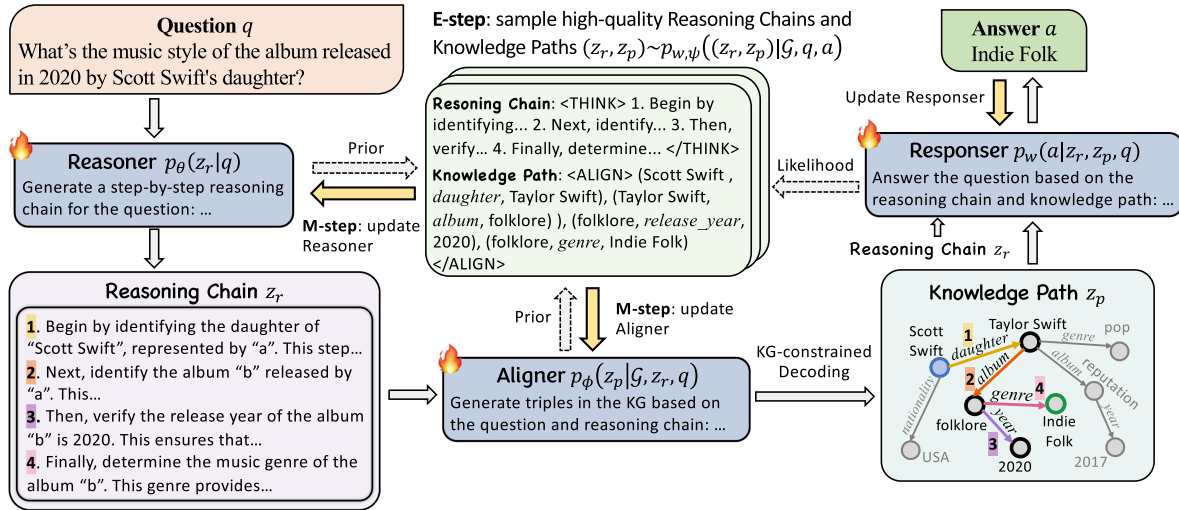


Fig. 2. Illustration of our RAR. First, Reasoner generates a latent reasoning chain z_r to address the question q . Second, Aligner takes the question q and the reasoning chain z_r as inputs to explore KG $\mathcal{G}$, producing a latent reasoning path z_p that aligns with z_r . Finally, Responder generates the final answer a by synthesizing the question q , reasoning chain z_r , and knowledge path z_p .

approaches like GraphRAG construct dynamic graphs to facilitate *summarization* and “global sensemaking” over unstructured corpora, RAR specifically targets *rigorous multi-hop reasoning* over pre-defined high-quality KGs. We leverage the graph topology to impose strict logical constraints on the reasoning process, rather than primarily for content aggregation or coverage.

In summary, integrating LLMs with KGs holds great promise for mitigating hallucinations, grounding answers in factual evidence, and enhancing reasoning over complex queries. However, realizing this potential requires principled approaches that combine the interpretability and efficiency of agent-based navigation with the expressiveness and robustness of path generation—challenges that RAR is designed to address.

III. APPROACH

Knowledge graphs (KGs) contain extensive factual knowledge in the form of triples: $\mathcal{G} = \{(e, r, e') | e, e' \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathcal{E}$ and $\mathcal{R}$ denote sets of entities and relations, respectively. The goal of knowledge graph question answering (KGQA) is to retrieve relevant facts from a KG $\mathcal{G}$ and generate an answer a in response to a given natural language question q .

A. Task Formalization

We introduce two latent variables, a reasoning chain z_r and a knowledge path z_p , working together to answer the question q based on a KG $\mathcal{G}$:

- Reasoning Chain z_r denotes a chain of discrete reasoning steps expressed in natural language, working together to address the question q .
- Knowledge Path z_p denotes an interconnected path of knowledge triples extracted from a KG $\mathcal{G}$.

Since neither z_r nor z_p are explicitly annotated in existing KGQA benchmarks, we treat them as latent variables, and employ a probabilistic mixture model to formalize the distribution of the answer a conditioned on the question q and

KG $\mathcal{G}$ as:

$$p_{w,\phi,\theta}(a|G, q) = \sum_{z_r, z_p} p_w(a|z_r, z_p, q) p_\phi(z_p|G, z_r, q) p_\theta(z_r|q), \quad (1)$$

where we assume a is conditionally independent of $\mathcal{G}$ given (z_r, z_p, q) , allowing factorization and summation over these conditional probabilities. On this basis, we introduce our framework, Reason-Align-Respond (RAR), that integrates three modules—Reasoner, Aligner, and Responder—to align LLM reasoning with knowledge graphs for KGQA, as illustrated in Fig. 2.

First, the **Reasoner** $p_\theta(z_r|q)$ generates a latent reasoning chain z_r to address the question q . The Reasoner generates a reasoning chain in the following format:

$$z_r = \langle \text{THINK} \rangle s_1 \dots s_t \langle \text{/THINK} \rangle,$$

where $\langle \text{THINK} \rangle$ and $\langle \text{/THINK} \rangle$ are special tokens denoting the start and end of the reasoning process, and each s_i is a discrete reasoning step.

Second, the **Aligner** $p_\phi(z_p|G, z_r, q)$ takes the question q and the reasoning chain z_r as inputs to explore KG $\mathcal{G}$, producing a latent knowledge path z_p that aligns with z_r . Aligner processes the prompt with q and z_r to generate a knowledge path in the following format:

$$z_p = \langle \text{ALIGN} \rangle \pi_1 \dots \pi_t \langle \text{/ALIGN} \rangle,$$

where $\langle \text{ALIGN} \rangle$ and $\langle \text{/ALIGN} \rangle$ mark the beginning and end of the knowledge path. Each π_i is a triple from the KG $\mathcal{G}$ formatted as $\pi_i = \langle \text{TRI} \rangle (e_i^h, r_i, e_i^t) \langle \text{/TRI} \rangle$, where $\langle \text{TRI} \rangle$ and $\langle \text{/TRI} \rangle$ bound the triple.

Finally, the **Responder** $p_w(a|z_r, z_p, q)$ generates the final answer a by synthesizing the question q , reasoning chain z_r , and knowledge path z_p .

In practice, we unify Reasoner and Aligner by merging their latent variables z_r and z_p into a single one $z = (z_r, z_p)$. This results in a consolidated module, referred to as **ReAligner**, whose parameters are denoted by ψ . With this simplification, the conditional distribution for the final answer a given a question q and a KG $\mathcal{G}$ is:

$$p_{w,\psi}(a|\mathcal{G}, q) = \sum_z p_w(a|q, z) p_\psi(z|\mathcal{G}, q), \quad (2)$$

where $z = (z_r, z_p)$ is the unified latent variable.

It is important to note that the two components in the mixture model, the ReAligner $p_\psi(z|\mathcal{G}, q)$ and the Responder $p_w(a|q, z)$, are modeled as two LLMs, and are optimized through instruction tuning. Instead of generating z_r and z_p separately, the ReAligner $p_\psi(z|\mathcal{G}, q)$ simultaneously outputs both a reasoning chain z_r and a knowledge path z_p .

B. Optimizing the Mixture Model With the EM Algorithm

Building on the mixture model established in Eq. (2), our goal is to learn the parameters (w, ψ) . A key challenge, however, is that our model's core component—the unified latent variable z —is not present in standard KGQA datasets, making direct supervised training impossible. To address this, we formulate the optimization as a maximum likelihood estimation (MLE) problem with latent variables and employ the Expectation-Maximization (EM) algorithm, a principled framework for this setting. This approach finds precedent in many classical probabilistic mixture models with latent variables, such as Gaussian Mixture Models (GMMs) [38], Probabilistic Latent Semantic Analysis (PLSA) [39], and recent neuro-symbolic reasoning methods [40], [41]. These models often exhibit a distributional form analogous to Eq. (2) and are conventionally optimized via the EM algorithm [42].

The primary distinction in our work is that the two components of our mixture model—ReAligner and Responder—are instantiated as LLMs. Consequently, the parameters we optimize correspond to the parameters of these LLMs. To the best of our knowledge, this represents the first application of the EM algorithm to finetune a mixture model comprised of LLMs.

Given the observed training data $\{(a_i, q_i, \mathcal{G})\}_{i=1}^M$, we aim to learn the parameters (w, ψ) by maximizing the log-likelihood of the training data with respect to Eq. (2), written as:

$$\max_{w,\psi} \mathcal{L}(w, \psi) = \sum_i \log \left(\sum_z p_w(a_i|q_i, z) p_\psi(z|\mathcal{G}, q_i) \right), \quad (3)$$

which is hard to optimize directly due to the log-sum structure.

According to Jensen's inequality, we have:

$$\begin{aligned} \mathcal{L}(w, \psi) &= \sum_i \log \sum_z q_i(z) \frac{p_w(a_i|q_i, z) p_\psi(z|\mathcal{G}, q_i)}{q_i(z)} \\ &\geq \underbrace{\sum_i \sum_z q_i(z) \log \left(\frac{p_w(a_i|q_i, z) p_\psi(z|\mathcal{G}, q_i)}{q_i(z)} \right)}_{\mathcal{L}_{\text{ELBO}}}, \quad (4) \end{aligned}$$

where $q_i(z)$ is a variational distribution. The second line provides a lower bound for $\mathcal{L}(w, \psi)$, which is known as the Evidence Lower Bound (ELBO), $\mathcal{L}_{\text{ELBO}}$. Equality holds when $q_i(z) = p_{w,\psi}(z|\mathcal{G}, q_i, a_i)$, the true posterior of z .

The fundamental idea of the EM algorithm is to maximize $\mathcal{L}(w, \psi)$ indirectly by iteratively maximizing its lower bound $\mathcal{L}_{\text{ELBO}}$, which transforms the original log-sum structure into a sum of log terms, making the optimization more tractable. The EM algorithm alternates between an Expectation step (E-step) and a Maximization step (M-step) until convergence.

E-step: Given current parameters $(w^{(t)}, \psi^{(t)})$ at iteration t , in the E-step, the variational distribution $q_i^{(t)}(z)$ is set to be the posterior of z given $(q_i, a_i, \mathcal{G})$:

$$\begin{aligned} q_i^{(t)}(z) &= p_{w^{(t)}, \psi^{(t)}}(z|\mathcal{G}, q_i, a_i) \\ &= \frac{p_{w^{(t)}}(a_i|q_i, z) p_{\psi^{(t)}}(z|\mathcal{G}, q_i)}{\sum_{z'} p_{w^{(t)}}(a_i|q_i, z') p_{\psi^{(t)}}(z'|\mathcal{G}, q_i)}, \quad (5) \end{aligned}$$

so that $\mathcal{L}_{\text{ELBO}} = \mathcal{L}(w, \psi)$ under the parameters $(w^{(t)}, \psi^{(t)})$.

M-step: Keeping $q_i^{(t)}(z)$ fixed, the M-step maximizes $\mathcal{L}_{\text{ELBO}}$ in Eq. (4) with respect to w and ψ . Ignoring terms that do not depend on (w, ψ) , the problem reduces to:

$$\begin{aligned} \max_{w,\psi} \mathcal{L}_{\text{ELBO}}(w, \psi) \\ = \sum_i \sum_z q_i^{(t)}(z) \log (p_w(a_i|q_i, z) p_\psi(z|\mathcal{G}, q_i)). \quad (6) \end{aligned}$$

where $q_i^{(t)}(z)$ is computed in the E-step (Eq. (5)).

This optimization naturally decomposes into separate instruction-tuning parameter updates for the Responder and ReAligner modules respectively:

$$w^{(t+1)} = \arg \max_w \sum_i \sum_z q_i^{(t)}(z) \log p_w(a_i|q_i, z), \quad (7)$$

$$\psi^{(t+1)} = \arg \max_\psi \sum_i \sum_z q_i^{(t)}(z) \log p_\psi(z|\mathcal{G}, q_i). \quad (8)$$

C. Iterative Training Process in the RAR Framework

Till now, we have provided the theoretical derivation of the EM algorithm for modeling Eq. (2). In this subsection, we describe how this algorithm is implemented within the RAR framework. As we have mentioned, in our framework, both components of the mixture model (ReAligner and Responder) are instantiated as LLMs. The EM algorithm is used to iteratively update the two LLM modules. During each iteration, the E-step infers the most likely latent contexts conditioned on current model parameters, while the M-step alternately finetunes the Responder and ReAligner parameters through instruction-tuning based on these inferred contexts. This iterative process enables both modules to progressively improve their reasoning and knowledge alignment capabilities.

Concretely, each iteration consists of three steps: (1) updating the Responder parameters w using contexts sampled from ReAligner in the M-step; (2) inferring the most likely latent

Algorithm 1: The Learning Algorithm of RAR.

Require: Knowledge Graph $\mathcal{G}$, training set $\{(q_i, a_i)\}$, max iterations T , number of sampling candidates N , number of high-quality samples K , learning rates η_w, η_ψ

- 1: Initialize parameters $w^{(0)}, \psi^{(0)}$
- 2: **for** $t = 1$ to T **do**
- 3: **for all** sample $(\mathcal{G}, q_i, a_i)$ in the training set **do**
- 4: // *M-step: Update the Responder* $p_w(a|q, z)$
- 5: Sample N candidate Reasoning Chains and Knowledge Paths $\{z_i^n\}_{n=1}^N$ from the ReAligner p_ψ based on Eq. (9).
- 6: Update the Responder parameters to $w^{(t)}$ using all N candidates (M-step for w) according to Eq. (10).
- 7: // *E-step: Sample high-quality candidates*
- 8: Compute the posterior as the score of the candidates, using the updated p_w and current p_ψ based on posterior in Eq. (11).
- 9: Compute the sample weights for the candidates with Eq. (12).
- 10: // *M-step: Update the ReAligner* $p_\psi(z|\mathcal{G}, q)$
- 11: Select the top K high-quality candidates $\{z_i^k\}$.
- 12: Update the ReAligner parameters to $\psi^{(t)}$ using a weighted log-likelihood from the high-quality candidates according to Eq. (13)
- 13: **end for**
- 14: **end for**
- 15: **return** Final parameters for the Responder ($p_{w^{(T)}}$) and the ReAligner ($p_{\psi^{(T)}}$)

contexts via the E-step; and (3) finetuning ReAligner parameters ψ on these inferred contexts in the M-step. This cycle is repeated until convergence or for a fixed number of steps, continually enhancing both reasoning quality and knowledge alignment. The overall learning workflow in RAR is summarized in Algorithm 1.

1) *Instruction Tuning the Responder Via the M-Step:* At each iteration, we first update Responder using candidate z to ensure it can fully utilize high-quality z to increase the probability of generating the correct answer. This step is crucial, as only a powerful Responder can ensure the sampling of high-quality z in the subsequent E-step for ReAligner.

For each training example $(\mathcal{G}, q_i, a_i)$, we sample N candidate Graph-aware Reasoning Chains with:

$$z_i^n \sim p_\psi(z|\mathcal{G}, q_i), \quad (9)$$

where $n = 1, \dots, N$.

These candidates, denoted as $\hat{z}_i = \{z_i^1, \dots, z_i^N\}$, are used to approximate the marginal likelihood objective in Eq. (7):

$$\sum_z q_i^{(t)}(z) \log p_w(a_i|q_i, z) \approx \frac{1}{N} \sum_{n=1}^N \log p_w(a_i|q_i, z_i^n). \quad (10)$$

We then update Responder parameters w using gradients from this approximation, making $p_w(a_i|q_i, z)$ more likely to generate the correct answer for the sampled z . In practice, this is implemented with standard instruction tuning of LLMs. To ensure the performance of Responder, we also design a cold starting

strategy for Responder in advance, which will be illustrated in Section IV.

2) *Inferring Latent Contexts Via the E-Step:* The E-step for ReAligner samples high-quality Graph-aware Reasoning Chains. This is done by computing the posterior $p_{w,\psi}(z|\mathcal{G}, q_i, a_i) \propto p_w(a_i|q_i, z)p_\psi(z|\mathcal{G}, q_i)$ using Eq. (5), for each candidate z_i^n , and ranking or sampling candidates by their score

$$S(z_i^n) = \log p_w(a_i|q_i, z_i^n) + \log p_\psi(z_i^n|\mathcal{G}, q_i). \quad (11)$$

The normalized weight for each candidate is obtained via the Softmax function as follows:

$$\pi_{z_i^n} = p_{w,\psi}(z_i^n|\mathcal{G}, q_i, a_i) = \frac{\exp(S(z_i^n))}{\sum_{j=1}^N \exp(S(z_i^j))}. \quad (12)$$

3) *Instruction Tuning ReAligner Via the M-Step:* M-step for ReAligner retains only z that are most consistent with both the updated Responder and ReAligner, which is a crucial step to ensure only the most promising reasoning chains and knowledge paths at each iteration are used for optimizing ReAligner.

Specifically, we select the top set of high-quality candidates $\{z_i^k\} = \{\text{top-}K \text{ Graph-aware Reasoning Chains}\}$ as pseudo-labels for optimizing ReAligner, and update ψ by maximizing weighted log-likelihood from the high-quality candidates

$$\mathcal{L}(\psi) = \sum_k \pi_{z_i^k} \log p_\psi(z_i^k|\mathcal{G}, q_i), \quad (13)$$

according to Eq. (8), which in practice corresponds to standard instruction tuning of p_ψ on the selected high-quality Graph-aware Reasoning Chains.

4) *The Rationale for EM in Our Framework:* The iterative nature of the EM algorithm is well suited for this progressive refinement. In each iteration, the E-step identifies the most likely or “best” latent reasoning chains produced by the current model that successfully link the question to the correct answer via a feasible KG alignment, effectively evaluating the current reasoning quality. The M-step then updates the parameters of the ReAligner module by training it on these high-quality reasoning chains, encouraging the model to generate more successful chains in subsequent iterations. Through this iterative E-M loop, the system gradually corrects flaws in Reasoning Chains and Knowledge Paths, steadily improving the quality, logical coherence, and KG-alignability of the generated latent reasoning.

For a direct illustration of this effect, Fig. 7 presents a case study where the quality of both the Reasoning Chains and Knowledge Paths visibly improves across successive training iterations. This progressive refinement is further validated by the broader experimental results in Fig. 5. As a result, our framework produces answers that are both higher in quality and more interpretable, while significantly reducing the risk of hallucination. By integrating these practical details, our EM-based optimization ensures that the model iteratively refines its latent reasoning and alignment, leading to more accurate and interpretable answers for KGQA.

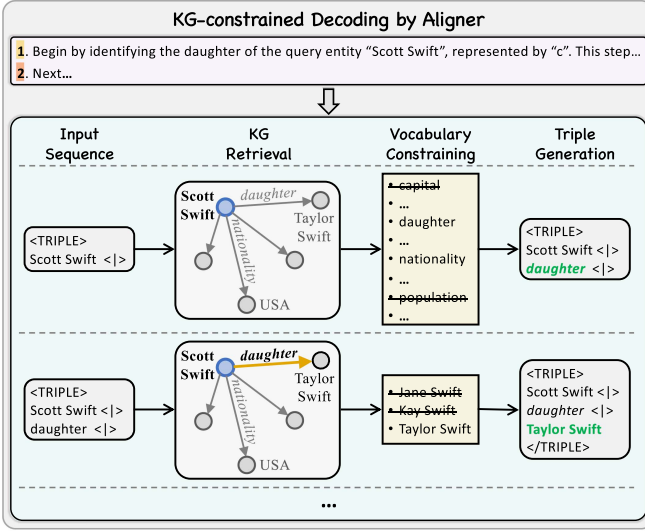


Fig. 3. Illustration of KG-constrained Decoding.

D. Techniques During Inference

During inference, given q , Reasoner generates z_r , while Aligner produces z_p . Responder synthesizes them to produce a . To enhance performance, we introduce three additional key techniques performed during the inference stage.

1) *KG-Constrained Decoding*: KG-constrained Decoding aims to prevent hallucinated triples that do not exist in the KG. When generating the Knowledge Path, Aligner may inadvertently produce triples absent from the KG. To address this, KG-constrained Decoding, as shown in Fig. 3, restricts the output tokens so that only tokens forming valid KG triples can be produced. In this way, the generated Knowledge Path strictly aligns with actual entities and relations in the KG. Related work [31], [43] also attempts to mitigate similar issues; our approach is tailored specifically to our framework.

For efficient KG triples retrieval, we implemented a Virtuoso-based Freebase instance with distributed high-speed SPARQL querying capabilities. Our system achieves a throughput of more than 2,000 requests per second, enabling rapid graph traversal and neighborhood node retrieval during the constrained decoding process. This high-performance infrastructure allows us to efficiently retrieve next-hop candidates for token constraints directly from the knowledge graph.

2) *Knowledge Path Expansion*: Knowledge Path Expansion addresses the potential incompleteness of initially generated Knowledge Paths. To illustrate, consider a question about countries that share borders with the United States. a Knowledge Path:

```
< ALIGN>< TRI> (US,borders,Mexico)
< /TRI>< /ALIGN>
```

is generated by Aligner. While correct, this represents only one instance of a broader pattern. By abstracting the specific instance into a template:

```
< ALIGN>< TRI> (US,borders,?country)
```

```
< /TRI>< /ALIGN>,
```

where $?country$ is a variable, we capture the fundamental relationship structure. Applied to the KG, this template retrieves all valid instances, such as:

```
< ALIGN>< TRI> (US,borders,Canada)
< /TRI>< /ALIGN>.
```

This method transforms a single Knowledge Path into a comprehensive query template, enabling more complete and exhaustive answers.

3) *LLM-Driven Consolidation*: LLM-driven Consolidation addresses the challenge of inconsistencies and noise that emerge when sampling multiple Reasoning Chains and Knowledge Paths. Multiple sampling helps increase coverage and improve the likelihood of correct answers, but inevitably introduces noise and conflicts between samples. To address this challenge, we propose using a powerful LLM as a “Consolidator” that analyzes and integrates multiple Reasoning Chains and Knowledge Paths to derive final answers, aiming to filter out candidates containing incorrect or non-verifiable hops and prioritize fully grounded reasoning chains for aggregation. This approach effectively preserves the benefits of multiple sampling while leveraging the LLM’s analytical capabilities to produce reliable answers.

IV. EXPERIMENT

A. Experiment Settings

Datasets: Following previous research [4], [6], [31], we evaluate our model on three representative KGQA datasets: WebQuestionsSP (WebQSP) [44], Complex WebQuestions (CWQ) [45] and GrailQA [46]. To ensure fair comparison, we adopt identical train and test splits as previous works [5], [47]. All three datasets are based on Freebase [2]. To reduce computational overhead and memory consumption, we utilize the same subgraphs as previous work [5]. Additionally, we preprocess Freebase by converting CVT (Compound Value Type) nodes, which represent n-ary relationships, into binary relationships by concatenating edge labels with “-” as the delimiter, following [43]. We also generalize our method to unseen KG using CommonsenseQA (CSQA) [48] and MedQA [49]. CSQA leverages ConceptNet [50], and MedQA uses a medical KG constructed from the Unified Medical Language System [51], allowing us to assess model generalizability across the unseen KGs.

Baselines: For WebQSP and CWQ, we compare RAR against 19 baselines across three categories: LLM reasoning methods that answer questions without relying on the KG, graph reasoning methods utilizing graph neural networks to learn KG representations, and KG-enhanced LLM reasoning methods including agent exploration and path generation methods. For GrailQA, we compare RAR against 13 baselines under four settings: overall, I.I.D., compositional, and zero-shot, following the standard evaluation protocol.

Evaluation Metrics: For WebQSP and CWQ, we adopt Hit and F1 as evaluation metrics. Hit checks whether the generated predictions match any correct answer, while F1 evaluates overall

answer coverage by balancing precision and recall. For GrailQA, we report Hit following the standard protocol. For the multiple-choice QA datasets CSQA and MedQA, we use accuracy as the evaluation metric.

Implementations: Our implementation uses Llama-3.1-8B [52] as the backbone for Reasoner, Aligner, and Responder. We also conduct extensive experiments with various lightweight LLMs ranging from 0.5 B to 8 B parameters [52], [53], [54]. All models share the same hyperparameter configuration: training for 3 epochs with a batch size of 4 and a learning rate of $2e - 5$. We employ a cosine learning rate scheduler with a warmup ratio of 0.03. The training is performed on 4 A6000 GPUs for each model variant. Regarding the decoding strategy, we adopt distinct mechanisms for training and inference. During the E-step of training, to effectively approximate the expectation with diverse latent variables, we employ temperature sampling to initially generate $N = 25$ candidates, where temperature is 1.2, top_k is 150, and top_p is 0.99. Subsequently, we select the top-10 candidates based on their posterior probability scores for optimization. During the inference stage, we opt for standard Beam Search with a beam size of 10, which consistently yielded superior performance compared to Group Beam Search in our experiments. Consequently, the resulting 10 candidates, generated via KG-constrained Decoding and Knowledge Path Expansion, are directly utilized for LLM-driven Consolidation handled by GPT-4o-mini.

Cold Starting: To initialize the training of Reasoner and Aligner, we leverage high-quality Reasoning Chains and Knowledge Paths derived from SPARQL queries in the WebQSP, CWQ, and GrailQA datasets. This approach prevents the models from generating malformed outputs during early training stages. The SPARQL queries in these datasets represent the gold-standard Reasoning Chain that human experts would use to solve questions using the KG. We decompose these SPARQL queries according to a predefined grammar, breaking them into atomic chunks that each represent a single reasoning step. For each chunk, we query Freebase to obtain the corresponding triples, then use GPT-4o to generate natural language reasoning chains based on these retrieved triples. Through this process, we generate a dataset of 2,000 high-quality Reasoning Chains with their corresponding Knowledge Paths for each question. This dataset enables us to perform cold-start pretraining of Reasoner and Aligner, teaching them to generate well-structured, step-by-step Reasoning Chains and Knowledge Paths with appropriate special tokens, a crucial foundation for subsequent optimization using the EM algorithm.

B. Main Results

Table I shows that RAR achieves significant gains on both WebQSP and CWQ datasets, improving the state-of-the-art by 13.6% and 23.1% in F1, and by 1.1% and 15.2% in Hit respectively. These remarkable improvements demonstrate the effectiveness of integrating structured KG knowledge with LLM reasoning. Remarkably, in Table V, RAR with Llama-2-7B, without KG-constrained Decoding or Knowledge Path Expansion, significantly outperforms previous state-of-the-art methods.

TABLE I
PERFORMANCE COMPARISON WITH DIFFERENT BASELINES ON WEBQSP AND CWQ

Methods	WebQSP		CWQ	
	Hit	F1	Hit	F1
Qwen2-7B [53]	50.8	35.5	25.3	21.6
Llama-2-7B [54]	56.4	36.5	28.4	21.4
Llama-3.1-8B [52]	55.5	34.8	28.1	22.4
GPT-4o-mini [55]	63.8	40.5	63.8	40.5
ChatGPT [56]	59.3	43.5	34.7	30.2
ChatGPT+Few-shot [57]	68.5	38.1	38.5	28.0
ChatGPT+CoT [10]	73.5	38.5	47.5	31.0
ChatGPT+Self-Consistency [11]	83.5	63.4	56.0	48.1
GraftNet [58]	66.7	62.4	36.8	32.7
NSM [59]	68.7	62.8	47.6	42.4
SR+NSM [60]	68.9	64.1	50.2	47.1
ReaRev [61]	76.4	70.9	52.9	47.8
KD-CoT [28]	68.6	52.5	55.7	-
EWEK-QA [62]	71.3	-	52.5	-
ToG (GPT-4) [4]	82.6	-	68.5	-
EffiQA [63]	82.9	-	69.5	-
KBQA-o1 (Llama-3.1-8B) [25]	-	59.8	-	-
RoG (Llama-2-7B) [5]	85.7	70.8	62.6	56.2
GNN-RAG+RA [29]	90.7	73.5	68.7	60.4
Plan-on-Graph (GPT-4) [6]	87.3	-	75.0	-
GCR (Llama-3.1-8B + GPT-4o-mini) [31]	92.2	74.1	75.8	61.7
RAR (Llama-3.1-8B + GPT-4o-mini)	93.3	87.7	91.0	84.8

TABLE II
PERFORMANCE COMPARISON WITH DIFFERENT BASELINES ON GRAILQA

Method	GrailQA			
	Overall	I.I.D.	Compositional	Zero-shot
<i>LLM-Only</i>				
ChatGPT [57]	29.4	-	-	-
ChatGPT+CoT [10]	28.1	-	-	-
ChatGPT+Self-Consistency [11]	29.6	-	-	-
<i>Prompting KG-Augmented LLM w/GPT-3.5 or others</i>				
KB-BINDER [27]	50.6	-	-	-
ToG [4]	68.7	70.1	56.1	72.7
Plan-on-Graph [26]	76.5	76.3	62.1	81.7
<i>Prompting KG-Augmented LLM w/GPT-4</i>				
ToG [4]	81.4	79.4	67.3	86.5
Plan-on-Graph [26]	84.7	87.9	69.7	88.6
<i>Fine-Tuned KG-Augmented LLM</i>				
RnG-KBQA [64]	68.8	86.2	63.8	63.0
TIARA [65]	73.0	87.8	69.2	68.0
FC-KBQA [66]	73.2	88.5	70.0	67.6
Pangu [67]	75.4	84.4	74.6	71.6
FlexKBQA [68]	62.8	71.3	59.1	60.6
GAIN [69]	76.3	88.5	73.7	71.8
KBQA-o1 (Llama-3.1-8B) [25]	71.9	77.8	76.3	68.1
RAR (Llama-3.1-8B)	85.5	90.1	77.7	87.1

We further evaluate RAR on GrailQA, which benchmarks the performance under three settings: I.I.D., Compositional, and Zero-shot. As shown in Table II, RAR (Llama-3.1-8B) achieves a Hit score of 85.5% in Overall, outperforming the recent fine-tuning baseline KBQA-o1 [25]. It also performs strongly on the I.I.D. and Compositional splits (90.1% and 77.7%), and remains competitive on the Zero-shot split compared to Plan-on-Graph [26], which relies on prompting a much larger closed-source GPT-4.

TABLE III
IMPACT OF DIFFERENT COMPONENTS ON CWQ

Variants	Hit	F1	Precision	Recall
RAR	91.0	84.8	84.9	89.0
RAR w/o LC	92.5	83.0	81.8	91.2
w/o Reasoner	94.1	80.3	75.6	94.7
w/o KD	61.3	48.9	54.3	50.8
w/o KPE	84.7	71.7	80.2	71.7

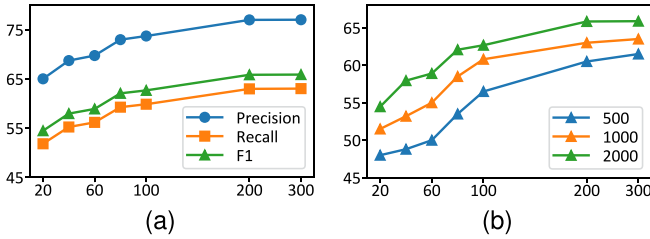


Fig. 4. (a) Impact of iteration steps of the EM algorithm; (b) Different numbers of gold samples for cold starting.

C. Ablation Study

As shown in Table III, removing LLM-driven Consolidation (LC) lowers precision but increases recall, since LC aims to eliminate noisy predictions. Excluding Reasoner causes a pronounced drop in precision but a rise in recall, indicating that Reasoning Chains guide Aligner to explore the KG more accurately, reducing hallucination and noise. Disabling Knowledge Path Expansion (KPE) diminishes performance, confirming its role in enriching Knowledge Paths. Most importantly, removing KG-constrained Decoding (KD) yields the largest performance decrease, underscoring the importance of restricting generation to valid KG paths.

D. Further Analyses

We further systematically investigate how our RAR improves reasoning quality, reliability, and generalizability for KGQA tasks. We analyze the impact of iterative EM updates, the quality of reasoning chains, KG-constrained Decoding, backbone selection, and zero-shot transfer to unseen KGs across multiple evaluation metrics.

1) *Impact of Iteration Steps*: As shown in Fig. 4(a), RAR exhibits consistent improvement across all metrics during EM updates. The performance rises rapidly in early stages and continues to refine over iterations, eventually reaching convergence with minimal fluctuations.

To assess the effect of cold-start dataset size, we initialized our EM-based model with 2,000, 1,000, and 500 samples. As shown in Fig. 4(b), performance consistently improved with larger initial datasets: the 2,000-sample model converged fastest and achieved the highest F1 score, while the 500-sample model showed slower early progress and stabilized at a lower performance level.

2) *Quality of Reasoning Chains*: Through manual evaluation of 500 randomly selected samples, we assess the quality

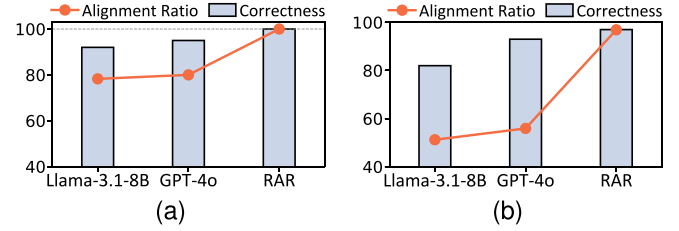


Fig. 5. Human evaluation of reasoning chains on (a) WebQSP and (b) CWQ.

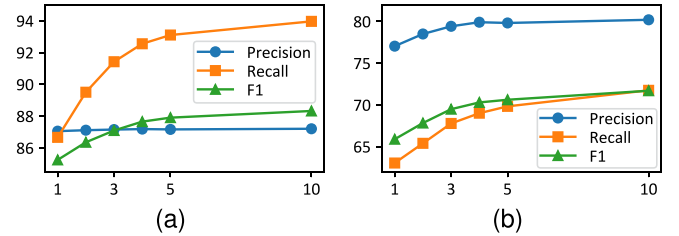


Fig. 6. Impact of different beam size on (a) WebQSP and (b) CWQ.

TABLE IV
EFFICIENCY AND PERFORMANCE OF RAR COMPARED TO DIFFERENT METHODS ON CWQ

Types	Methods	Hit	Avg. Runtime (s)
Path Generation	GNN-RAG	66.8	1.73
	RoG	62.6	2.68
	GCR	75.8	3.72
Agent Exploration	ToG	68.5	18.89
	EffiQA	69.5	-
Ours	RAR	91.0	4.38

of Reasoning Chains using two criteria: reasoning correctness and KG alignment. The correctness metric evaluates whether the Reasoning Chain successfully solves the given question, while the alignment metric measures how well the reasoning steps correspond to valid KG paths. As shown in Fig. 5, RAR substantially outperforms both GPT-4o and baseline methods across both metrics. These results demonstrate that by aligning Reasoning Chains to KG structures, our approach not only improves the reliability of the reasoning process but also enhances its interpretability.

3) *KG-Constrained Decoding Effectiveness*: We examine the effectiveness of KG-constrained Decoding in mitigating hallucinations and maintaining efficiency. Our method achieves *zero hallucinations* in Knowledge Paths when answers are correct, while without constraints, even correct answers show a 44% hallucination rate. The efficiency evaluation reveals minimal computational overhead. Particularly noteworthy is the comparison with GCR, the previous state-of-the-art method using KG constraints. As shown in Table IV, our approach achieves a 15.2% improvement in answer accuracy over GCR, with only a marginal increase in runtime from Reasoning Chain generation. This modest overhead is well justified by the significant gains in answer reliability and interpretability.

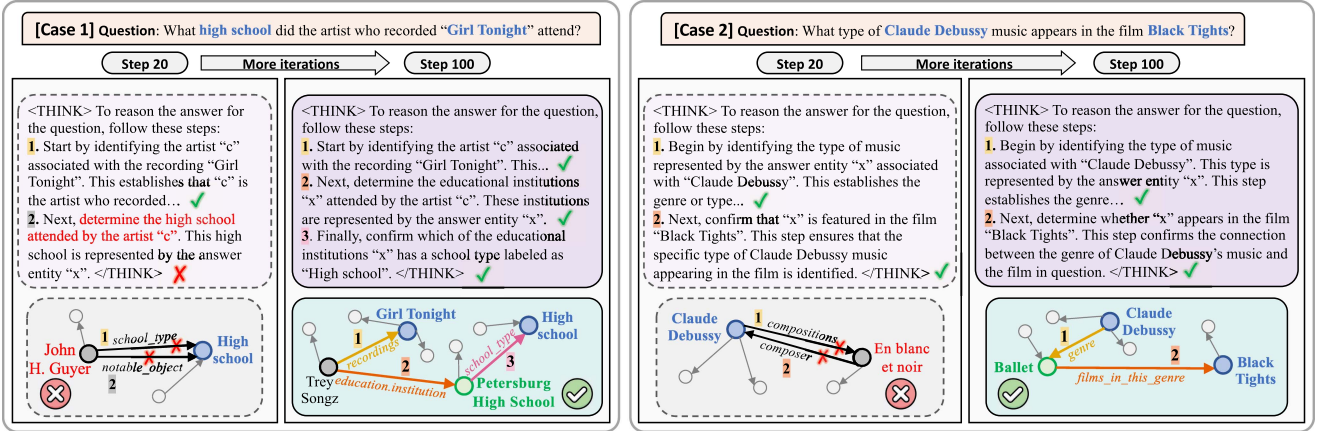


Fig. 7. Examples of Reasoning Chains and Knowledge Paths generated by RAR under different iteration steps.

TABLE V
IMPACT OF USING DIFFERENT LLM BACKBONES FOR REASONER, ALIGNER AND LLM-DRIVEN CONSOLIDATION

Components	Variants	Hit	F1
Reasoner and Aligner	Llama-2-7B	84.0	68.9
	Llama-3.1-8B	85.2	72.6
	Qwen2-0.5B	71.3	56.0
	Qwen2-1.5B	72.0	56.6
	Qwen2-7B	81.4	67.1
LLM-driven Consolidation	GPT-4o-mini	91.0	84.8
	GPT-4o	92.8	84.9
	Qwen2-7B	88.7	82.2
	Llama-3.1-8B	90.6	83.7
	Llama-3.1-70B	92.4	83.0

TABLE VI
ZERO-SHOT TRANSFERABILITY TO UNSEEN KG

Dataset	GPT-4o-mini	GCR	RAR
CSQA [48]	91	94	94
MedQA [49]	75	79	80

4) *Impact of Beam Size:* As shown in Fig. 6, increasing beam size allows RAR to explore more potential Reasoning Chains and Knowledge Paths, leading to improved performance across all metrics. This demonstrates that examining multiple candidate solutions helps identify better Reasoning Chains and Knowledge Paths for responses of higher quality.

5) *Impact of Different LLM Backbone on CWQ:* Table V demonstrates that larger LLMs generally achieve better performance, with Llama-3.1-8B and GPT-4o delivering the strongest results for backbone and LLM-based Consolidation (LC), respectively.

6) *Zero-Shot Generalizability to Unseen KGs:* Following [31], we evaluate RAR's zero-shot transfer capabilities on CSQA [48] and MedQA [49]. To adapt to these unseen domains, we employ a two-step inference strategy. First, we utilize the Reasoner trained on CWQ to generate natural language reasoning chains, based on the premise that the Reasoner has acquired a generalizable capability to plan and decompose problem-solving steps independent of KG schemas. Subsequently, rather than continuing with the trained models, we transition to a raw Llama-3.1-8B base model equipped with KG-constrained Decoding to generate knowledge paths. This design leverages the base model's general ability to strictly adhere to the target KG's specific ontology, avoiding the domain shift issues where the

model fine-tuned on Freebase might struggle with unseen graph structures. The results in Table VI show that RAR achieves superior zero-shot performance compared to GPT-4o-mini on both datasets. Notably, RAR achieves comparable performance to GCR, as both methods effectively leverage the KG structure to constrain the decoding process, enabling precise reasoning without requiring additional training on the target KG.

E. Case Study

To investigate the effect of EM iterations, in Fig. 7, we also present two representative cases that demonstrate the evolution of RAR's behavior across iterations of the EM algorithm. These cases showcase distinct improvements in RAR's ability to generate Reasoning Chains and Knowledge Paths. In Case 1, we examine RAR's response to the query "What high school did the artist who recorded 'Girl Tonight' attend?" The early-stage model demonstrates suboptimal verification behavior by directly searching for high school information without following the proper verification sequence: first identifying educational institutions, then verifying the specific type as high school. This Reasoning Chain prevents proper alignment with the Knowledge Path in the KG. In contrast, the later-stage model generates both Reasoning Chains and Knowledge Paths effectively, producing a higher quality Reasoning Chain that successfully aligns with the Knowledge Path in the KG. Case 2 examines RAR's handling of "What type of Claude Debussy music appears in the film Black Tights?" Here, we observe a different pattern of improvement. While the early-stage model generates the same Reasoning Chain as the later-stage model, it fails to generate Knowledge Paths that fully align with and reflect this reasoning, resulting in a misaligned Knowledge Path that does not lead to the correct



Fig. 8. Examples of Reasoning Chains and Knowledge Paths generated by RAR.

answer. The later-stage model maintains consistency between Reasoning Chains and Knowledge Paths, thus arriving at the correct answer. These cases validate the effectiveness of the EM approach. Fig. 8 illustrates the qualitative results by our RAR framework.

V. CONCLUSION

In this paper, we present RAR, a novel framework that integrates LLM reasoning with knowledge graphs for KGQA through three key components—Reasoner, Aligner, and Responder. We formulate this process as a latent variable mixture model and optimize it using the Expectation-Maximization algorithm. By combining human-like reasoning capabilities of LLMs with the structured constraints of knowledge paths, our framework reduces incorrect KG path exploration and yields more accurate and reliable KGQA results. Through extensive

experiments on multiple benchmarks, we demonstrate that RAR achieves state-of-the-art performance while maintaining interpretability and computational efficiency. These results demonstrate that RAR successfully bridges the gap between LLM reasoning and structured knowledge, offering a promising direction for building reliable and interpretable QA systems.

In the future, we plan to extend this work in two primary directions. First, to further minimize manual annotation efforts in extremely low-resource domains, we aim to develop a fully automated initialization strategy. This approach will leverage strong general-purpose LLMs with KG-constrained decoding to generate candidate knowledge paths and reasoning chains, subsequently applying our EM-based feedback mechanism to iteratively filter and refine these noisy samples. Second, we intend to enhance the framework's robustness against incomplete KG coverage. We will explore fallback mechanisms that bridge reasoning gaps by exploiting the LLM's

intrinsic parametric knowledge or by incorporating external evidence retrieved from unstructured text and dynamically constructed KGs.

REFERENCES

- [1] M. U. Hadi et al., "Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects," TechRxiv, 2023, doi: [10.36227/techrxiv.23589741](https://doi.org/10.36227/techrxiv.23589741).
- [2] K. D. Bollacker, C. Evans, P. K. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Vancouver, BC, Canada, 2008, pp. 1247–1250, doi: [10.1145/1376616.1376746](https://doi.org/10.1145/1376616.1376746).
- [3] K. Liang et al., "A survey of knowledge graph reasoning on graph types: Static, dynamic, and multi-modal," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 9456–9478, Dec. 2024, doi: [10.1109/TPAMI.2024.3417451](https://doi.org/10.1109/TPAMI.2024.3417451).
- [4] J. Sun et al., "Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph," in *Proc. 12th Int. Conf. Learn. Representations*, 2024. [Online]. Available: <https://openreview.net/forum?id=nnVO1PvbTv>
- [5] L. Luo, Y. Li, G. Haffari, and S. Pan, "Reasoning on graphs: Faithful and interpretable large language model reasoning," in *Proc. 12th Int. Conf. Learn. Representations*, ICLR 2024, Vienna, Austria, 2024. [Online]. Available: <https://openreview.net/forum?id=ZGNWW7xZ6Q>
- [6] L. Chen, P. Tong, Z. Jin, Y. Sun, J. Ye, and H. Xiong, "Plan-on-graph: Self-correcting adaptive planning of large language model on knowledge graphs," in *Proc. Adv. Neural Inf. Process. Syst. 38: Annu. Conf. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2024, pp. 37665–37691. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/4254e856d01a5e7b7ea050477c3ef9b9-Abstract-Conference.html
- [7] OpenAI, "Learning to reason with LLMs," 2024. [Online]. Available: <https://openai.com/index/learning-to-reason-with-llms/>
- [8] D. Guo et al., "DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning," 2025, *arXiv:2501.12948*.
- [9] F. Bao, C. Xu, and O. Mendelevitch, "DeepSeek-R1 hallucinates more than deepSeek-V3," 2025, Accessed: Feb. 10, 2025. [Online]. Available: <https://www.vectara.com/blog/deepseek-r1-hallucinates-more-than-deepseek-v3>
- [10] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in *Proc. Adv. Neural Inf. Process. Syst. 35: Annu. Conf. Neural Inf. Process. Syst.*, New Orleans, LA, USA, 2022, pp. 24824–24837. [Online]. Available: http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html
- [11] X. Wang et al., "Self-consistency improves chain of thought reasoning in language models," in *Proc. 11th Int. Conf. Learn. Representations*, Kigali, Rwanda, 2023. [Online]. Available: <https://openreview.net/forum?id=1PL1NIMMrw>
- [12] S. Yao et al., "Tree of thoughts: Deliberate problem solving with large language models," in *Proc. Adv. Neural Inf. Process. Syst. 36: Annu. Conf. Neural Inf. Process. Syst.*, New Orleans, LA, USA, 2023, pp. 11809–11822. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html
- [13] P. Yu et al., "ALERT: Adapting language models to reasoning tasks," 2022, *arXiv:2212.08286*.
- [14] M. D. Hoffman et al., "Training chain-of-thought via latent-variable inference," in *Proc. Adv. Neural Inf. Process. Syst. 36: Annu. Conf. Neural Inf. Process. Syst.*, 2023, pp. 72819–72841, New Orleans, LA, USA. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/e69a9560c450ca76584d9eb37e7f5ae8-Abstract-Conference.html
- [15] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, and X. Wu, "Unifying large language models and knowledge graphs: A roadmap," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 7, pp. 3580–3599, Jul. 2024.
- [16] S. N. Aakur and S. Sarkar, "Leveraging symbolic knowledge bases for commonsense natural language inference using pattern theory," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13185–13202, Nov. 2023.
- [17] W. Wang, Y. Yang, and F. Wu, "Towards data- and knowledge-driven AI: A survey on neuro-symbolic computing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 2, pp. 878–899, Feb. 2025.
- [18] M. Nayyeri, C. Xu, M. M. Alam, J. Lehmann, and H. S. Yazdi, "Logicenn: A neural based knowledge graphs embedding model with logical rules," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 7050–7062, Jun. 2023.
- [19] Y. Zhang, Q. Yao, and J. T. Kwok, "Bilinear scoring function search for knowledge graph learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 1458–1473, Feb. 2023.
- [20] J. Wang, Z. Zhang, Z. Shi, J. Cai, S. Ji, and F. Wu, "Duality-induced regularizer for semantic matching knowledge graph embeddings," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 1652–1667, Feb. 2023.
- [21] S. Cheng et al., "Call me when necessary: LLMs can efficiently and faithfully reason over structured environments," in *Findings of the Assoc. for Comput. Linguist.: ACL 2024*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: ACL, 2024, pp. 4275–4295. [Online]. Available: <https://aclanthology.org/2024.findings-acl.254/>
- [22] Y. Gu et al., "Middleware for LLMs: Tools are instrumental for language agents in complex environments," in *Proc. 2024 Conf. Empirical Methods Natural Lang. Process.*, Eds. Miami, Florida, USA: ACL, 2024, pp. 7646–7663. [Online]. Available: <https://aclanthology.org/2024.emnlp-main.436/>
- [23] X. Huang et al., "QueryAgent: A reliable and efficient reasoning framework with environmental feedback based self-correction," in *Proc. 62nd Annu. Meeting Assoc. Comput. Linguist.*, Bangkok, Thailand: ACL, 2024, pp. 5014–5035. [Online]. Available: <https://aclanthology.org/2024.acl-long.274/>
- [24] Z. Nie, R. Zhang, Z. Wang, and X. Liu, "Code-style in-context learning for knowledge-based question answering," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 17, 2024, pp. 18833–18841.
- [25] H. Luo et al., "KBQA-ol: Agentic knowledge base question answering with Monte Carlo tree search," in *Proc. 42nd Int. Conf. Mach. Learn.*, Vancouver, BC, Canada, 2025, pp. 41177–41199. [Online]. Available: <https://openreview.net/forum?id=QuecSemZly>
- [26] X. Tan et al., "Paths-over-graph: Knowledge graph empowered large language model reasoning," in *Proc. ACM Web Conf.*, Sydney, NSW, Australia, 2025, pp. 3505–3522, doi: [10.1145/3696410.3714892](https://doi.org/10.1145/3696410.3714892).
- [27] T. Li, X. Ma, A. Zhuang, Y. Gu, Y. Su, and W. Chen, "Few-shot in-context learning on knowledge base question answering," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguist.*, Toronto, Canada: ACL, 2023, pp. 6966–6980. [Online]. Available: <https://aclanthology.org/2023.acl-long.385/>
- [28] K. Wang et al., "Knowledge-driven CoT: Exploring faithful reasoning in LLMs for knowledge-intensive question answering," 2023, *arXiv:2308.13259*.
- [29] C. Mavromatis and G. Karypis, "GNN-RAG: Graph neural retrieval for efficient large language model reasoning on knowledge graphs," in *Proc. Findings Assoc. Comput. Linguist.: ACL, 2025*, pp. 16682–16699.
- [30] J. Jiang, K. Zhou, Z. Dong, K. Ye, W. X. Zhao, and J.-R. Wen, "StructGPT: A general framework for large language model to reason over structured data," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2023, pp. 9237–9251.
- [31] L. Luo, Z. Zhao, G. Haffari, Y.-F. Li, C. Gong, and S. Pan, "Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models," in *Proc. 42nd Int. Conf. Mach. Learn.*, 2025, pp. 41540–41565.
- [32] B. Peng et al., "Graph retrieval-augmented generation: A survey," 2024, *arXiv:2408.08921*.
- [33] D. Edge et al., "From local to global: A graph RAG approach to query-focused summarization," 2024, *arXiv:2404.16130*.
- [34] Z. Guo, L. Xia, Y. Yu, T. Ao, and C. Huang, "LightRAG: Simple and fast retrieval-augmented generation," 2024, *arXiv:2410.05779*.
- [35] B. J. Gutierrez, Y. Shu, Y. Gu, M. Yasunaga, and Y. Su, "HippoRAG: Neurobiologically inspired long-term memory for large language models," in *Proc. Adv. Neural Inf. Process. Syst. 38: Annu. Conf. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2024, pp. 59532–59569. [Online]. Available: http://papers.nips.cc/paper_files/paper/2024/hash/6dde001d07ca4f319af96a3024f6dbd1-Abstract-Conference.html
- [36] Z. Li et al., "StructRAG: Boosting knowledge intensive reasoning of LLMs via inference-time hybrid information structurization," in *Proc. 13th Int. Conf. Learn. Representations*, 2025. [Online]. Available: <https://openreview.net/forum?id=GhexuBLxbO>
- [37] L. Liang et al., "KAG: Boosting LLMs in professional domains via knowledge augmented generation," in *Proc. Companion Proc. ACM Web Conf.*, 2025, pp. 334–343.
- [38] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.
- [39] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1999, pp. 50–57.
- [40] P. Sen, B. W. S. R. de Carvalho, R. Riegel, and A. G. Gray, "Neuro-symbolic inductive logic programming with logical neural networks," in *36th AAAI Conf. Artif. Intell.*, AAAI 2022, *34th Conf. Innov. Appl. Artif. Intell.*, *12th Symp. Educ. Adv. Artif. Intell.*, Virtual Event, 2022, pp. 8212–8219, doi: [10.1609/aaai.v36i8.20795](https://doi.org/10.1609/aaai.v36i8.20795).

- [41] M. Qu, J. Chen, L. A. C. Xhonneux, Y. Bengio, and J. Tang, "RNNLogic: Learning logic rules for reasoning on knowledge graphs," in *Proc. 9th Int. Conf. Learn. Representations*, Austria, 2021. [Online]. Available: <https://openreview.net/forum?id=tGZu6DlbreV>
- [42] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc.: Ser. B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.
- [43] K. Li, T. Zhang, X. Wu, H. Luo, J. Glass, and H. Meng, "Decoding on graphs: Faithful and sound reasoning on knowledge graphs through generation of well-formed chains," 2024, *arXiv:2410.18415*.
- [44] W. Yih, M. Richardson, C. Meek, M. Chang, and J. Suh, "The value of semantic parse labeling for knowledge base question answering," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguist.*, Berlin, Germany, 2016, pp. 201–206, doi: [10.18653/v1/p16-2033](https://doi.org/10.18653/v1/p16-2033).
- [45] A. Talmor and J. Berant, "The web as a knowledge-base for answering complex questions," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Hum. Lang. Technol.*, 2018, pp. 641–651.
- [46] Y. Gu et al., "Beyond I.I.D.: Three levels of generalization for question answering on knowledge bases," in *Proc. Web Conf.*, 2021, pp. 3477–3488.
- [47] J. Jiang, K. Zhou, X. Zhao, and J.-R. Wen, "UniKGQA: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph," in *Proc. 11th Int. Conf. Learn. Representations*, 2023. [Online]. Available: <https://openreview.net/forum?id=Z63RvyAZ2Vh>
- [48] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "CommonsenseQA: A question answering challenge targeting commonsense knowledge," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Hum. Lang. Technol.*, 2019, pp. 4149–4158.
- [49] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, and P. Szolovits, "What disease does this patient have? A large-scale open domain question answering dataset from medical exams," *Appl. Sci.*, vol. 11, no. 14, 2021, Art. no. 6421.
- [50] R. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: An open multilingual graph of general knowledge," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4444–4451.
- [51] M. Yasunaga et al., "QA-GNN: Reasoning with language models and knowledge graphs for question answering," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Hum. Lang. Technol.*, 2021, pp. 535–546. [Online]. Available: <https://aclanthology.org/2021.naacl-main.45/>
- [52] Meta, "Build the future of AI with meta Llama 3," 2024. [Online]. Available: <https://llama.meta.com/llama3/>
- [53] A. Yang et al., "Qwen2 technical report," 2024, *arXiv:2407.10671*.
- [54] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," 2023, *arXiv:2307.09288*.
- [55] OpenAI, "Hello GPT-4o," 2024. [Online]. Available: <https://openai.com/index/hello-gpt-4o/>
- [56] OpenAI, "Introducing chatGPT," 2022. [Online]. Available: <https://openai.com/index/chatgpt/>
- [57] T. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1877–1901.
- [58] H. Sun, B. Dhingra, M. Zaheer, K. Mazaitis, R. Salakhutdinov, and W. Cohen, "Open domain question answering using early fusion of knowledge bases and text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4231–4242.
- [59] G. He, Y. Lan, J. Jiang, W. X. Zhao, and J.-R. Wen, "Improving multi-hop knowledge base question answering by learning intermediate supervision signals," in *Proc. 14th ACM Int. Conf. Web Search Data Mining*, 2021, pp. 553–561.
- [60] J. Zhang et al., "Subgraph retrieval enhanced model for multi-hop knowledge base question answering," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguist.*, 2022, pp. 5773–5784.
- [61] C. Mavromatis and G. Karypis, "ReaRev: Adaptive reasoning for question answering over knowledge graphs," in *Proc. Findings Assoc. Comput. Linguist.: EMNLP* 2022, 2022, pp. 2447–2458.
- [62] M. Dehghan et al., "EWEK-QA: Enhanced web and efficient knowledge graph retrieval for citation-based question answering systems," in *Proc. 62nd Annu. Meeting Assoc. Comput. Linguist.*, Eds. Bangkok, Thailand, 2024, pp. 14169–14187. [Online]. Available: <https://aclanthology.org/2024.acl-long.764>
- [63] Z. Dong et al., "EffiQA: Efficient question-answering with strategic multi-model collaboration on knowledge graphs," in *Proc. 31st Int. Conf. Comput. Linguist.*, Abu Dhabi, UAE, 2025, pp. 7180–7194. [Online]. Available: <https://aclanthology.org/2025.coling-main.479/>
- [64] X. Ye, S. Yavuz, K. Hashimoto, Y. Zhou, and C. Xiong, "RNG-KBQA: Generation augmented iterative ranking for knowledge base question answering," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguist.*, 2022, pp. 6032–6043.
- [65] Y. Shu et al., "TIARA: Multi-grained retrieval for robust question answering over large knowledge base," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2022, pp. 8108–8121.
- [66] L. Zhang et al., "FC-KBQA: A fine-to-coarse composition framework for knowledge base question answering," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguist.*, 2023, pp. 1002–1017.
- [67] Y. Gu, X. Deng, and Y. Su, "Don't generate, discriminate: A proposal for grounding language models to real-world environments," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguist.*, 2023, pp. 4928–4949.
- [68] Z. Li et al., "FlexKBQA: A flexible LLM-powered framework for few-shot knowledge base question answering," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 18608–18616.
- [69] Y. Shu and Z. Yu, "Distribution shifts are bottlenecks: Extensive evaluation for grounding language models to knowledge bases," in *Proc. 18th Conf. Eur. Chapter Assoc. Comput. Linguistics: Student Res. Workshop*, 2024, pp. 71–88.



Xiangqing Shen received the BS degree from Southeast University, China, in 2016, the MS degree from the China University of Mining and Technology, in 2020, and the PhD degree from the Nanjing University of Science and Technology, in 2025. He is currently an assistant professor in the School of Intelligence Science and Technology, Nanjing University, China. His research interests include natural language processing, large language models, multimodal learning, and knowledge graphs. His work on dense commonsense knowledge graph has won the ACL 2023 outstanding paper award.



Fanfan Wang received the BS degree in automation from the Nanjing University of Science and Technology, Nanjing, China, in 2019, and the PhD degree from the Nanjing University of Science and Technology, in 2025. Her research interests include natural language processing and affective computing.



Zinong Yang received the BS and MS degrees in computer science from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2022. He is currently working toward the PhD degree in the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests include natural language processing and financial artificial intelligence.



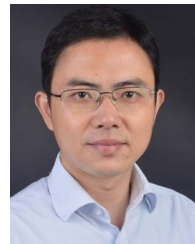
Bing Wang received the BS and PhD degree in chemical engineering from Tsinghua University, China, in 2011 and 2016, respectively. He is now an associate professor in East China University of Science and Technology, Shanghai. His research interests include atmospheric dispersion modeling, quantitative risk assessment to chemical spills, source term estimation for air pollution dispersion, knowledge graph and process safety analysis.



Chengqing Zong (Fellow, IEEE) received the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, China, in 1998. He is currently a professor with the Institute of Automation, Chinese Academy of Sciences, China. He is serving as the president of the Association for Computational Linguistics (ACL). He previously served as the president of the Asian Federation of Natural Language Processing from 2019 to 2021. He also served as Program Committee Co-Chair for top-tier international conferences such as ACL 2015 and COLING 2020, and as General Conference Chair for ACL 2021. He is a Fellow of IEEE/ACL/AAIA/CAAI/CCF, and a member of the Academia Europaea.



Wenli Du received the PhD degree in automation from ECUST, in 2005. She is a full professor in School of Information Science and Engineering, associate vice president and Head of Graduate School with the East China University of Science and Technology (ECUST). She serves as Dean of National Technology Innovation Center for Smart Process Manufacturing, and leads research in industrial process modelling, control, optimization, and AI/ML applications.



Rui Xia received the BS degree from Southeast University, China, in 2004, the MS degree from East China University of Science and Technology, in 2007, and the PhD degree from the Institute of Automation, Chinese Academy of Sciences, in 2011. He is currently a full professor in the School of Intelligence Science and Technology, Nanjing University, China. His research interests include natural language processing, large language models, text mining and affective computing. He has won the ACL 2019 and ACL 2023 Outstanding Paper Awards.