

Think-on-Graph 3.0: Efficient and Adaptive LLM Reasoning on Heterogeneous Graphs via Multi-Agent Dual-Evolving Context Retrieval

Anonymous ACL submission

Abstract

Graph-based Retrieval-Augmented Generation (GraphRAG) has become the important paradigm for enhancing Large Language Models (LLMs) with external knowledge. However, existing approaches are constrained by their reliance on high-quality knowledge graphs: manually built ones are not scalable, while automatically extracted ones are limited by the performance of LLM extractors, especially when using smaller, local-deployed models. To address this, we introduce Think-on-Graph 3.0 (ToG-3), a novel framework featuring a Multi-Agent Context Evolution and Retrieval (MACER) mechanism. Its core contribution is the dynamic construction and iterative refinement of a Chunk-Triplets-Community heterogeneous graph index, powered by a Dual-Evolution process that adaptively evolves both the query and the retrieved sub-graph during reasoning. ToG-3 dynamically builds a targeted graph index tailored to the query, enabling precise evidence retrieval and reasoning even with lightweight LLMs. Extensive experiments demonstrate that ToG-3 outperforms compared baselines on both deep and broad reasoning benchmarks, and ablation studies confirm the efficacy of the components of MACER framework.

1 Introduction

The rapid advancement of both commercial (OpenAI, 2025; AI, 2025a; Comanici et al., 2025) and open-source Large Language Models (LLMs) (Yang et al., 2025; AI, 2025b; Liu et al., 2024; Zeng et al., 2025; Gan et al., 2023) has significantly enhanced the accessibility of generative AI capabilities for both end-users and developers. Retrieval-augmented generation (RAG) (Gao et al., 2023) has become a popular method for grounding Large Language Models (LLMs) with external knowledge, addressing

issues like knowledge cutoff and hallucination. ToG (Sun et al., 2023; Ma et al., 2024) pioneered an iterative hybrid RAG framework that tightly couples text and KGs retrieval, though their approach relies on pre-existing structured KGs such as Freebase and Wikidata. On the other hand, methods like GraphRAG (Edge et al., 2024) and LightRAG (Guo et al., 2024) address this issue by constructing a graph directly from the input documents. They create an entity-based graph to enhance information retrieval and summarization. However, as shown in Figure 1, the quality of the generated graph is highly dependent on the LLM’s ability to accurately extract entities and relationships, which can be a bottleneck for lightweight models like Qwen2.5-7B~72B (Yang et al., 2024), which is broadly deployed in private and offline environments. Moreover, these methods often separate the handling of local and global questions.

To overcome these limitations, we introduce **Think-on-Graph 3.0** (ToG-3), a new RAG framework that integrates the strengths of both paradigms. Our core contribution lies in the introduction of a novel Chunk-Triplets-Community heterogeneous graph architecture and a novel MACER (Multi-Agent Context Evolution and Retrieval) mechanism, which pioneeringly incorporates a dual-evolution mechanism of **Evolving Query** and **Evolving Sub-Graph** for precise evidence retrieval. Figure 2 illustrates the key distinctions between ToG-3 and classical RAG paradigms such as NaiveRAG and GraphRAG. ToG-3 introduces a novel dual-evolution mechanism—comprising Evolving Query and Evolving Subgraph—that dynamically refines both the query representation and the graph structure in an iterative manner. This approach addresses a critical limitation of prior RAG methods, which typically construct a static graph index in a single pass without adapting to the actual query.

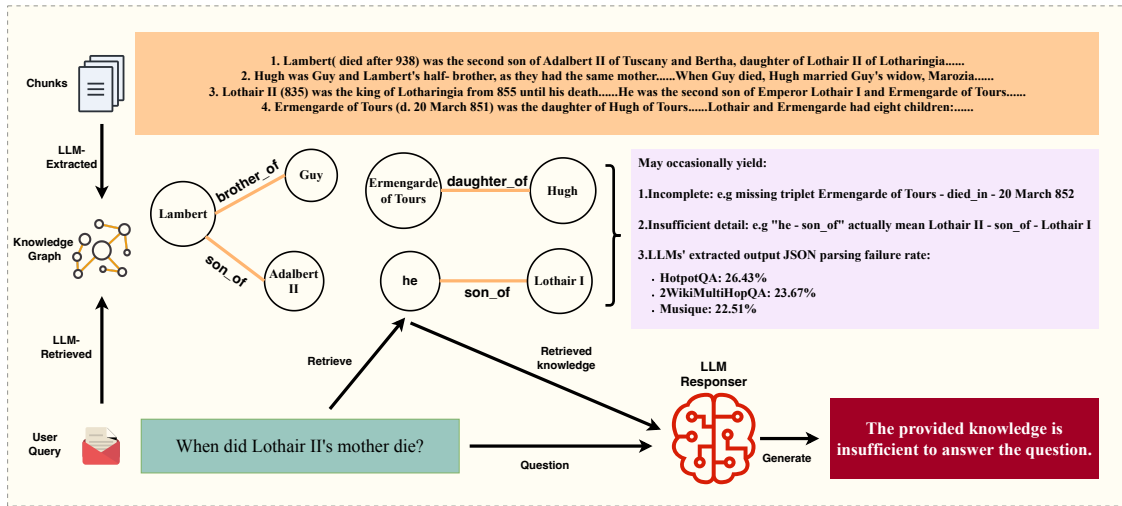


Figure 1: Performance Limitations of Graph-Based RAG systems under Resource-Constrained and Locally-Deployed Scenarios. In such scenarios, developers typically adopt open-source models such as Llama or Qwen as the backbone LLMs. Limitations like incomplete extracted triplets, insufficient extraction details and parsing failure may lead to insufficient knowledge provision, ultimately resulting in failure to adequately answer the query.

The framework is particularly suited for resource-constrained and on-premises deployment scenarios, where lightweight open-source LLMs (e.g., Llama or Qwen) are often employed as the backbone of the RAG system.

Our key contributions are summarized as follows:

1. We propose **MACER** (Multi-Agent Context Evolution and Retrieval), a novel multi-agent framework that introduces a dual-evolution mechanism integrating Evolving Query and Evolving Sub-Graph within graph-based RAG. This design significantly enhances retrieval performance and complex reasoning capabilities, especially when using lightweight open-source LLMs as the backbone of the RAG system.
2. We present **ToG-3**, a unified reasoning system that effectively combines the complementary advantages of prior graph-based and ToG methods through a Chunk-Triplet-Community Heterogeneous Graph Index and a Dual-Evolving Context Retrieval Loop Process.
3. We conduct extensive experiments on both **Deep and Broad Reasoning Tasks**, demonstrating that our approach consistently supports multi-hop inference and large-scale contextual integration, achieving competitive results across diverse benchmarks.

2 Related Work

2.1 Graph-Based Retrieval-Augmented Generation

Recent advances in retrieval-augmented generation (RAG) have increasingly emphasized structural awareness to improve reasoning depth and contextual coherence. Edge et al. (2024) propose GraphRAG, which builds a knowledge graph (KG) from documents via LLM-based entity and relation extraction, then applies community detection to generate hierarchical summaries for global sensemaking. Guo et al. (2024) introduce LightRAG, which employs a dual-level retrieval system combining low-level fact retrieval and high-level semantic discovery using a compact KG, improving both efficiency and coverage. Further building on this idea, Gutiérrez et al. (2024, 2025) present a non-parametric continual learning framework that uses Personalized PageRank over an open KG to enable associative, multi-hop reasoning. Other structure-augmented RAG methods include RAPTOR (Sarathi et al., 2024), Chen et al. (2023) enhance sense-making but often introduce noise through uncontrolled summarization or lack explicit support for multi-hop reasoning. Note that RL-based frameworks such as GraphRAG-R1 (Yu et al., 2025) and Graph-R1 (Luo et al., 2025a) utilize existing Graph-based RAG methods as their retrieval components and train an end-to-end agentic framework. Our work, in contrast, proposes a complementary approach to improve

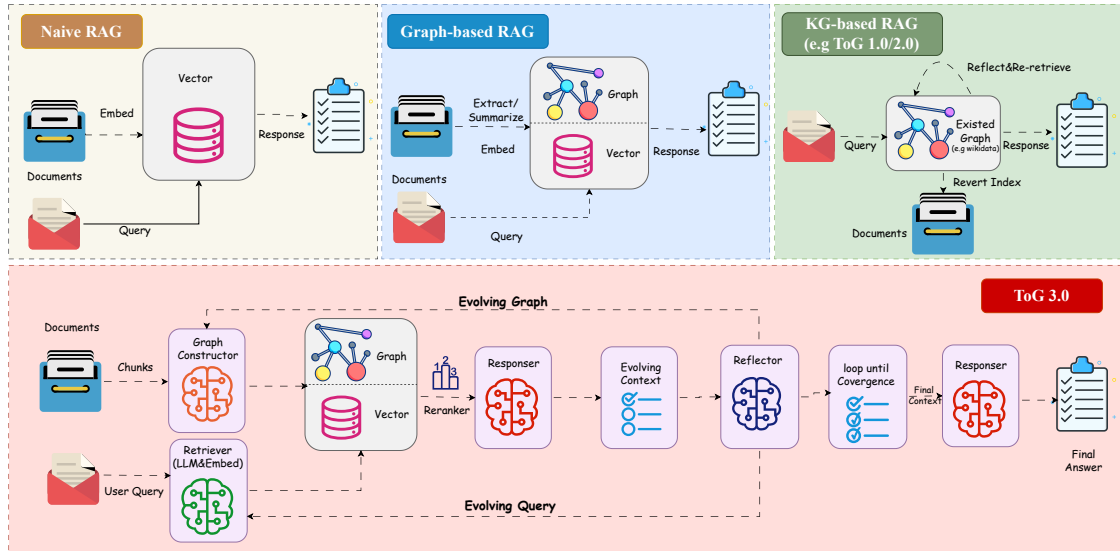


Figure 2: Evolution of Retrieval-Augmented Generation Paradigms. (a) Naive RAG embeds raw documents and performs single-shot retrieval. (b) Graph-based RAG pre-builds a static graph once and retrieves from it. (c) ToG-3 introduces a *four-agent* loop—Retriever, Constructor, Reflector, Reranker, Responder—where the graph and the query sub-tasks *co-evolve* at runtime, yielding dynamic, query-adaptive context that converges to a minimal, sufficient subgraph.

the underlying retrieval paradigm itself. Consequently, ToG-3 could also serve as a plug-in component to enhance such RL frameworks.

2.2 Knowledge Graphs in RAG and Hybrid Approaches

The integration of structured knowledge into LLM reasoning has long been pursued to improve faithfulness and interpretability. Early KG-augmented RAG systems retrieve triples from static external knowledge bases such as Wikidata or Freebase to ground model outputs (Sun et al., 2023). However, these sources are often incomplete, outdated, or misaligned with domain-specific content. To overcome this, hybrid RAG frameworks (Ma et al., 2024) combine unstructured text and structured KGs to balance breadth and precision. Chain-of-Knowledge (CoK) (Li et al., 2024) retrieves from multiple structured sources including Wikipedia, Wikidata, and Wikitable to ground LLM responses. HybridRAG (Sarmah et al., 2024) fuses vector-based and KG-based retrievers, demonstrating superior reasoning performance compared to either modality alone.

2.3 Iterative and Reflective Reasoning in LLMs

Enabling LLMs to reason iteratively has been shown to improve accuracy and faithfulness. ITER-RETGEN (Shao et al., 2023) introduces an iterative loop that alternates between retrieval and

generation, using generated hypotheses to guide further search. Trivedi et al. (2023) combine Chain-of-Thought (CoT) with retrieval, interleaving reasoning steps with evidence gathering, significantly improving performance on multi-hop QA. Self-RAG (Asai et al., 2023) equips LLMs with reflection tokens to decide when to retrieve and whether the output is hallucinated. ReAct (Yao et al., 2023a) combines reasoning traces with external actions, enabling task decomposition and environment interaction. Other efforts focus on continual learning for LLMs, where RAG serves as a non-parametric alternative to fine-tuning (Shi et al., 2024). Continual pretraining (Jin et al., 2022) and instruction tuning (Zhang et al., 2023) can update model parameters but suffer from catastrophic forgetting (Huang et al., 2024). Model editing methods (Yao et al., 2023b) offer fine-grained updates but struggle with generalization.

3 Methodology

Think-on-Graph 3.0 (ToG-3) introduces a novel *Multi-Agent Context Evolution and Retrieval (MACER)* framework for open-domain question answering.

3.1 Problem Formulation

Let $\mathcal{D} = \{d_i\}_{i=1}^N$ be a text corpus. The objective is to answer a user query q with an answer a^* that is both accurate and *faithful* to the source corpus,

derived from a *minimal, sufficient subgraph* \mathcal{G}_q^* of a heterogeneous graph \mathcal{G} constructed from \mathcal{D} :

$$\mathcal{G}_q^* = \underset{\mathcal{G}' \subseteq \mathcal{G}}{\operatorname{argmin}} |\mathcal{G}'| \quad \text{subject to} \quad \operatorname{Suff}(q, \mathcal{G}') = 1, \quad (1)$$

where $\operatorname{Suff}(\cdot, \cdot) \in \{0, 1\}$ is a function judging the sufficiency of a subgraph for answering the query.

Existing methods face a critical dilemma: **(1)** Systems like ToG-1 or 2 rely on high-quality, pre-constructed KGs, limiting their applicability to private or specialized domains. **(2)** Corpus-based GraphRAG methods (e.g., GraphRAG, LightRAG) build a static graph from \mathcal{D} in one go. Their performance is bottlenecked by the quality of this initial graph, which in turn depends heavily on the capability of the LLM used for information extraction.

3.2 Heterogeneous Graph Index: Schema and Construction

3.2.1 Node and Edge Schema

The Constructor Agent builds a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with three node types:

- **Chunks** (\mathcal{C}): Sentence-level text passages from the corpus.
- **Triplets** (\mathcal{T}): Semantic triples (s, p, o) extracted from chunks, annotated with entity and relation types ($\operatorname{type}_s, \operatorname{type}_p, \operatorname{type}_o$).
- **Communities** (\mathcal{M}): Summaries of entity clusters obtained via Leiden clustering on the entity co-occurrence graph, each condensed into an abstract.

Edges are defined by three type relations:

- **OPENREL** (s, p, o) : Connects entities s and o via predicate p extracted by the LLM, forming an open-domain semantic triple.
- **MENTIONEDIN** (t, c) : Connects a triplet t to the chunk c from which it was extracted.
- **SUMMARYFOR** (m, e) : Connects a community summary node m to an entity e that belongs to that community.

This unified schema allows both fine-grained (chunk/triplet) and high-level (community) information to be retrieved seamlessly within a single vector space, effectively addressing the local/global retrieval dichotomy of prior GraphRAG systems.

3.2.2 Offline Index Construction

Algorithm 1 in Appendix. B details the one-time construction of the universal index \mathcal{G} . A key design choice is the use of a single frozen encoder E_θ (e.g., jina-mebedding-v3 (Sturua et al., 2024)) to embed all nodes—regardless of type—into a unified 1024-dimensional dense vector space. This enables efficient vector search across all node types during retrieval.

3.3 The MACER Process: Multi-Agent Context Evolution and Retrieval

The core of ToG-3 is the online MACER loop (Algorithm 2), an iterative process of retrieval, generation, and reflection that dynamically evolves the context subgraph \mathcal{G}_k . We formalize this process as an episodic Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r)$.

State Space (\mathcal{S}) : At each step k , the state $s_k = (q, \mathcal{G}_k, \mathcal{H}_k)$ captures the complete reasoning context, including the original query q , the current evidence subgraph \mathcal{G}_k retrieved by Retriever Agent π_{ret} and reranked by Reranker Agent π_{rer} , and the trajectory history $\mathcal{H}_k = (q'_i, a_i, r_i, \mathcal{G}_i)_{i=0}^{k-1}$ of all previous sub-queries, answers, rewards, and sub-graphs.

Action Space (\mathcal{A}) : The Reflector Agent π_{ref} serves as the policy network. Its action a_k at state s_k is either to generate a targeted refinement sub-query q'_k (to continue the reasoning process) or to output the STOP action (to terminate the episode).

Reward Function (r) : Upon the Response Agent generating an answer a_k , the Reflector immediately provides a sparse, binary reward r_k :

$$r_k = \begin{cases} 1 & \text{if } \operatorname{Suff}(q, \mathcal{G}_k, a_k) = 1 \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

This reward signal is produced by the Reflector Agent to determine if the current context evidence is sufficient to answer the user’s query.

Transition Dynamics (P) Given the current state s_k and an action a_k (which corresponds to issuing a sub-query q'_k), the transition to the next state s_{k+1} occurs deterministically according to the following update rules: The constructor agent π_{const} applies the transition operator using the generated sub-query q'_k and the current graph state \mathcal{G}_k to produce an updated graph \mathcal{G}_{k+1} . This step including iterative sequence of *evolving queries* and

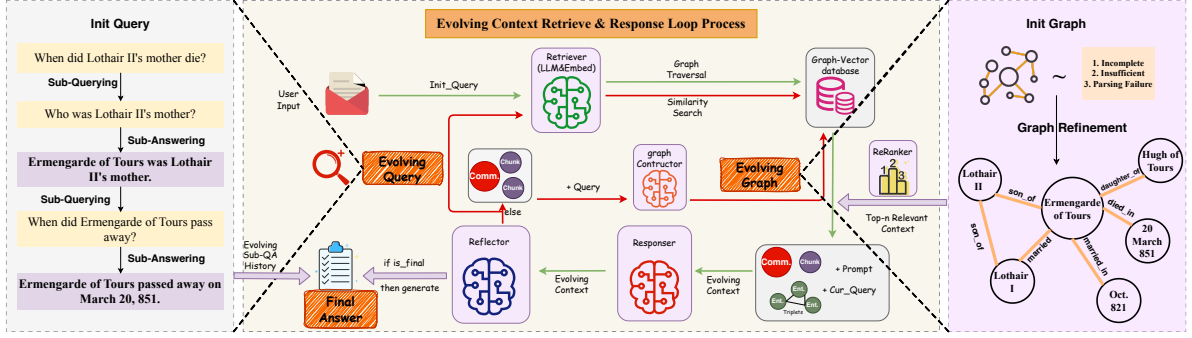


Figure 3: Multi-Agent Dual-Evolving Context Retrieval-Response Loop. The Retriever fetches an initial chunk–triplet–community subgraph and the Reranker reranks and selects the top-n most relevant pieces of evidence.. The Response Agent produces an answer; the Reflector Agent judges sufficiency (reward=1/0). If insufficient (reward=0), the Reflector evolves the query into sub-queries while the Constructor evolves the subgraph (sub-graph refinement). The loop repeats until the context becomes sufficient or the horizon is reached, after which the Response Agent synthesizes the final answer from the full trajectory.

293 *evolving sub-graphs* reflects the structural evolu- 294 tion of the graph based on the agent’s reason- 295 ing action, formally defined by the recurrence:

296
$$q'_k = \pi_{\text{ref}}^{\text{evolve}}(q, \mathcal{G}_k), \quad (3)$$

297
$$\mathcal{G}_{k+1} = \pi_{\text{const}}^{\text{evolve}}(q'_k, \mathcal{G}_k), \quad (4)$$

298 The action history \mathcal{H}_{k+1} is augmented with a 299 new tuple recording the executed sub-query q'_k , the 300 corresponding action a_k , the reward r_k received, 301 and the resulting graph state \mathcal{G}_{k+1} . This ensures 302 a comprehensive trace of the reasoning trajectory, 303 which is essential for credit assignment and subse- 304 quent learning.

305
$$\mathcal{H}_{k+1} = \mathcal{H}_k \cup (q'_k, a_k, r_k, \mathcal{G}_{k+1}) \quad (5)$$

306
$$a^* \leftarrow \pi_{\text{resp}}^{\text{final}}(q, \mathcal{H}_k) \quad (6)$$

308 The complete MACER process, now cast as an 309 MDP, is summarized in Algorithm 2. The loop 310 continues until the Reflector’s policy π_{ref} outputs 311 the STOP action (via $r_k = 1$) or a maximum hori- 312 zon K is reached. The final answer a^* is syn- 313 thesized from the full trajectory \mathcal{H}_k of states and 314 actions, ensuring faithfulness to the evolved evi- 315 dence. This MDP formulation provides the formal 316 foundation for establishing the convergence 317 of the MACER process under mild assumptions, 318 as detailed in Appendix. K. This iterative refine- 319 ment allows ToG-3 to start from a potentially weak 320 initial graph but *specialize* it towards the reason- 321 ing path of the specific query, converging on a 322 high-quality evidence subgraph \mathcal{G}_q^* . This evolu- 323 ing and refinement mechanism alleviate the three 324 fundamental weaknesses of small LMs in static

GraphRAG, including incomplete triplet recall, in- 325 sufficient knowledge details and high parsing fail- 326 ure of LLMs’ output, as mentioned in Section 1. 327

4 Experiment 328

4.1 Experimental Setup 329

330 **Datasets** To comprehensively evaluate the reason- 331 ing capabilities of RAG systems, we con- 332 duct experiments on two distinct categories 333 of tasks: **Deep Reasoning Tasks** including 334 HotpotQA (Yang et al., 2018), 2WikiMulti- 335 HopQA (Ho et al., 2020) and Musique (Trivedi 336 et al., 2022) and **Broad Reasoning Tasks** includ- 337 ing 4 subsets of UltraDomain (Qian et al., 2025) 338 benchmark. Detailed statistics for all datasets are 339 provided in Table 3 and Appendix. C.

340 **Evaluation Metrics** For **Deep Reasoning** 341 **Tasks**, we follow standard QA evaluation prac- 342 tices with **Exact Match (EM)**(Following ToG 343 and ToG-2 (Sun et al., 2023; Ma et al., 2024), we 344 employ a substring-based Exact Match metric.) 345 and **F1 Score**. For **Broad Reasoning Tasks**, we 346 adopt a multi-dimensional LLM-based evaluation 347 approach including **Comprehensiveness**, **Diver-** 348 **sity** and **Empowerment** following (Guo et al., 349 2024). Metrics detail are provide Appendix.E.

350 **Baselines** We compare ToG-3 against the fol- 351 lowing state-of-the-art RAG methods across all 352 datasets, including NaiveRAG (Gao et al., 2023), 353 ToG-2 (Ma et al., 2024), GraphRAG (Edge 354 et al., 2024), LightRAG (Guo et al., 2024), 355 MiniRAG (Fan et al., 2025) and HippoRAG- 356 2 (Gutiérrez et al., 2025). Baselines details can 357 be found in Appendix.D. For graph-based meth-

ods, we maintain identical chunk sizes (1024 tokens) and use the same LLM (Qwen2.5-32B-Instruct (Yang et al., 2024)) for all extraction and generation tasks to eliminate model capability variations. Implementation details are provide Appendix.A.

4.2 Result of Deep Reasoning Benchmark

Result Analysis from a Method Perspective.

Results shown in Table 1 represent the average of three independent reasoning experiments. Previous Graph-based methods like GraphRAG that rely on LLM-based graph construction show limited performance. Their performance is the lowest, particularly in terms of F1 scores as shown in Figure 4b, which can be attributed to a lack of focus on deep factual reasoning and a tendency to produce verbose responses, resulting in low token-level recall. More detailed precision and recall results are provided in Appendix. F.1. ToG-2, without leveraging well-curated knowledge graphs like Freebase and Wikidata, demonstrates moderate performance in open-domain settings. NaiveRAG achieves competitive third-place results by avoiding graph construction limitations and relying solely on retrieved documents for response generation. HippoRAG-2 emerges as the strongest baseline, employing an efficient embedding model with Personalized PageRank algorithm and LLM-based triple filtering to achieve second-best performance. However, our proposed method consistently outperforms all competitors, achieving the highest average EM (0.474) and F1 (0.345) scores across all three benchmarks. This superior performance is attributed to our novel Chunk-Triplets-Community heterogeneous graph architecture and the Multi-Agent Context Evolution and Retrieval (MACER) framework, which enables adaptive subgraph refinement and evolving query decomposition for complex reasoning tasks and overcomes the graph construction challenges that plague other graph-based RAG systems. Additional Baselines are provided in Appendix.H.

Result Analysis from a Dataset Perspective.

As shown in Figure 4, the average performance of the baselines and our method across the HotpotQA, 2WikiMultiHopQA, and Musique datasets generally follows a descending trend. This pattern can be attributed to the following reasons: HotpotQA (Yang et al., 2018): Although widely

used, this dataset has been shown to provide a weaker test of multi-hop reasoning due to the presence of numerous spurious cues and shortcut signals (Trivedi et al., 2022; Gutiérrez et al., 2024). Musique (Trivedi et al., 2022): A challenging multi-hop QA dataset comprising approximately requiring 2–4 hops, which emphasizes a comprehensive evaluation of multi-step reasoning abilities. Musique is designed to feature diverse and complex reasoning paths, necessitating the integration of information across multiple hops to arrive at correct answers.

4.3 Result of Broad Reasoning Tasks

As shown in Figure 5, The four heatmaps clearly demonstrate that the five methods can be distinctly divided into two clusters: the upper-right region (predominantly red, indicating superior performance) and the lower-left region (predominantly blue, indicating inferior performance). Specifically, ToG-3, GraphRAG, and LightRAG exhibit significantly higher win rates compared to NaiveRAG and HippoRAG-2. Detailed win rates (%) of baselines v.s. ToG-3 across four datasets are provided in Table 5 of Appendix. F. Our framework outperforms NaiveRAG by substantial margins (up 75.0% average win rate on all four datasets), highlighting the limitations of chunk-based retrieval for complex queries. While GraphRAG shows competitive performance in comprehensiveness due to its extensive community summarization and retrieval, ToG-3 achieves better balance across all metrics, particularly excelling in diversity and empowerment through its heterogeneous graph architecture that integrates chunk-level, triplet-level, and community-level information. Detailed ELO rating calculation for broad reasoning tasks can be found in Appendix. F.3. The multi-agent dual-evolving context retrieval mechanism enables both deep knowledge reasoning through entity-relation exploration and broad community reasoning. Our analysis reveals that, on average, 20% of the samples require one evolving-context iteration, 32% require two iterations, and 48% require three iterations.

Detailed comparison of time and token Consumption across different methods are provided in Appendix.G. Case studies of ToG-3 retrieval and response output are provided in Appendix. I.

Table 1: Exact Match (EM) and F1 scores on Deep Reasoning datasets. We highlight the **best**, **second-best**, and **third-best** methods with different background color shades.

Method	HotpotQA		2WikiMultihopQA		Musique		Average	
	EM	F1	EM	F1	EM	F1	EM	F1
NaiveRAG	0.634	0.365	0.382	0.189	0.230	0.143	0.415	0.232
ToG-2	0.308	0.153	0.401	0.194	0.103	0.105	0.271	0.151
GraphRAG	0.337	0.011	0.439	0.018	0.109	0.008	0.295	0.012
LightRAG	0.308	0.013	0.420	0.023	0.082	0.009	0.270	0.015
MiniRAG	0.213	0.012	0.125	0.018	0.067	0.007	0.135	0.012
HippoRAG-2	0.612	0.534	0.491	0.254	0.212	0.145	0.438	0.311
Ours	0.654	0.569	0.527	0.291	0.241	0.174	0.474 \uparrow 8.2%	0.345 \uparrow 10.9%

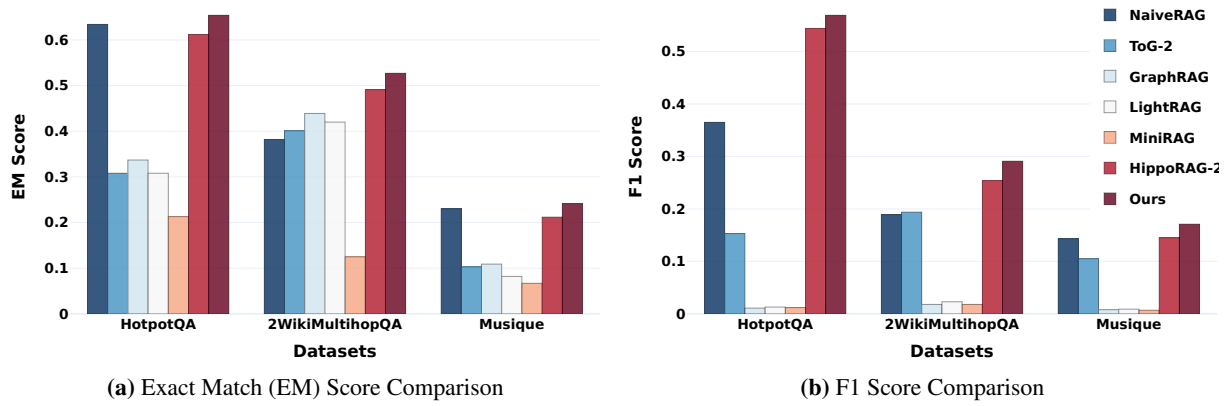


Figure 4: Performance comparison of different RAG methods on multi-hop QA datasets. (a) Exact Match scores measure the percentage of questions where the model’s answer exactly matches the ground truth. (b) F1 scores provide a harmonic mean of precision and recall for token-level answer matching.

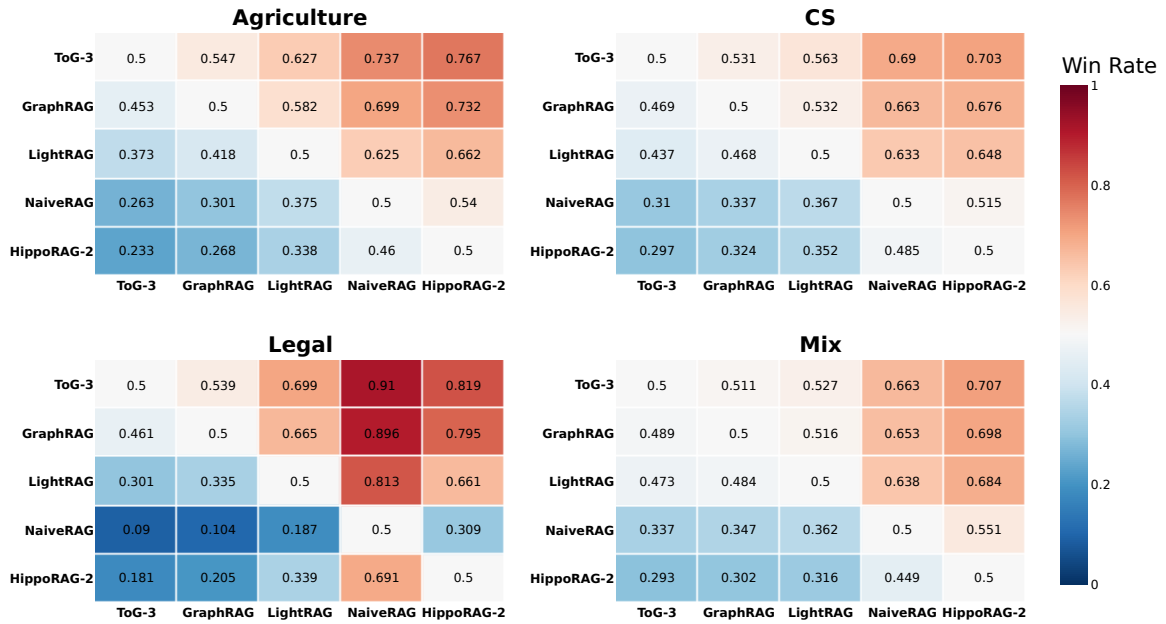


Figure 5: ELO-based Pairwise Win Rate Matrices Across Four Benchmark Datasets. Each heatmap visualizes win probabilities derived from direct head-to-head experimental comparisons, transformed through the ELO framework to ensure transitive consistency. The diagonal of the heatmap is set to a default value of 0.5, indicating self-comparison of the method.

Table 2: Ablation studies of MACER components and foundation model scaling. Standard ToG-3 settings incorporates all MACER components, employs the Qwen2.5-32B-instruct as the backbone LLM, and utilizes the Jina-v3-embedding model for representation encoding and Jina-reranker-v2 for reranking the retrieved evidence.

Ablation Setting	HotpotQA		2WikiMultihopQA		Musique		Average	
	EM	F1	EM	F1	EM	F1	EM	F1
MACER Components Ablation								
w/o Evolving Query	0.614	0.495	0.440	0.227	0.198	0.141	0.417	0.288
w/o Evolving Sub-Graph	0.629	0.525	0.486	0.258	0.223	0.158	0.446	0.314
w/o Community Node	0.656	0.572	0.514	0.283	0.236	0.169	0.469	0.341
Foundation Model Scaling Abalation								
LLM Model								
Qwen2.5-14B	0.587	0.521	0.480	0.255	0.218	0.154	0.428	0.310
Qwen2.5-72B	0.683	0.592	0.550	0.305	0.255	0.182	0.496	0.360
Embedding Model								
Qwen3-Embed-0.6B	0.653	0.571	0.532	0.294	0.244	0.176	0.476	0.347
Qwen3-Embed-4B	0.658	0.577	0.535	0.296	0.247	0.179	0.480	0.351

4.4 Abalation Study

Abalation Study of MACER component Our ablation study reveals the relative importance of each MACER component for deep reasoning performance. The most significant performance degradation occurs when removing the evolving query mechanism (average performance drop of 12.0% in EM and 16.5% in F1), underscoring its critical role in complex question answering, especially when using smaller LLMs. Removing sub-graph refinement causes a moderate performance decrease (average drop of 6.0% in EM and 9.0% in F1), indicating its importance in adapting the knowledge structure to the specific reasoning context. Interestingly, community nodes show the smallest impact on deep reasoning tasks (a slight drop in the average EM and F1 scores), suggesting that while they contribute to performance, the chunk and triplet representations carry most of the relevant information for precise answer generation. However, in broad reasoning tasks, community nodes are essential for comprehensive coverage and diversity, highlighting the complementary roles of different node types in our heterogeneous graph architecture. Note that the reranker agent also delivers a 4.6% improvement in EM and a 10.6% improvement in F1. This is because, during multi-turn RAG processes, an excessive amount of retrieved evidence can otherwise impair the response quality of the responder agent.

Abalation Study of used foundation model

The foundation model scaling analysis reveals several important patterns. First, LLM capacity has a substantially greater impact on performance than embedding model size. Scaling from

Qwen2.5-14B to Qwen2.5-72B yields a 15.9% average improvement in EM scores, highlighting the critical role of reasoning capability in complex QA tasks. Second, larger embedding models provide consistent but more modest improvements. Qwen3-Embed-0.6B shows a slight average EM improvement over jina-embeddings-v3, while Qwen3-Embed-4B provides a 1.7% improvement. This suggests that while retrieval quality matters and larger embedding models contribute to better performance, the LLM’s reasoning capacity remains the primary bottleneck for complex reasoning tasks. These findings provide practical guidance for resource allocation in real-world deployments.

5 Conclusion

In this work, we introduced Think-on-Graph 3.0, a novel framework that fundamentally rethinks the paradigm of RAG for complex reasoning. By proposing the Multi-Agent Context Evolution Retrieval (MACER) mechanism and a dynamic Chunk-Triplets-Community heterogeneous graph architecture, we address critical limitations in both existing graph-based RAG methods and knowledge-graph-dependent approaches. Our comprehensive experimental evaluation demonstrates that ToG-3 achieves state-of-the-art performance across multiple challenging benchmarks. This adaptive capability proves particularly valuable for overcoming the quality constraints of static graph construction and the domain limitations of pre-existing knowledge bases. The framework’s ability to work with light LLMs also opens possibilities for more efficient and deployable AI systems.

526 **Limitations**

527 Of course our work has several limitations. First,
528 constrained by GPU resources, our experiments
529 are primarily conducted with LLMs up to 72B pa-
530 rameters and embedding models up to 4B parame-
531 ters—though these sizes are practical for most de-
532 velopers and small-to-medium enterprises for lo-
533 cal deployment. Second, the evolving query and
534 sub-graph refinement components increase infer-
535 ence latency, typically 2–3× slower than base-
536 line methods, making our approach more suitable
537 for accuracy-critical applications where sacrific-
538 ing speed for improved knowledge fidelity is ac-
539 ceptable. Third, the same mechanisms result in
540 longer context inputs, which demand larger GPU
541 memory capacity for efficient processing. These
542 limitations could be mitigated through model dis-
543 tillation, optimized graph traversal algorithms, and
544 dynamic context pruning techniques in future im-
545 provement.

546 **Ethical Considerations**

547 This research focuses on improving the techni-
548 cal performance of knowledge-enhanced language
549 models. This work utilizes only public benchmark
550 datasets and adheres to strict reproducibility stan-
551 dards. While our framework improves text genera-
552 tion capability, we acknowledge potential risks of
553 generating misleading content and note that per-
554 formance may reflect biases inherent in base mod-
555 els. We follow the ACL ethical guidelines when
556 conducting the research in this paper.

557 **Information About Use Of AI Assistants**

558 In the preparation of this work, the author used AI-
559 assisted technology (specifically, large language
560 models such as GPT-5 and Deepseek-V3) ex-
561 clusively for text refinement purposes. The AI
562 was employed to assist in proofreading, correct-
563 ing grammatical errors, and polishing linguistic
564 expressions to improve the clarity and readability
565 of the manuscript.

566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619

References

Anthropic AI. 2025a. [Introducing claude 4](#).

Meta AI. 2025b. [The llama 4 herd: The beginning of a new era of natively multimodal ai innovation](#).

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. [Self-rag: Learning to retrieve, generate, and critique through self-reflection](#).

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. 2023. [Walking down the memory maze: Beyond context limit through interactive reading](#).

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *arXiv preprint arXiv:2507.06261*.

Benoit Dherin, Michael Munn, Hanna Mazzawi, Michael Wunder, and Javier Gonzalvo. 2025. [Learning without training: The implicit dynamics of in-context learning](#). *arXiv preprint arXiv:2507.16003*.

Junnan Dong, Siyu An, Yifei Yu, Qian-Wen Zhang, Linhao Luo, Xiao Huang, Yunsheng Wu, Di Yin, and Xing Sun. 2025. [Youtu-graphrag: Vertically unified agents for graph retrieval-augmented complex reasoning](#). *arXiv preprint arXiv:2508.19855*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. [From local to global: A graph rag approach to query-focused summarization](#). *arXiv preprint arXiv:2404.16130*.

Tianyu Fan, Jingyuan Wang, Xubin Ren, and Chao Huang. 2025. [Minirag: Towards extremely simple retrieval-augmented generation](#). *arXiv preprint arXiv:2501.06713*.

Ruyi Gan, Ziwei Wu, Renliang Sun, Junyu Lu, Xiaojun Wu, Dixiang Zhang, Kunhao Pan, Ping Yang, Qi Yang, Jiaying Zhang, et al. 2023. [Ziya2: Data-centric learning is all llms need](#). *arXiv preprint arXiv:2311.03301*.

Junqi Gao, Xiang Zou, Ying Ai, Dong Li, Yichen Niu, Bqing Qi, and Jianxing Liu. 2025. [Graph counselor: Adaptive graph exploration via multi-agent synergy to enhance llm reasoning](#). *arXiv preprint arXiv:2506.03939*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. [Retrieval-augmented generation for large language models: A survey](#). *arXiv preprint arXiv:2312.10997*, 2(1).

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. [Lightrag: Simple and fast retrieval-augmented generation](#). 620
621
622

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*. 623
624
625
626
627

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. [From rag to memory: Non-parametric continual learning for large language models](#). 628
629
630
631

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, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics. 632
633
634
635
636
637
638
639

Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024. [Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1416–1428, Bangkok, Thailand. Association for Computational Linguistics. 640
641
642
643
644
645
646
647
648

Yiqian Huang, Shiqi Zhang, and Xiaokui Xiao. 2025. [Ket-rag: A cost-efficient multi-granular indexing framework for graph-rag](#). *arXiv preprint arXiv:2502.09304*. 649
650
651
652

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2022. [Lifelong pretraining: Continuously adapting language models to emerging corpora](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4764–4780, Seattle, United States. Association for Computational Linguistics. 653
654
655
656
657
658
659
660
661

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with pagedattention](#). In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*. 662
663
664
665
666
667
668

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2024. [Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources](#). In *International Conference on Learning Representations*. 669
670
671
672
673
674

675	Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang,	Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo	732
676	Bochao Wu, Chengda Lu, Chenggang Zhao,	Wang, Chen Lin, Yeyun Gong, Lionel M Ni, Heung-	733
677	Chengqi Deng, Chenyu Zhang, Chong Ruan, et al.	Yeung Shum, and Jian Guo. 2023. Think-on-	734
678	2024. Deepseek-v3 technical report. <i>arXiv preprint</i>	graph: Deep and responsible reasoning of large lan-	735
679	<i>arXiv:2412.19437</i> .	guage model on knowledge graph. <i>arXiv preprint</i>	736
680	Haoran Luo, Guanting Chen, Qika Lin, Yikai Guo,	<i>arXiv:2307.07697</i> .	737
681	Fangzhi Xu, Zemin Kuang, Meina Song, Xiaobao	Vincent A Traag, Ludo Waltman, and Nees Jan	738
682	Wu, Yifan Zhu, Luu Anh Tuan, et al. 2025a.	Van Eck. 2019. From louvain to leiden: guarantee-	739
683	Graph-r1: Towards agentic graphrag framework via	ing well-connected communities. <i>Scientific reports</i> ,	740
684	end-to-end reinforcement learning. <i>arXiv preprint</i>	9(1):1–12.	741
685	<i>arXiv:2507.21892</i> .	Harsh Trivedi, Niranjana Balasubramanian, Tushar	742
686	Haoran Luo, Guanting Chen, Yandan Zheng, Xi-	Khot, and Ashish Sabharwal. 2022. MuSiQue: Mul-	743
687	aobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin	ti-hop questions via single-hop question composition.	744
688	Kuang, Meina Song, Yifan Zhu, et al. 2025b.	<i>Transactions of the Association for Computational</i>	745
689	Hypergraphrag: Retrieval-augmented generation	<i>Linguistics</i> , 10:539–554.	746
690	via hypergraph-structured knowledge representa-	Harsh Trivedi, Niranjana Balasubramanian, Tushar	747
691	tion. <i>arXiv preprint arXiv:2503.21322</i> .	Khot, and Ashish Sabharwal. 2023. Interleav-	748
692	Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li,	ing retrieval with chain-of-thought reasoning for	749
693	Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo.	knowledge-intensive multi-step questions.	750
694	2024. Think-on-graph 2.0: Deep and faithful large	Feng Wang, Yuqing Li, and Han Xiao. 2025. jina-	751
695	language model reasoning with knowledge-guided	reranker-v3: Last but not late interaction for	752
696	retrieval augmented generation.	listwise document reranking. <i>arXiv preprint</i>	753
697	OpenAI. 2025. Introducing gpt-5.	<i>arXiv:2509.25085</i> .	754
698	Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong	Derong Xu, Pengyue Jia, Xiaopeng Li, Yingyi	755
699	Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang.	Zhang, Maolin Wang, Qidong Liu, Xiangyu Zhao,	756
700	2025. Memorag: Boosting long context processing	Yichao Wang, Huifeng Guo, Ruiming Tang, et al.	757
701	with global memory-enhanced retrieval augmenta-	2025. Align-grag: Reasoning-guided dual align-	758
702	tion. In <i>Proceedings of the ACM Web Conference</i>	ment for graph retrieval-augmented generation.	759
703	<i>2025 (TheWebConf 2025)</i> , Sydney, Australia. ACM.	<i>arXiv preprint arXiv:2505.16237</i> .	760
704	ArXiv:2409.05591.	An Yang, Anfeng Li, Baosong Yang, Beichen Zhang,	761
705	Bhaskarjit Sarmah, Benika Hall, Rohan Rao, Sunil Pa-	Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao,	762
706	tel, Stefano Pasquali, and Dhagash Mehta. 2024.	Chengen Huang, Chenxu Lv, et al. 2025. Qwen3	763
707	Hybridrag: Integrating knowledge graphs and vector	technical report. <i>arXiv preprint arXiv:2505.09388</i> .	764
708	retrieval augmented generation for efficient informa-	An Yang, Baosong Yang, Beichen Zhang, Binyuan	765
709	tion extraction. <i>arXiv preprint arXiv:2408.04948</i> .	Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayi-	766
710	Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh	heng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian	767
711	Khanna, Anna Goldie, and Christopher D. Manning.	Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang,	768
712	2024. RAPTOR: recursive abstractive processing	Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang,	769
713	for tree-organized retrieval. In <i>The Twelfth Inter-</i>	Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei	770
714	<i>national Conference on Learning Representations,</i>	Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men,	771
715	<i>ICLR 2024, Vienna, Austria, May 7-11, 2024.</i> Open-	Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia,	772
716	Review.net.	Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang	773
717	Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie	Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu	774
718	Huang, Nan Duan, and Weizhu Chen. 2023. En-	Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5	775
719	hancing retrieval-augmented large language models	technical report. <i>arXiv preprint arXiv:2412.15115</i> .	776
720	with iterative retrieval-generation synergy.	Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Ben-	777
721	Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin,	gio, William Cohen, Ruslan Salakhutdinov, and	778
722	Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna	Christopher D Manning. 2018. Hotpotqa: A dataset	779
723	Ebrahimi, and Hao Wang. 2024. Continual learning	for diverse, explainable multi-hop question answer-	780
724	of large language models: A comprehensive survey.	ing. In <i>Proceedings of the 2018 Conference on Em-</i>	781
725	<i>arXiv preprint arXiv:2404.16789</i> .	<i>pirical Methods in Natural Language Processing</i> ,	782
726	Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram,	pages 2369–2380.	783
727	Michael Günther, Bo Wang, Markus Krimmel, Feng	Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak	784
728	Wang, Georgios Mastrobas, Andreas Koukounas,	Shafran, Karthik Narasimhan, and Yuan Cao. 2023a.	785
729	Andreas Koukounas, Nan Wang, and Han Xiao.	React: Synergizing reasoning and acting in language	786
730	2024. jina-embeddings-v3: Multilingual embed-	models. In <i>International Conference on Learning</i>	787
731	dings with task lora.	<i>Representations (ICLR)</i> .	788

- 789 Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan
790 Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and
791 Ningyu Zhang. 2023b. [Editing large language mod-](#)
792 [els: Problems, methods, and opportunities](#). In *Pro-*
793 *ceedings of the 2023 Conference on Empirical Meth-*
794 *ods in Natural Language Processing*, pages 10222–
795 10240, Singapore. Association for Computational
796 Linguistics.
- 797 Chuanyue Yu, Kuo Zhao, Yuhan Li, Heng Chang,
798 Mingjian Feng, Xiangzhe Jiang, Yufei Sun, Jia Li,
799 Yuzhi Zhang, Jianxin Li, et al. 2025. Graphrag-r1:
800 Graph retrieval-augmented generation with process-
801 constrained reinforcement learning. *arXiv preprint*
802 *arXiv:2507.23581*.
- 803 Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin
804 Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao
805 Zeng, Jiajie Zhang, et al. 2025. Glm-4.5: Agen-
806 tic, reasoning, and coding (arc) foundation models.
807 *arXiv preprint arXiv:2508.06471*.
- 808 Zihan Zhang, Meng Fang, Ling Chen, and
809 Mohammad-Reza Namazi-Rad. 2023. [CITB:](#)
810 [A benchmark for continual instruction tuning](#). In
811 *Findings of the Association for Computational*
812 *Linguistics: EMNLP 2023*, pages 9443–9455, Sin-
813 gapore. Association for Computational Linguistics.
- 814 Yibo Zhao, Jiapeng Zhu, Ye Guo, Kangkang He,
815 and Xiang Li. 2025. E²graphrag: Streamlining
816 graph-based rag for high efficiency and effective-
817 ness. *arXiv preprint arXiv:2505.24226*.

Appendices

Within this supplementary material, we elaborate on the following aspects:

- Appendix A: Implementation Details and Hyperparameters
- Appendix B: Detailed ToG-3 Algorithms
- Appendix C: Datasets Statistics and Details
- Appendix D: Baselines Details
- Appendix E: Evaluation Metrics
- Appendix F: More Experiment Results and Details
- Appendix G: Comparison of Time and Token Consumption
- Appendix H: Additional Baselines
- Appendix I: Case Study for ToG-3
- Appendix J: Graph Visualization Examples
- Appendix K: Theoretical Support for ToG-3
- Appendix L: LLM Prompts

A Implementation Details

We implement ToG-3 experiments with the following configuration: **Data Processing:** Chunk size is set to 1024 tokens with 20-token overlap between consecutive chunks to maintain contextual continuity. **Multi-Agent hyperparameter:** Constructor Agent extracts a maximum of 2 knowledge triplets per chunk and employs hierarchical Leiden clustering (Traag et al., 2019) with maximum cluster size of 5 for community detection. Retriever Agent retrieves top-5 most relevant nodes using hybrid vector-graph similarity matching. Then, the Reranker reranks the top-2 relevant evidence nodes (or triples) within this retrieved subgraph. Reflector/Responder Agent utilizes the top-2 retrieved passages as context for answer generation. **Backend Infrastructure:** LLM service is based on Qwen2.5-32B-Instruct (Yang et al., 2024) deployed with vLLM (Kwon et al., 2023) engine using bfloat16 precision and prefix caching enabled and greedy-search generation method, which is more stable than the Qwen3 model in mixed reasoning mode in our task; embeddings are generated using Jina-embeddings-v3 (1024-dimensional) (Sturua et al., 2024); we

use jina-reranker-v2 (Wang et al., 2025) as the reranker model; Our server is equipped with 8 A100 40GB cards, AMD EPYC 256-core Processor, 2TB memory, and Ubuntu 20.04.1 system. and the hybrid vector-graph storage is implemented using Neo4j community edition¹ for efficient knowledge representation and retrieval, see Appendix.J for visualized graph example.

B ToG-3 Algorithms

Algorithms 1 and 2 present the two-stage pipeline of ToG-3. The first stage constructs a heterogeneous graph index comprising chunks, triplets, and communities, while the second stage implements a Multi-Agent Context Evolution and Retrieval (MACER) loop featuring a novel dual-evolution mechanism—Evolving Query and Evolving Subgraph—that dynamically refines both the query representation and the graph structure through iterative interaction.

C Dataset Detail

This section presents a comprehensive statistical overview of the **Deep and Broad datasets** we use in this paper, including detailed statistics metadata and licensing information, as summarized in Table 3. Additionally, we provide individual descriptions of each dataset to elucidate their respective characteristics and intended use cases.

C.1 Deep Reasoning Datasets

- **HotpotQA** (Yang et al., 2018): A crowd-sourced question answering dataset built on English Wikipedia, comprising approximately 113K questions. Each question is constructed to require the combination of information from the introductory sections of two Wikipedia articles for answering. The dataset provides two gold paragraphs per question, along with a list of sentences identified as supporting facts necessary to answer the question. HotpotQA includes various reasoning strategies such as bridge questions (involving missing entities), intersection questions (e.g., “what satisfies both property A and property B?”), and comparison questions (comparing two entities through a common attribute). It is available in two settings: a *few-shot distractor setting* where models are provided with 10

¹<https://neo4j.com/product/community-edition>

Table 3: Statistics of Deep Reasoning and Broad Reasoning Datasets. Metrics abbreviations: Comp. (Comprehensiveness), Div. (Diversity), Emp. (Empowerment).

Dataset	Corpus Size	Chunks	Entities/Relations	Communities	Metrics	License
Deep Reasoning Tasks						
HotpotQA	9,809	9,812	37,358/30,987	5,041	EM, F1	Apache-2.0
2WikiMultihopQA	6,119	6122	19,311/21,077	3,417	EM, F1	Apache-2.0
Musique	11,254	11,300	32,842/39,134	6,258	EM, F1	CC-BY-4.0
Broad Reasoning Tasks						
CS	10	2,134	3,530/33,507	1,166		
Agriculture	12	2,025	6,043/12,571	1,039	Comp., Div., Emp.	Apache-2.0
Legal	94	5,900	26,180/44,334	1,359		
Mix	61	658	2,784/5,089	425		

paragraphs including the gold ones, and an *open-domain full-wiki setting* where models must retrieve relevant passages from the entire Wikipedia corpus given only the question.

- **2WikiMultihopQA** (Ho et al., 2020): A multi-hop question answering dataset that contains complex questions requiring reasoning over multiple Wikipedia paragraphs. Each question is designed to necessitate logical connections across different pieces of information to arrive at the correct answer.
- **Musique** (Trivedi et al., 2022): A challenging multi-hop QA dataset containing approximately 25K 2–4 hop questions, constructed by composing single-hop questions from five existing single-hop QA datasets. It is designed to feature diverse and complex reasoning paths, requiring models to integrate information from multiple hops to generate correct answers. The dataset emphasizes comprehensive evaluation of multi-step reasoning capabilities.

C.2 Broad Reasoning Datasets

The following datasets are curated from the UltraDomain (Qian et al., 2025) benchmark. The benchmark construction leverages financial reports, legal contracts, and 428 college textbooks across 18 distinct domains to evaluate model versatility and adaptability in specialized and broad application scenarios:

- **CS:** Computer science domain focusing on data science, software engineering, and programming topics, requiring technical comprehension and analytical reasoning.
- **Agriculture:** Covers agricultural practices including beekeeping, crop production,

and disease prevention, demanding domain-specific knowledge integration.

- **Legal:** Derived from legal contracts and documents, focusing on corporate legal practices, regulatory compliance, and governance, requiring precise interpretation of nuanced legal language.
- **Mix:** Contains diverse contexts from college textbooks spanning natural sciences, humanities, and social sciences, testing generalization capabilities across interdisciplinary topics.

D Baselines

This section presents the baseline methods evaluated in this paper, encompassing both classical algorithms such as NaiveRAG and GraphRAG, as well as recently proposed approaches including LightRAG, ToG-2, and HippoRAG-2. Baselines are as follows:

- **NaiveRAG** (Gao et al., 2023): A standard chunk-based retrieval baseline that segments raw texts into chunks and stores them in a vector database using text embeddings. For queries, it generates vectorized representations to directly retrieve text chunks based on semantic similarity.
- **GraphRAG** (Edge et al., 2024): A graph-enhanced RAG system that utilizes an LLM to extract entities and relationships from text, representing them as nodes and edges. It generates community summaries through graph clustering and employs both local (entity-based) and global (community-based) retrieval strategies for comprehensive information access.
- **LightRAG** (Guo et al., 2024): A graph-structured RAG framework that employs a

Algorithm 1 Offline Construction of Heterogeneous Index Graph \mathcal{G}

Require: Corpus $\mathcal{D} = \{d_i\}_{i=1}^N$, lightweight LM $\mathcal{L}_{\text{light}}$, encoder E_θ

Ensure: Heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

- 1: $\mathcal{V} \leftarrow \emptyset, \mathcal{E} \leftarrow \emptyset$
- 2: $\mathcal{C} \leftarrow \text{SplitIntoChunks}(\mathcal{D}) \triangleright$ Sentence-level segmentation
- 3: $\mathcal{V} \leftarrow \mathcal{V} \cup \mathcal{C}$
- 4: **for** each chunk $c \in \mathcal{C}$ **do**
- 5: $\mathcal{T}_c \leftarrow \mathcal{L}_{\text{light}}(c) \triangleright$ Extract semantic triplets $(s, p, o, \text{type}_s, \text{type}_p, \text{type}_o)$
- 6: $\mathcal{V} \leftarrow \mathcal{V} \cup \mathcal{T}_c$
- 7: **for** each triplet $t \in \mathcal{T}_c$ **do**
- 8: $\mathcal{E} \leftarrow \mathcal{E} \cup \{\text{MENTIONEDIN}(t, c)\}$
- 9: **end for**
- 10: **end for**
- 11: $G_e \leftarrow \text{BuildEntityCoOccurrenceGraph}(\mathcal{T})$
 $\triangleright \mathcal{T}$ is all triplets
- 12: $\{M_\ell\}_\ell \leftarrow \text{LeidenClustering}(G_e)$
- 13: **for** each community M_ℓ **do**
- 14: $m_\ell \leftarrow \mathcal{L}_{\text{light}}(M_\ell) \triangleright$ Generate community summary
- 15: $\mathcal{V} \leftarrow \mathcal{V} \cup \{m_\ell\}$
- 16: **for** each entity $e \in M_\ell$ **do**
- 17: $\mathcal{E} \leftarrow \mathcal{E} \cup \{\text{SUMMARYFOR}(m_\ell, e)\}$
- 18: **end for**
- 19: **end for**
- 20: **Encode** every node $v \in \mathcal{V}$ using $E_\theta \triangleright$ Unified dense encoding
- 21: **return** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

982 dual-level retrieval system combining low-
983 level entity retrieval with high-level knowl-
984 edge discovery. It integrates graph structures
985 with vector representations for efficient re-
986 trieval of related entities and their relation-
987 ships.

- 988 • **ToG-2** (Ma et al., 2024): A knowledge
989 graph-based framework implements a tight-
990 coupling hybrid RAG paradigm that itera-
991 tively retrieves information from both un-
992 structured texts and structured knowledge
993 sources. It alternates between graph retrieval
994 and context retrieval for in-depth knowledge
995 exploration.
- 996 • **HippoRAG-2** (Gutiérrez et al., 2025): A
997 non-parametric continual learning frame-
998 work that leverages Personalized PageRank

Algorithm 2 ToG-3: Multi-Agent Context Evolution and Retrieval (MACER) Loop

Require: Query q , heterogeneous graph \mathcal{G} , LLM \mathcal{L} , max rounds K

Ensure: Final answer a^*

- 1: $k \leftarrow 0, \mathcal{G}_0 \leftarrow \text{Retriever}(q, \mathcal{G}) \triangleright$ Initial retrieval
- 2: $\mathcal{H}_0 \leftarrow \{(q, \mathcal{G}_0, \text{init})\} \triangleright$ Initialize trajectory history
- 3: **repeat**
- 4: $G_k \leftarrow \pi_{\text{rer}}(q, \mathcal{G}_{k-1}) \triangleright$ Reranker Agent rerank and select the sub-graph
- 5: $a_k \leftarrow \pi_{\text{resp}}(q, \mathcal{G}_k, \mathcal{H}_k) \triangleright$ Response Agent generates answer
- 6: $r_k \leftarrow \pi_{\text{ref}}^{\text{suff}}(q, \mathcal{G}_k, a_k) \triangleright$ Reflector judges sufficiency
- 7: **if** $r_k = 1$ **then break**
- 8: **end if**
- 9: $q'_k \leftarrow \pi_{\text{ref}}^{\text{evolve}}(q, \mathcal{G}_k) \triangleright$ Reflector evolves query
- 10: $\mathcal{G}_{k+1} \leftarrow \pi_{\text{const}}^{\text{evolve}}(q'_k, \mathcal{G}_k) \triangleright$ Constructor evolves subgraph
- 11: $\mathcal{H}_{k+1} \leftarrow \mathcal{H}_k \cup \{(q'_k, a_k, r_k, \mathcal{G}_{k+1})\}$
- 12: $k \leftarrow k + 1$
- 13: **until** $k = K$
- 14: $a^* \leftarrow \pi_{\text{resp}}^{\text{final}}(q, \mathcal{H}_k) \triangleright$ Synthesize answer from full trajectory
- 15: **return** a^*

algorithm over an open knowledge graph 999
constructed using LLM-extracted triples. It 1000
enhances multi-hop reasoning capabilities 1001
through sophisticated graph traversal and 1002
passage integration mechanisms. 1003

E Metrics 1004

We employ different evaluation protocols for the 1005
two task categories: 1006

For **Deep Reasoning Tasks**, we follow standard 1007
QA evaluation practices as ToG (Sun et al., 2023; 1008
Ma et al., 2024) and HippoRAG (Gutiérrez et al., 1009
2024, 2025): 1010

- **Exact Match (EM)**: Measures the percent- 1011
age of predictions that exactly match the 1012
ground truth answer. Specifically, we fol- 1013
lows the Substring-based EM metric (used 1014
in ToG/ToG-2 (Sun et al., 2023; Ma et al., 1015
2024)) to robustly assess answer accuracy in 1016
longer, natural-language response generated 1017
by LLMs, which goes through the whole re- 1018

1019 sponse to check whether the answer is in.

- 1020 • **F1 Score:** Computes word-level overlap be- 1061
1021 tween predictions and ground truth answers. 1062

1022 For **Broad Reasoning Tasks**, we adopt a multi- 1063
1023 dimensional LLM-based evaluation approach due 1064
1024 to the complexity and open-ended nature of these 1065
1025 queries following LightRAG (Guo et al., 2024): 1066

- 1026 • **Comprehensiveness (Comp.):** Measures 1067
1027 how thoroughly the answer addresses all as- 1068
1028 pects of the question. 1069
- 1029 • **Diversity (Div.):** Assesses the variety of per- 1070
1030 spectives and insights provided in the answer. 1071
- 1031 • **Empowerment (Emp.):** Evaluates how well 1072
1032 the answer enables informed understanding 1073
1033 and judgment. 1074

1034 The LLM-based evaluation uses GPT-4o-mini as 1075
1035 judge, with careful attention to prompt design and 1076
1036 answer ordering to avoid positional bias. The 1077
1037 LLM evaluation prompt is shown in Appendix.L 1078

1038 F More Experiment Results and Details 1079

1039 This section presents extended experimental re- 1080
1040 sults, including detailed precision and recall met- 1081
1041 rics on Deep Reasoning tasks, as well as one-to- 1082
1042 one win rates from Broad Reasoning tasks. The 1083
1043 pairwise win rates are converted into a unified 1084
1044 ELO rating system, with the resulting ratings vi- 1085
1045 sualized in the heatmap shown in Figure 5.

1046 F.1 Precision and Recall Rate Results 1086

1047 Table 4 reveals the underlying reason for the re- 1087
1048 latively low F1 scores of GraphRAG and Ligh- 1088
1049 tRAG: these methods are not specifically designed 1089
1050 for deep reasoning tasks. By examining both pre- 1090
1051 cision/recall metrics and output cases, we observe 1091
1052 that excessively long or unfocused responses tend 1092
1053 to substantially reduce recall, thereby diminishing 1093
1054 overall F1 performance. 1094

1055 F.2 Result Detail in Braod Reasoning Tasks 1095

1056 Table 5 presents the pairwise win rates (%) 1096
1057 of baseline methods against ToG-3 across four 1097
1058 datasets and four evaluation dimensions. The re- 1098
1059 sults demonstrate that ToG-3 consistently outper- 1099
1060 forms all compared baselines. 1100

F.3 ELO Rating Calculation for Broad Reasoning Tasks 1061

1062 This appendix details the mathematical framework 1063
1064 and computational process for deriving ELO rat- 1065
1066 ings from pairwise comparison data across four 1066
1067 benchmark datasets. The ELO rating system 1067
1068 provides a mathematically consistent approach to 1068
1069 quantify relative performance differences between 1069
1070 retrieval-augmented generation methods. The 1070
1071 ELO rating system transforms raw win rates into 1071
1072 a logarithmic scale that ensures transitive consis- 1072
1073 tency in performance rankings. The core transfor- 1073
1074 mation is defined as follows: 1074

For a given method i with win rate w_i against 1075
1076 the reference method (ToG-3), the ELO rating dif- 1075
1077 ference is calculated as: 1076

$$\Delta R_i = 400 \cdot \log_{10} \left(\frac{1}{w_i} - 1 \right) \quad 1077$$

The absolute ELO rating for method i is then: 1078

$$R_i = R_{\text{ref}} - \Delta R_i \quad 1079$$

where $R_{\text{ref}} = 1600$ is the reference rating for ToG- 1080
1081 3. 1081

The win probability between any two methods i 1082
1083 and j with ratings R_i and R_j is given by: 1083

$$P(i \text{ beats } j) = \frac{1}{1 + 10^{(R_j - R_i)/400}} \quad 1084$$

G Analysis of Computation Cost 1085

G.1 Comparison of Time Consumption 1086

1087 The Table 6 reveal a consistent accuracy- 1087
1088 efficiency trade-off across all datasets. We ob- 1088
1089 served that during the indexing phase, GraphRAG 1089
1090 required the longest processing time, averaging 1090
1091 13.10 hours. This is primarily due to its need 1091
1092 to extract a large number of triplets and generate 1092
1093 community summaries. In comparison, both ToG- 1093
1094 3 and LightRAG showed similar indexing times, 1094
1095 at 10.13 and 10.06 hours respectively. Although 1095
1096 ToG-3 also involves community summary gener- 1096
1097 ation, it constructs the graph more efficiently by 1097
1098 extracting fewer relational structures during graph 1098
1099 initialization compared to both LightRAG and 1099
1100 GraphRAG. While LightRAG achieve faster infer- 1100
1101 ence times, they suffer from lower accuracy due 1101
1102 to redundant graph elements or simpler retrieval 1102
1103 mechanisms. While HippoRAG-2 achieves com- 1103
1104 petitive performance and faster inference speed, 1104
1105 it still falls short of the EM scores attained by 1105

Table 4: Comprehensive Evaluation Metrics of five RAG methods across three deep reasoning datasets. The best results of each dataset are marked in **bold**.

Method	HotpotQA			2WikiMultihopQA			Musique		
	F1	R	P	F1	R	P	F1	R	P
NaiveRAG	0.365	0.593	0.346	0.189	0.345	0.168	0.143	0.280	0.126
GraphRAG	0.011	0.423	0.006	0.018	0.456	0.009	0.008	0.266	0.004
LightRAG	0.013	0.393	0.007	0.023	0.429	0.012	0.009	0.224	0.005
MiniRAG	0.012	0.372	0.006	0.018	0.403	0.009	0.007	0.203	0.003
ToG-3	0.569	0.675	0.492	0.291	0.496	0.208	0.174	0.302	0.122

P: Precision, R: Recall. ToG-3 achieves best F1 while maintaining high precision-recall balance.

Table 5: Win rates (%) of baselines v.s. ToG-3 across four datasets and four evaluation dimensions. The better results of each dataset are marked in **bold**.

Metrics	Agriculture		CS		Legal		Mix	
	NaiveRAG	ToG-3	NaiveRAG	ToG-3	NaiveRAG	ToG-3	NaiveRAG	ToG-3
Comprehensiveness	26.1%	73.9%	30.1%	69.9%	10.1%	89.9%	32.5%	67.5%
Diversity	16.9%	83.1%	29.7%	70.3%	7.3%	92.7%	25.9%	74.1%
Empowerment	27.2%	72.8%	30.5%	69.5%	10.1%	89.9%	36.2%	63.8%
Overall	26.3%	73.7%	31.0%	69.0%	9.0%	91.0%	33.7%	66.3%
	GraphRAG	ToG-3	GraphRAG	ToG-3	GraphRAG	ToG-3	GraphRAG	ToG-3
Comprehensiveness	44.5%	55.5%	47.3%	52.7%	47.3%	52.7%	49.3%	50.7%
Diversity	42.1%	57.9%	46.1%	53.9%	44.5%	55.5%	49.7%	50.3%
Empowerment	22.9%	77.1%	40.9%	59.1%	27.3%	72.7%	36.1%	63.9%
Overall	45.3%	54.7%	46.9%	53.1%	46.1%	53.9%	48.9%	51.1%
	LightRAG	ToG-3	LightRAG	ToG-3	LightRAG	ToG-3	LightRAG	ToG-3
Comprehensiveness	36.6%	63.4%	43.3%	56.7%	31.3%	68.7%	45.3%	54.7%
Diversity	29.7%	70.3%	39.7%	60.3%	25.7%	74.3%	37.0%	63.0%
Empowerment	38.2%	61.8%	43.7%	56.3%	31.3%	68.7%	49.7%	50.3%
Overall	37.3%	62.7%	43.7%	56.3%	30.1%	69.9%	47.3%	52.7%
	HippoRAG-2	ToG-3	HippoRAG-2	ToG-3	HippoRAG-2	ToG-3	HippoRAG-2	ToG-3
Comprehensiveness	22.2%	77.8%	29.3%	70.7%	19.3%	80.7%	27.3%	72.7%
Diversity	16.5%	83.5%	25.7%	74.3%	15.0%	85.0%	21.4%	78.6%
Empowerment	25.5%	74.5%	30.6%	69.4%	19.3%	80.7%	31.7%	68.3%
Overall	23.3%	76.7%	29.7%	70.3%	18.1%	81.9%	29.3%	70.7%

ToG-3. GraphRAG’s expensive two-stage indexing yields suboptimal results despite longer processing times. ToG-3 demonstrates an effective balance: its efficient heterogeneous graph construction produces refined knowledge bases across all datasets, and while its multi-agent reasoning requires higher inference time, this cost is directly justified by its best performance on all benchmarks, making it ideal for quality-sensitive applications requiring reliable reasoning capabilities. Note that the reranker model is relatively small and reduces the input length to the LLM, thus having minimal impact on inference time. Detailed token consumption for graph construction and inference across different methods are provided in Appendix.G.2.

G.2 Comparison of Token Consumption

Our proposed ToG-3 framework achieves a more favorable balance between inference efficiency and performance. As shown in Table 7, compared to GraphRAG, ToG-3 saves approximately 60% of token consumption during the graph construction phase (an average of 5.03 vs. 12.82 million tokens), which benefits from the dynamic graph construction mechanism that avoids the overhead of pre-building large-scale static knowledge graphs. Although ToG-3’s average inference token consumption per sample (72.1 tokens) is higher than that of GraphRAG (32.3 tokens) and LightRAG (23.1 tokens), this increased inference overhead is the necessary cost for achieving precise multi-hop reasoning—our multi-agent evolution mechanism effectively decomposes complex questions and fo-

Table 6: Computational cost comparison across datasets between Graph-based methods. The best EM score of each dataset are marked in **bold**. ToG-3 achieves the best accuracy with efficient indexing and justified inference cost.

Dataset	Method	Graph Statistics			Indexing Time (h)	Inference Time (s/q)	Avg. EM
		Entities	Relations	Communities			
HotpotQA	ToG-3	37,358	30,987	5,041	12.5	17.13	0.645
	HippoRAG-2	92,145	22,047	-	11.2	4.85	0.612
	GraphRAG	94,376	73,265	10,981	15.8	8.91	0.337
	LightRAG	94,578	76,157	-	12.1	6.54	0.308
2WikiMultihopQA	ToG-3	19,311	21,077	3,417	8.2	15.07	0.527
	HippoRAG-2	48,251	11,540	-	7.6	4.12	0.491
	GraphRAG	50,556	37,840	6,261	10.3	7.45	0.439
	LightRAG	50,177	37,995	-	7.8	5.23	0.420
Musique	ToG-3	32,842	39,134	6,258	9.7	13.34	0.291
	HippoRAG-2	112,270	26,581	-	10.1	4.92	0.212
	GraphRAG	106,042	83,139	9,407	13.2	9.37	0.109
	LightRAG	94,621	75,923	-	10.3	7.12	0.082
Average	ToG-3	29,837	30,399	4,905	10.13	15.18	0.474
	HippoRAG-2	84,222	20,056	-	9.63	4.63	0.438
	GraphRAG	83,658	64,748	8,883	13.10	8.58	0.295
	LightRAG	79,792	63,358	-	10.06	6.30	0.270

cuses on critical evidence through deep iterative query and sub-graph evolution, ultimately translating into superior answer quality (as demonstrated by the performance gains in Table 2). This design trade-off indicates that ToG-3 achieves higher overall efficiency and accuracy by shifting computational resources from the expensive pre-construction phase to the targeted reasoning phase. Note that, since LLM inference speed is comparable across methods, token consumption is directly proportional to the primary time overhead.

H Additional Baselines

As shown in Table 8, under the same experimental setup, we conduct a comprehensive comparison with a range of graph-enhanced RAG baselines proposed in recent years. Across all three multi-hop reasoning benchmarks, ToG-3 significantly outperforms all compared methods on every metric. Specifically, on the HotpotQA dataset, ToG-3 achieves an EM score of 0.654, surpassing the next best performers, Youtu-GraphRAG (0.600) and Graph Counselor (0.580). A similar trend of superior performance is observed on the 2WikiMultihopQA and Musique datasets. The consistent and comprehensive lead of ToG-3 in both EM and F1 scores demonstrates that our proposed dynamic heterogeneous graph evolution and multi-agent collaboration mechanism can more effectively support complex, deep multi-hop reasoning tasks.

I Case Study for ToG-3

This section provides a detailed case study of ToG-3 in deep reasoning task (Figure 6) and broad rea-

soning task (Figure 7 and Figure 8), offering an intuitive demonstration of the execution dynamics of its dual-evolution mechanism—comprising Evolving Query and Evolving Subgraph—across multi-step reasoning processes.

J Graph Visualization Examples

This section details two constructed graph used in our study: the 2WikiMultihopQA subset (exemplifying deep reasoning) and the computer science domain graph from UltraDomain (exemplifying broad reasoning), which are visualized with Neo4j community edition².

2WikiMultihopQA Dataset: Exemplar of Depth Reasoning 2WikiMultihopQA is designed to test depth reasoning—the ability to perform multi-step, sequential inference over entity-relation paths. Each question requires traversing at least two "hops" (e.g., first identifying a person's profession, then linking that profession to a historical event, and finally combining both to answer a causal query). This structure forces models to engage in complex semantic chaining, where errors in early steps propagate, challenging robustness in long-range dependency handling. The dataset's sparse yet densely connected knowledge graphs emphasize precision in step-by-step reasoning over surface-level pattern matching. A structural overview highlighting its multi-hop nature is shown in Figure 9.

Computer Science Domain Graph in UltraDomain: Exemplar of Breadth Reasoning The

²<https://neo4j.com/product/community-edition>



Figure 6: Case Study: Evolving Query and Evolving Sub-Graph for Multi-Hop Reasoning. This example demonstrates how an initial knowledge sub-graph fails to answer a complex question, prompting a decomposition into a sub-query. The Constructor Agent refines the sub-graph with additional biographical facts, enabling the Response Agent to generate the correct answer. The process highlights the dynamic, iterative nature of self-evolving context retrieval.

```

Question:
How do different regression metrics (e.g., RMSE, MAE, R2) vary by model and dataset?

Evidence (from ToG-3):
Mean Squared Error -> is_related_to -> Root Mean Squared Error...
Mean Squared Error -> is_similar_to -> Mean Absolute Error...
summary_23 -> SUMMARY_FOR -> RMSE.MSE emphasizes larger errors...
summary_287 -> SUMMARY_FOR -> MAE. ARIMA model performance evaluated using MAE...

Answer 1(from ToG-3):
Different regression metrics such as RMSE, MAE, and R2 can vary by model and dataset...
RMSE penalizes large errors more heavily due to squared terms, while MAE is robust to outliers...
R2 measures explained variance but can be misleading on small or overfitted datasets...

Answer 2(from GraphRAG):
The provided community summary does not detail specific variations...To evaluate how these
metrics vary:
1. Train different regression models on your datasets...
2. Compute RMSE, MAE, R2 using MLlib.RegresionMetrics...
3. Compare performance across models and datasets...

Evaluation Result (JSON):
{
  "Comprehensiveness": {
    "Winner": "Answer 1",
    "Explanation": "Answer 1 provides a detailed explanation of various regression metrics..."
  },
  "Diversity": {
    "Winner": "Answer 2",
    "Explanation": "Answer 2 presents a different perspective by incorporating specific tools..."
  },
  "Empowerment": {
    "Winner": "Answer 1",
    "Explanation": "Answer 1 empowers the reader by explaining what each metric means..."
  },
  "Overall Winner": {
    "Winner": "Answer 1",
    "Explanation": "Answer 1 is the overall winner as it provides a comprehensive understanding..."
  }
}

```

Figure 7: Case Study: Comparing Regression Metrics Across Models and Datasets. This example illustrates how two reasoning systems answer a technical ML question: GraphRAG emphasizes practical implementation (e.g., using Spark’s MLlib), while ToG3 focuses on theoretical distinctions between RMSE, MAE, and R². An evaluator selects the more comprehensive and empowering answer based on evidence from the knowledge graph.

Question:

What policy changes are needed to foster equitable access to nutritious food?

Evidence (from ToG-3):

summary_216 -> SUMMARY_FOR -> Food Security. ****Supermarket and Food Security****: The reliance on supermarkets for food is viewed as a threat...
Victory Garden -> promoted -> Food Security... Supermarket -> threatens -> Food Security..."
White House Childhood Obesity Task Force -> recommended -> Farm to School...
Food Works -> collaborates.with -> Sauvie Island Organics...
Michael Taylor -> interprets -> Food Modernization Act... background in representing Monsanto...

Answer 1(from ToG-3):

policy changes that could foster equitable access to nutritious food may include implementing subsidies...
increasing funding for farm to school programs in underserved communities...
establishing regulations that prioritize the availability of nutritious options in school meal programs...
and creating educational initiatives to promote awareness about nutrition and healthy eating habits...
these changes could help address disparities in access to nutritious food...

Answer 2(from GraphRAG):

1. ****strengthen conflict of interest policies****: prevent individuals with significant industry ties...
 2. ****enhance transparency and accountability****: implement transparent processes...
 3. ****support diverse food producers****: promote a variety of food producers...
 4. ****expand nutrition education and assistance programs****: enhance nutrition education...
 5. ****ensure regulatory independence****: guarantee that regulatory bodies are independent...
 6. ****promote local food systems****: encourage local food production through grants, subsidies...
 7. ****support community gardens and urban agriculture****: provide funding and resources...
 8. ****promote sustainable farming practices****: offer incentives for sustainable farming...
- These measures aim to create a more resilient and sustainable food system...

Evaluation Result (JSON):

```
{
  {
    "Comprehensiveness": {
      "Winner": "Answer 2",
      "Explanation": "Answer 2 provides a more detailed and structured set of policy changes... "
      "broader scope than Answer 1..."
    },
    "Diversity": {
      "Winner": "Answer 2",
      "Explanation": "Answer 2 presents a wider variety of perspectives... "
      "including regulatory, community, and economic solutions."
    },
    "Empowerment": {
      "Winner": "Answer 1",
      "Explanation": "Answer 1 empowers readers by offering a comprehensive overview..."
      "facilitates informed judgment."
    },
    "Overall Winner": {
      "Winner": "Answer 2",
      "Explanation": "Answer 2 emerges as the overall winner due to its superior"
      "comprehensiveness, diversity, and empowerment."
    }
  }
}
```

Figure 8: Case Study: Policy Recommendations for Equitable Food Access. This example illustrates the full reasoning pipeline: a complex policy question is answered by two different systems (GraphRAG and ToG-3), supported by retrieved knowledge snippets. An evaluator then compares both responses across multiple dimensions, selecting the more comprehensive, diverse, and empowering answer as the winner.

Table 7: Comparison of token consumption for graph construction and inference across different methods.M means Millions.

Method	Avg. Graph Construction Tokens	Avg. Inference Tokens per Sample
ToG-3	5.03M	72.1
GraphRAG	12.82M	32.3
LightRAG	4.92M	23.1
HippoRAG-2	5.01M	20.6

Table 8: Comparison of additional Graph-based RAG methods across multi-hop reasoning benchmarks. The best performance in each column is marked in **bold**.

Method	HotpotQA		2WikiMultihopQA		Musique	
	EM	F1	EM	F1	EM	F1
Youtu-GraphRAG (Dong et al., 2025)	0.600	0.450	0.470	0.230	0.205	0.135
Graph Counselor (Gao et al., 2025)	0.580	0.434	0.464	0.219	0.203	0.137
RAPTOR (Sarhi et al., 2024)	0.580	0.400	0.420	0.200	0.190	0.120
HyperGraphRAG (Luo et al., 2025b)	0.538	0.337	0.456	0.265	0.195	0.124
E ² GraphRAG (Zhao et al., 2025)	0.420	0.080	0.450	0.075	0.130	0.040
Align-GRAG (Xu et al., 2025)	0.442	0.222	0.432	0.251	0.172	0.116
KET-RAG (Huang et al., 2025)	0.452	0.328	0.425	0.221	0.160	0.102
ToG-3 (Ours)	0.654	0.569	0.527	0.291	0.241	0.174

computer science domain graph from UltraDomain represents breadth reasoning—focused on expansive coverage of concepts and their interrelations. It includes a wide range of CS entities (from foundational data structures/algorithms to applied distributed systems/cloud services) and diverse relationship types (e.g., implements, runs_on, contains). This breadth challenges models to navigate a large, heterogeneous concept space, where connections span disparate subfields (e.g., linking a programming language to a database, or an algorithm to hardware). For instance, understanding how Spark relates to Hadoop, Kafka, and multiple programming languages requires integrating knowledge across multiple domains, reflecting the need for broad, cross-concept awareness. A visualization of this graph, illustrating its extensive node and edge diversity, is provided in Figure 10.

K Theoretical Support: Implicit Dynamics of In-Context Learning

The iterative refinement process in MACER and dual-evolving mechanism is not merely heuristic but possesses theoretical grounding through the lens of implicit in-context learning dynamics. Recent work by (Dherin et al., 2025) demonstrates that transformer-based models can perform in-context learning by implicitly modifying their MLP weights through attention mechanisms. We

extend this theoretical framework to explain the convergence properties of our multi-agent reasoning process.

Implicit Weight Updates via Attention Dynamics The trajectory history \mathcal{H}_k serves as an *in-context prompt* that induces implicit low-rank updates to the frozen LLM’s parameters. Specifically, for a transformer module with MLP layer weights W , the context \mathcal{H}_k generates an implicit weight update ΔW_k through the attention mechanism:

$$\Delta W_k = \frac{(W \Delta A_k) A(q)^\top}{\|A(q)\|^2},$$

$$\text{where } \Delta A_k = A(\mathcal{H}_k, q) - A(q). \quad (7)$$

Here, $A(\cdot)$ denotes the activation pattern from the attention layer, $A(q)$ represents the baseline activation without context, and $A(\mathcal{H}_k, q)$ captures the contextualized activation with the full reasoning history. The term ΔA_k quantifies the information injected by the evolving context \mathcal{H}_k . The low-rank nature of ΔW_k ensures efficient and targeted parameter updates without catastrophic forgetting of pre-trained knowledge.

MDP Policy as an Implicit Function of Context Recall from Section 3.3 that the Reflector Agent’s policy π_{ref} maps states $s_k = (q, \mathcal{G}_k, \mathcal{H}_k)$ to actions (sub-queries or STOP). Under the implicit

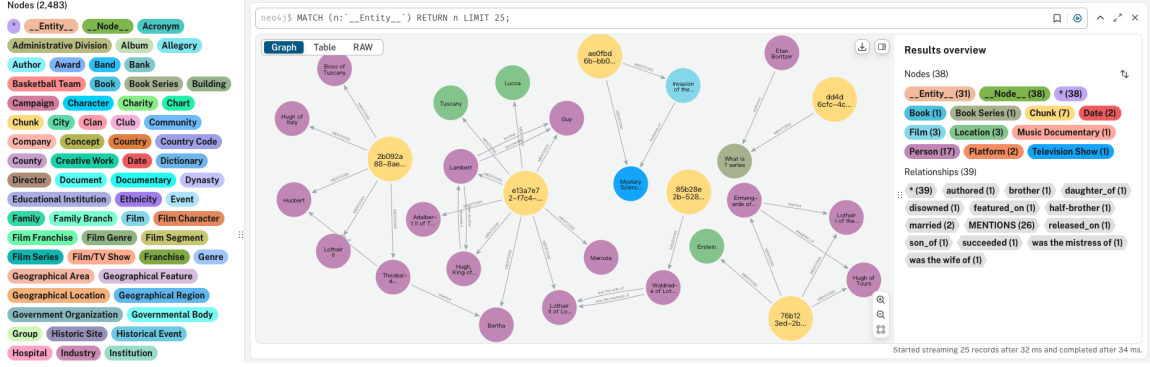


Figure 9: Structural overview of the 2WikiMultihopQA subset, exemplifying depth reasoning through multi-hop entity-relation paths (e.g., traversing ”person → profession → historical event” to answer causal queries).

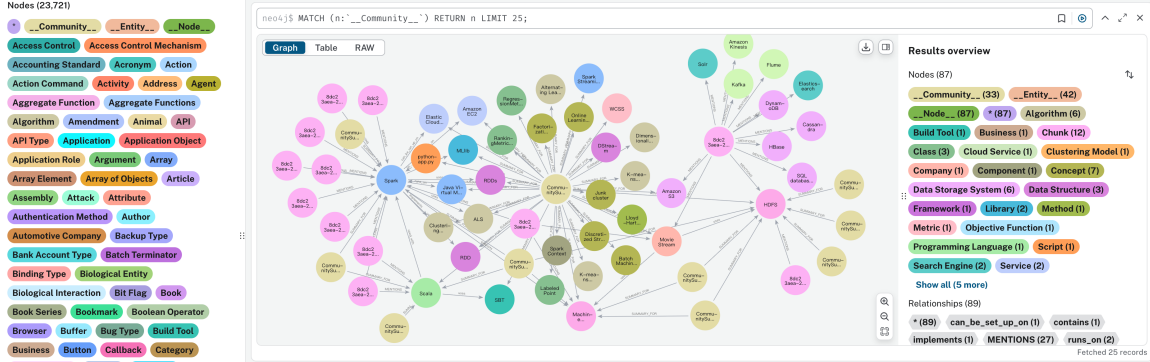


Figure 10: Visualization of the computer science domain graph in UltraDomain, showcasing breadth reasoning via diverse node types (e.g., programming languages like Scala/Spark, frameworks like HDFS/Kafka) and relationship types (e.g., implements, runs_on, contains).

learning view, π_{ref} is not a fixed network but an emergent policy π_k shaped by ΔW_k . Thus, the sequence $\{\pi_k\}_{k=1}^K$ constitutes a trajectory of implicitly adapted policies driven by the evolving context \mathcal{H}_k .

Convergence via Regret Minimization We analyze convergence through the lens of episodic regret minimization in the MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r)$. Let $V_{s_k}^\pi = \mathbb{E}_\pi \left[\sum_{i=k}^K \gamma^{i-k} r_i \mid s_k \right]$ denote the value of policy π at state s_k , and let $V_{s_k}^* = \max_\pi V_{s_k}^\pi$ be the optimal value. The cumulative regret over K steps is:

$$\mathcal{R}(K) = \sum_{k=1}^K (V_{s_k}^* - V_{s_k}^{\pi_k}). \quad (8)$$

We establish sublinear regret growth $\mathcal{R}(K) = o(K)$ under the following mild assumptions:

Assumption 1 (Realizability). *There exists a policy π^* such that $\text{Suff}(q, \mathcal{G}_q^*) = 1$, and π^* is representable by the implicit policy class induced by in-context prompts of the form $(\mathcal{H}; q)$.*

Assumption 2 (Bounded Gradient Norm). *The implicit gradient direction g_k , defined as the reward-sensitive update signal from \mathcal{H}_k , satisfies $\|g_k\| \leq G$ for some constant $G > 0$.*

Under these assumptions, the following properties hold:

Property 1 (Smooth Policy Evolution). The value function evolves smoothly with respect to implicit updates:

$$\|V^{\pi_{k+1}} - V^{\pi_k}\|_\infty \leq L \|g_k\| + \mathcal{O}(\|g_k\|^2), \quad (9)$$

for some Lipschitz constant $L > 0$, ensuring stable policy transitions.

Property 2 (Expected Policy Improvement). Each refinement step yields non-negative expected improvement:

$$\mathbb{E} [V_{s_k}^{\pi_{k+1}} - V_{s_k}^{\pi_k} \mid \mathcal{H}_k] \geq \eta \|g_k\|^2 - \sigma_k, \quad (10)$$

where $\eta > 0$ and $\{\sigma_k\}$ is a martingale difference sequence with $\mathbb{E}[\sigma_k \mid \mathcal{H}_k] = 0$. This follows from the fact that evolving sub-queries generated by the Reflector target knowledge gaps, and the Constructor’s evolving graph refinement increases the likelihood of sufficiency.

Property 3 (Vanishing Implicit Gradient). As the context becomes increasingly informative, the room for improvement diminishes:

$$\lim_{k \rightarrow \infty} \|g_k\| = 0 \quad \text{almost surely.} \quad (11)$$

This is guaranteed by Assumption 1 (Realizability) and the finite horizon K , which ensures the process either reaches a sufficient subgraph ($r_k = 1$) or exhausts its budget.

Together, these properties imply that the sequence $\{\pi_k\}$ converges to a policy π^\dagger satisfying $V_{s_1}^{\pi^\dagger} \geq V_{s_1}^* - \epsilon$ for arbitrarily small $\epsilon > 0$ as $K \rightarrow \infty$. In practice, with a reasonable horizon (e.g., $K = 3$), MACER reliably converges to a sufficient context \mathcal{G}_q^* for faithful answer synthesis.

This analysis establishes that the MACER loop performs an implicit form of policy gradient ascent on the reward landscape defined by context sufficiency, with convergence guarantees rooted in stochastic approximation theory and in-context learning dynamics, providing rigorous foundations for the empirical effectiveness of our reward-based evolving context mechanism.

L Prompt Templates

Our framework employs a multi-stage, prompt-driven reasoning pipeline that integrates structured knowledge graph (KG) extraction, community-based summarization, iterative sub-query decomposition, sub-graph refinement, and faithful answer synthesis. Each stage is governed by a specialized prompt template designed to ensure modularity, interpretability, and factual consistency. The complete sequence of prompts is as follows:

1. **KG Triples Extraction:** As shown in Figure 11, given raw textual input, this prompt instructs the model to extract structured subject-relation-object triples (e.g., entity1 -> relation -> entity2) to construct a fine-grained knowledge sub-graph. This step transforms unstructured text into a queryable graph structure.
2. **Generate Community Summary:** As shown in Figure 12, based on densely connected sub-graphs (communities), this prompt synthesizes a concise natural language summary that captures the core themes and relationships within each community, enabling high-level semantic indexing and retrieval.
3. **Keyword Expansion for Retrieval Augmentation:** As shown in Figure 13, to improve recall in the querying phase, this prompt generates a set of synonyms and related terms from the original query, considering variations in capitalization, pluralization, and common phrasings, separated by delimiter symbols.
4. **Evolving Sub-Query Decomposition:** As shown in Figure 14, for complex multi-hop questions, this prompt recursively decomposes the current query into simpler, context-answerable sub-questions, guided by previously retrieved information and reasoning traces, enabling stepwise information gathering.
5. **Evolving Sub-Graph Refinement:** As shown in Figure 15, this prompt cleans and enhances the retrieved or extracted sub-graph by removing irrelevant triples, normalizing entity names, and optionally filling in strongly supported missing links, thereby improving the signal-to-noise ratio for downstream reasoning.
6. **Final Answer Synthesis:** As shown in Figure 16, in the final stage, the model generates a concise, context-grounded answer using *only* the refined evidence, with explicit instructions to avoid hallucination or reliance on prior knowledge. If the answer cannot be determined, it returns “Unknown” to maintain factual integrity.

These prompts work in concert to enable structured, interpretable, and reliable reasoning over hybrid text-and-graph knowledge sources. And Figure 17 shows the LLM evaluation prompt in the broad reasoning task. Their modular design allows for independent tuning and auditing, making the overall system transparent and robust to noise and ambiguity.

```

-Goal-
Given a text document, identify all entities and their entity types from the text and all relationships among the identified entities.
Given the text, extract up to {max_knowledge_triplets} entity-relation triplets.

-Steps-
1. Identify all entities. For each, extract:
entity_name | entity_type | entity_description

2. Identify all related (source, target) pairs. For each, extract:
source_entity | target_entity | relation | relationship_description

3. Output valid JSON only:
{ "entities": [...], "relationships": [...] }

-An Output Example-
{
  "entities": [
    { "entity_name": "Albert Einstein", "entity_type": "Person", "entity_description": "..."},
    { "entity_name": "Theory of Relativity", "entity_type": "Scientific Theory", "entity_description": "..."},
    { "entity_name": "Nobel Prize in Physics", "entity_type": "Award", "entity_description": "..."}
  ],
  "relationships": [
    { "source_entity": "Albert Einstein", "target_entity": "Theory of Relativity", "relation": "developed", "relationship_description": "..."},
    { "source_entity": "Albert Einstein", "target_entity": "Nobel Prize in Physics", "relation": "won", "relationship_description": "..."}
  ]
}

-Real Data-
#####
text: {text}
#####
output: ;

```

Figure 11: KG Triplets Extraction Prompt Template. The template provides structured instructions for extracting entities and relationships from text, with clear formatting for both input requirements and JSON output format.

```

role="system"
You are provided with a set of relationships from a knowledge graph, each represented as
entity1 -> entity2 -> relation -> relationship_description.
Your task is to create a summary of these relationships. The summary should include: Names
of the entities involved, A concise synthesis of the relationship descriptions. The goal is
to capture the most critical and relevant details that highlight the nature
and significance of each relationship. Ensure the summary is coherent and integrates
information to emphasize key aspects. Avoid redundancy and maintain clarity.

role="user"
#####
text: {community_info}
#####

assistant:
% Generated summary based on {community_info} will appear here.

```

Figure 12: Community Summary Template. This template provides structured instructions for extracting entities and relationships from text, with clear formatting for input specifications and expected JSON-like output format.

```

role="system"
Given some initial query, generate synonyms or related keywords up to {max_keywords} in
total,
considering possible cases of capitalization, pluralization, common expressions, etc.
Provide all synonyms/keywords separated by '^' symbols: 'keyword1^keyword2^...'.
Note: result should be in one line, separated by '^' symbols.

role="user"
----
QUERY: {query_str}
----

assistant:
% Example: KEYWORDS: machine learning^ML learning machines^AI models^neural networks^deep
learning ...

```

Figure 13: Keyword Expansion Prompt Template. This template instructs the model to generate up to {max_keywords} synonyms or related terms for a given query, formatted as a single line separated by '^' symbols.

```

role="system"
The original question is as follows: {query_str}
We have an opportunity to answer some, or all of the question from a knowledge source.
Context information for the knowledge source is provided below, as well as previous reasoning
steps.
Given the context and previous reasoning, return a question that can be answered from the
context.
This question can be the same as the original question, or represent a subcomponent.
It should not be irrelevant to the original question.
If no further information can be extracted, return 'None'.

Examples:
Question: How many Grand Slam titles does the winner of the 2020 Australian Open have?
Knowledge source context: Provides names of the winners of the 2020 Australian Open
Previous reasoning: None
Next question: Who was the winner of the 2020 Australian Open?

Question: How many Grand Slam titles does the winner of the 2020 Australian Open have?
Knowledge source context: Includes biographical info for each winner
Previous reasoning:
- Who was the winner of the 2020 Australian Open?
- The winner was Novak Djokovic.
Next question: How many Grand Slam titles does Novak Djokovic have?

Current Input:
Question: {query_str}
Knowledge source context: {context_str}
Previous reasoning: {prev_reasoning}

assistant:
% Output: <decomposed sub-question> OR 'None'

```

Figure 14: Step-wise Query Evolution and Decomposition Prompt Template. This template guides the model to recursively break down a complex question into answerable sub-questions based on available context and prior reasoning, enabling multi-hop reasoning over knowledge sources.

```

role="system"
You are given a sub-graph extracted from a knowledge graph, represented as a list of triples:
entity1 -> relation -> entity2.
This sub-graph may contain irrelevant, redundant, or incomplete information.
Your task is to refine the sub-graph by:
Removing irrelevant or noisy triples not related to the query, Filling in missing but inferable
relationships (if strongly supported),
Ensuring entity names are normalized (e.g., consistent capitalization, singular/plural).
Return the refined sub-graph in the same triple format, one per line.
If no refinement is needed, return the original sub-graph.
If all triples are irrelevant, return 'None'.

Example Input:
Query: What are the major achievements of Marie Curie?
Sub-graph:
Marie Curie -> won -> Nobel Prize in Physics
Marie Curie -> born in -> Warsaw
Marie Curie -> spouse -> Pierre Curie
Apple Inc. -> founded by -> Steve Jobs

Refined Output:
Marie Curie -> won -> Nobel Prize in Physics
Marie Curie -> won -> Nobel Prize in Chemistry
Marie Curie -> spouse -> Pierre Curie
(Note: Added Chemistry prize based on strong prior knowledge; removed birthplace and unrelated
Apple fact)

Current Input:
Query: {query_str}
Sub-graph:
{subgraph_triples}

assistant:

```

Figure 15: Sub-Graph Evolution and Refinement Prompt Template. This template guides the model to clean, complete, and normalize a noisy or incomplete knowledge sub-graph in response to a given query, improving its relevance and coherence for downstream reasoning.

```

role="system"
Context information is provided below.
You must answer the query using only this context, and not any prior knowledge.
Do not make assumptions or add information not present in the context.
If the answer cannot be determined from the context, respond with 'Unknown'.

-----
{context_str}
-----

Query: {query_str}

Instructions:
Extract or synthesize the answer strictly from the provided context.
Keep the answer concise and factual.
Avoid phrases like "The context states that..." | just give the answer.

assistant:
% Final answer derived solely from context.

```

Figure 16: Final Answer Synthesis Prompt Template. This template enforces faithful response generation based exclusively on retrieved context, a core principle in Retrieval-Augmented Generation (RAG) systems. It suppresses model hallucination by explicitly forbidding the use of prior knowledge.

```

role="system"
You are an expert tasked with evaluating two answers to the same question
based on three criteria: Comprehensiveness, Diversity, and Empowerment.

Evaluation Criteria:
• Comprehensiveness:
How much detail does the answer provide to cover all aspects
and sub-questions implied by the original query?
• Diversity:
How varied and rich is the answer in providing different perspectives,
evidence sources, or reasoning paths?
• Empowerment:
How well does the answer help the reader understand the topic
and make informed judgments or decisions?

Instructions:
Compare Answer 1 and Answer 2 for each criterion.
Choose the better answer and explain why.
Select an overall winner based on balance across all three.

Input:
Question: {query}
Answer 1: {answer1}
Answer 2: {answer2}

Output Format (JSON):
{
  "Comprehensiveness": {
    "Winner": "Answer 1 or Answer 2",
    "Explanation": "..."
  },
  "Diversity": {
    "Winner": "Answer 1 or Answer 2",
    "Explanation": "..."
  },
  "Empowerment": {
    "Winner": "Answer 1 or Answer 2",
    "Explanation": "..."
  },
  "Overall Winner": {
    "Winner": "Answer 1 or Answer 2",
    "Explanation": "..."
  }
}

```

Figure 17: Answer Evaluator Prompt Template. This template guides a dedicated agent to compare two candidate responses along three dimensions: comprehensiveness, diversity, and empowerment, promoting high-quality, informative, and user-centered answer selection in multi-agent systems.