COMPASSRAG: NAVIGATING MULTI-TURN GRAPH RAG WITH DYNAMIC EVIDENCE MEMORY AND RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Retrieval-Augmented Generation (GraphRAG) has emerged as an effective paradigm for multi-hop reasoning by exploiting relational structures among entities and passages. Recent agentic retrieval methods further demonstrate that iterative retrieval and reasoning can improve complex knowledge tasks, motivating the extension of GraphRAG to multi-turn settings. However, existing multi-turn GraphRAG methods face two limitations: evidence from earlier turns is often left implicit and unstructured in the reasoning context, and graph retrieval adapts weakly to evolving evidence needs across turns. Thus, we introduce CompassRAG, which serves as a compass for multi-turn GraphRAG by maintaining dynamic evidence memory and guiding subsequent retrieval. CompassRAG first maintains a Dynamic Evidence Graph (DEG) that records answer-relevant facts as source-linked triples, providing an explicit and traceable working memory across turns. It further adapts graph retrieval to the evolving reasoning objective through a query-aware relation graph, where edge weights are conditioned on the current query to guide propagation toward more relevant evidence. Together, these components enable multi-turn GraphRAG to accumulate verifiable evidence while adaptively navigating the retrieval space. Experiments on three multi-hop question answering benchmarks demonstrate that CompassRAG consistently outperforms existing baselines. Further analysis shows that dynamic evidence memory enables reliable reasoning with reduced dependence on raw retrieval history, while dynamic retrieval improves evidence acquisition.

1 INTRODUCTION

Retrieval-Augmented Generation (RAG) has become a widely adopted paradigm for grounding Large Language Models (LLMs) with external knowledge (Lewis et al., 2020; Gao et al., 2023). By providing LLMs with relevant external evidence at inference time, RAG can reduce hallucinations and improve the factuality and knowledge coverage of generated responses (Huang et al., 2025; Augenstein et al., 2024). However, conventional retrieval largely treats retrieved passages as independent pieces of evidence, making it difficult to capture relational dependencies that span multiple passages. Graph Retrieval-Augmented Generation (GraphRAG) addresses this limitation by explicitly modeling relations among entities and passages, enabling structured retrieval and multi-hop reasoning over interconnected evidence (Edge et al., 2024a; Gutiérrez et al., 2024). Recent GraphRAG systems have demonstrated the effectiveness of graph-based retrieval for knowledge-intensive Question Answering (QA) (Gutiérrez et al., 2025; Wu et al., 2026).

In parallel, recent advances in agentic retrieval have demonstrated the benefits of iterative interaction with external knowledge (Jin et al., 2025; Song et al., 2025). Instead of relying on a single retrieval step, an agent can repeatedly reason over available evidence, identify missing information, formulate new queries, and retrieve additional knowledge. In the context of GraphRAG, multi-turn reasoning has recently been explored along two main directions. Reinforcement Learning (RL)-based approaches train LLMs to acquire multi-turn reasoning and retrieval policies, enabling autonomous query generation and graph retrieval through dedicated policy optimization (Luo et al., 2025; Yu et al., 2026). Learning-free approaches (Gutiérrez et al., 2024; 2025), on the other hand,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

obtain multi-turn behavior without additional policy training by coupling GraphRAG methods with existing iterative reasoning frameworks (Yao et al., 2023; Trivedi et al., 2023).

Despite these advances, the evidence management and retrieval mechanisms underlying multi-turn GraphRAG remain comparatively underexplored. ❶ Evidence discovered in earlier turns is often left implicit and unstructured in the LLM reasoning context. Although retrieved passages can be carried across turns, the intermediate facts established during reasoning are not explicitly represented, making it difficult to selectively retain useful evidence, trace conclusions back to their sources, and avoid accumulating redundant or irrelevant context. ❷ Existing graph retrieval mechanisms are not well suited to iterative evidence acquisition across turns. While the agent may formulate different queries across turns, existing graph-based retrieval methods typically use propagation strengths that are largely static. Consequently, the retriever may repeatedly favor broadly relevant graph connections instead of adapting its exploration toward the evidence currently being sought.

To address those limitations, we present **CompassRAG**, a framework that serves as a compass for navigating multi-turn GraphRAG with dynamic evidence memory and retrieval. CompassRAG maintains a Dynamic Evidence Graph (DEG) as an explicit evidence memory that organizes answer-relevant facts as source-linked triples and incrementally consolidates them through evidence-update operations across reasoning turns. To control context growth of multi-turn reasoning, we introduce memory folding strategies that progressively replace raw retrieval observations with compact evidence representations, reducing dependence on the full retrieval history. CompassRAG further employs a query-aware graph retrieval mechanism, where edge weights are conditioned on the current query to adapt graph propagation to the evolving reasoning objective. The retrieved candidates are further refined using the current DEG, encouraging the acquisition of complementary evidence across turns. Together, these components form a continual loop in which the reasoner updates its evidence memory, formulates the next query, retrieves new evidence, and incorporates it into the DEG. Our contributions are summarized as follows:

- We introduce CompassRAG, a multi-turn GraphRAG framework centered on a Dynamic Evidence Graph (DEG) that explicitly maintains answer-relevant facts together with their supporting sources across reasoning turns.
- We propose a query-aware graph retrieval mechanism that adapts graph propagation to the evolving retrieval objective through query-conditioned edge weighting, enabling more targeted evidence acquisition across turns.
- Experiments on three standard multi-hop QA benchmarks demonstrate that CompassRAG consistently outperforms existing GraphRAG and multi-turn RAG methods.

2 Related Work

2.1 Graph Retrieval-Augmented Generation

RAG grounds LLM generation with external textual evidence, but flat passage retrieval struggles to capture dependencies among information distributed across multiple passages. GraphRAG addresses this limitation by organizing corpus knowledge with explicit structural representations for retrieval and reasoning (Han et al., 2024; Peng et al., 2025; Zhang et al., 2025). Existing GraphRAG methods broadly follow two representative paradigms for organizing and retrieving such structural knowledge. One line of work organizes the corpus into hierarchical structures through recursive clustering or community detection (Sarthi et al., 2024; Edge et al., 2024a;b). Earlier works, like RAPTOR (Sarthi et al., 2024) and MS-GraphRAG (Edge et al., 2024a), recursively cluster and summarize text to build a structured index or hierarchical community at multiple levels of abstraction. Another increasingly prominent line organizes passages and their relational facts into fine-grained graphs, enabling graph algorithms to retrieve and propagate relevance across distributed evidence (Gutiérrez et al., 2024; Guo et al., 2025; Gutiérrez et al., 2025). HippoRAG (Gutiérrez et al., 2024) first employs Personalized PageRank (PPR) to propagate relevance over an extracted relational graph for multi-hop retrieval, while HippoRAG 2 (Gutiérrez et al., 2025) further incorporates passage nodes into graph propagation. Recent work improves this PPR-based retrieval paradigm from the perspective of retrieval graph construction. LinearRAG (Zhuang et al., 2026) removes explicit relations and constructs a lightweight entity-extraction-based tri-graph, whereas

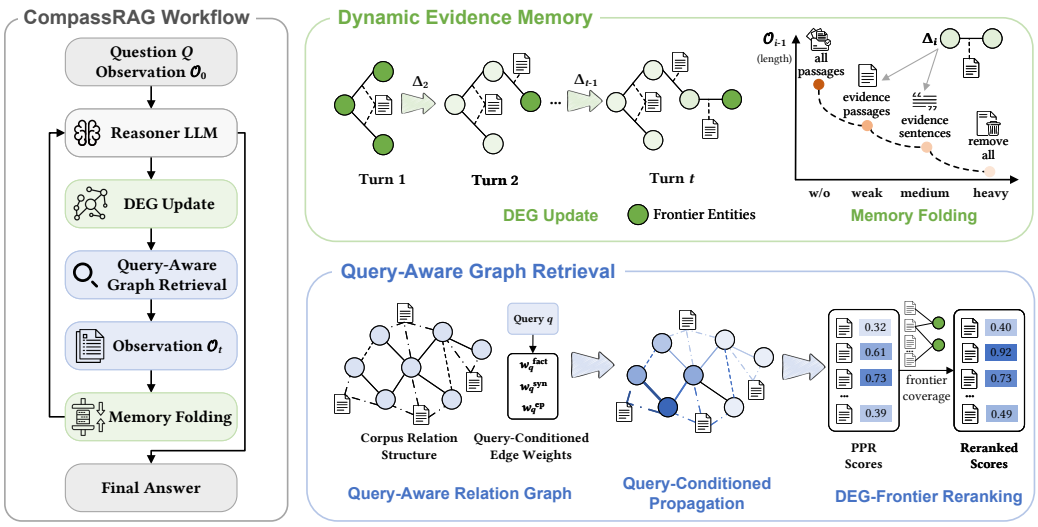

Figure 1: Overview of CompassRAG. The left panel illustrates the multi-turn workflow of CompassRAG. **Dynamic Evidence Memory** maintains source-grounded evidence across turns, while **Query-Aware Graph Retrieval** adapts graph propagation to the current query.

MemGraphRAG (Wu et al., 2026) employs memory-based multi-agent graph construction to resolve conflicting facts. QAFD-RAG (Zhou et al., 2026) introduces query-aware flow diffusion using query–entity edge weights, without explicitly modeling relation semantics.

Despite their effectiveness, most existing PPR-based GraphRAG methods only use relations to establish graph connectivity, ignoring their semantic relevance to the query. Consequently, graph propagation can favor structurally salient connections without sufficiently distinguishing relations relevant to the retrieval objective.

2.2 Agentic Retrieval-Augmented Generation

Agentic RAG extends conventional retrieval into an iterative process in which LLMs repeatedly reason over available information, formulate retrieval queries, and acquire additional evidence (Singh et al., 2025; Zhang et al., 2026). Learning-based methods learn such interaction behaviors through RL algorithms, often using compact open-source LLMs to acquire multi-turn reasoning and retrieval policies (Jin et al., 2025; Song et al., 2025). Representative works such as Search-R1 (Jin et al., 2025) and R1-Searcher (Song et al., 2025) train LLMs to jointly learn reasoning and search behaviors, while Graph-R1 (Luo et al., 2025) and GraphRAG-R1 (Yu et al., 2026) extend policy optimization to graph-based retrieval and reasoning. These methods improve agentic RAG by explicitly optimizing how LLMs reason and formulate queries, but require dedicated policy training. In parallel, learning-free methods directly leverage the reasoning capabilities of strong LLMs and conduct agentic RAG through inference-time reasoning and retrieval interleaving, without additional policy optimization (Yao et al., 2023; Trivedi et al., 2023; Li et al., 2025). Existing GraphRAG retrievers (Gutiérrez et al., 2024; 2025; Wu et al., 2026) can similarly be incorporated into such iterative reasoning frameworks to perform repeated graph retrieval across turns.

However, across both paradigms, accumulated evidence is typically carried forward as retrieved observations and raw reasoning text in the LLM context, leaving intermediate facts implicit and difficult to organize or trace to their supporting sources.

3 CompassRAG

This section presents the technical design of CompassRAG. §3.1 provides a framework overview of the multi-turn inference loop and the interaction among evidence updates and retrieval. §3.2 describes dynamic evidence memory, including its source-grounded evidence representation and

three memory-folding mechanisms. Finally, §3.3 introduces query-aware graph retrieval through the query-aware relation graph, query-conditioned propagation, and DEG-frontier reranking.

3.1 Framework Overview

As illustrated in Figure 1, CompassRAG couples dynamic evidence memory with query-aware graph retrieval in a multi-turn reasoning loop. Given a question Q, CompassRAG maintains a query-specific Dynamic Evidence Graph (DEG) $\mathcal{M}_t$ that records the evidence established up to turn t. Inference starts from an empty DEG, and query-aware graph retrieval first retrieves an initial observation $\mathcal{O}_0$ using the original question. At each subsequent turn, the reasoner LLM incorporates newly established source-grounded facts into the DEG through evidence updates Δ_t and either produces the final answer or formulates a focused retrieval query q_t. Query-aware graph retrieval then uses q_t together with the current DEG $\mathcal{M}_t$ to acquire the next observation $\mathcal{O}_t$, after which previously processed retrieval context is folded according to the accumulated evidence. This process is repeated until the answer is produced or the retrieval budget is reached.

3.2 Dynamic Evidence Memory

Multi-turn reasoning requires previously established evidence to remain accessible without repeatedly carrying the entire retrieval history. To this end, CompassRAG maintains a DEG as an explicit, query-specific evidence memory that records answer-relevant facts together with their supporting sources. The DEG is incrementally updated as new observations are retrieved, while memory folding progressively replaces processed passage text with the structured evidence retained in the DEG.

3.2.1 DEG Update

We represent the evidence accumulated up to turn t as a query-specific DEG

$$\mathcal{M}_t = (\mathcal{V}_t^{\text{ent}} \cup \mathcal{V}_t^{\text{doc}}, \mathcal{E}_t^{\text{rel}}), \tag{1}$$

where $\mathcal{V}_t^{\text{ent}}$ contains entity nodes, $\mathcal{V}_t^{\text{doc}}$ contains source-passage pointers, and each edge $e = (s, r, o) \in \mathcal{E}_t^{\text{rel}}$ represents a directed subject–relation–object fact from entity s to entity o with relation r. Each evidence edge is associated with its supporting passage $d \in \mathcal{V}_t^{\text{doc}}$, explicitly recording which retrieved passage supports the retained fact.

At turn t, the reasoner LLM jointly considers the question Q, the current DEG $\mathcal{M}_{t-1}$, and the folded reasoning context containing the latest retrieved passages $\mathcal{O}_{t-1}$, then produces two types of evidence updates:

$$\Delta_t = \Delta_t^{\text{add}} \cup \Delta_t^{\text{merge}}, \tag{2}$$

where

$$\Delta_t^{\text{add}} = \{(s, r, o, d) \mid d \in \mathcal{O}_{t-1},\ d \models (s, r, o),\ (s, r, o) \text{ is useful for } Q\}. \tag{3}$$

Δ_t^{add} adds source-grounded facts extracted from the latest retrieved passages, while Δ_t^{merge} merges entity aliases by retaining one canonical entity and redirecting the edges of its alias to that entity. These updates are applied to $\mathcal{M}_{t-1}$ to obtain $\mathcal{M}_t$. Only facts grounded in the latest retrieved passages are admitted, ensuring that every retained fact remains explicitly linked to its supporting source. The DEG is then carried forward together with the reasoning history, while the amount of previously retrieved passage text retained in that history is controlled by the memory-folding mechanisms described next.

3.2.2 Memory Folding

As evidence accumulates in the DEG, retaining all previously retrieved passages becomes increasingly redundant. We therefore use memory folding to replace processed retrieval history with compact evidence representations. We call a retrieved passage a cited passage if it supports at least one fact added to the DEG. After processing observation $\mathcal{O}_i$, the set of cited passages is

$$\mathcal{S}_i = \{d \in \mathcal{O}_i \mid \exists (s, r, o, d) \in \Delta_{i+1}\}. \tag{4}$$

Memory folding is applied only after $\mathcal{O}_i$ has been fully processed, so newly retrieved passages are always available in full during the current reasoning turn. To control context growth at different levels, we introduce three folding strategies, namely weak, medium, and heavy folding.

Weak Folding. Weak folding removes passages that contribute no evidence to the DEG and retains cited passages in full:

$$\mathrm{Fold}_{\mathrm{weak}}(\mathcal{O}_i) = \{(d, \mathrm{Pass}(d)) \mid d \in \mathcal{S}_i\}. \tag{5}$$

Medium Folding. Medium folding further reduces each cited passage to sentences that mention entities or relations recorded in the current DEG. The folded observation is

$$\mathrm{Fold}_{\mathrm{medium}}(\mathcal{O}_i) = \{(d, \mathcal{X}_i(d)) \mid d \in \mathcal{S}_i\}, \tag{6}$$

where $\mathcal{X}_i(d)$ denotes the DEG-relevant sentences retained from cited passage d:

$$\mathcal{X}_i(d) = \{x \in \mathrm{Sent}(d) \mid \mathrm{Match}(x, \mathcal{M}_{i+1}) = 1\}, \tag{7}$$

and $\mathrm{Match}(x, \mathcal{M}_{i+1})$ indicates whether sentence x mentions a DEG entity or relation.

Heavy Folding. Heavy folding removes all previously retrieved passage text and retains only the DEG and a lightweight retrieval trace recording how many retrieved passages contribute evidence to the DEG. In particular, a count of zero indicates that the current retrieval returns no useful evidence and provides a signal for subsequent reasoning to revise the retrieval query.

Across all three variants, only the latest DEG state is retained, progressively shifting the reasoning context from raw retrieval history toward structured evidence memory.

3.3 Query-Aware Graph Retrieval

Given a retrieval query, CompassRAG first instantiates a query-aware relation graph with query-conditioned edge weights, and then performs PPR over the resulting graph to retrieve candidate passages, which are further refined using the current DEG through frontier-based reranking.

3.3.1 Query-Aware Relation Graph

Following prior GraphRAG methods (Gutiérrez et al., 2024; 2025), we adopt OpenIE to extract facts from corpus passages and construct a corpus relation structure $\mathcal{R} = (\mathcal{V}, \mathcal{E})$ offline:

$$\mathcal{V} = \mathcal{V}^{\mathrm{ent}} \cup \mathcal{V}^{\mathrm{pass}}, \qquad \mathcal{E} = \mathcal{E}^{\mathrm{fact}} \cup \mathcal{E}^{\mathrm{syn}} \cup \mathcal{E}^{\mathrm{ep}}, \tag{8}$$

where $\mathcal{V}^{\mathrm{ent}}$ and $\mathcal{V}^{\mathrm{pass}}$ denote entity and passage nodes, respectively. Each extracted fact $f = (s, r, o)$ induces a fact edge in $\mathcal{E}^{\mathrm{fact}}$ between entities s and o, with f retained as its semantic representation. $\mathcal{E}^{\mathrm{syn}}$ connects semantically similar entities, and $\mathcal{E}^{\mathrm{ep}}$ links entities to passages in which they occur.

For each retrieval query q, CompassRAG instantiates a query-aware relation graph

$$\mathcal{G}_q = (\mathcal{V}, \mathcal{E}, \mathcal{W}_q), \qquad \mathcal{W}_q = \{w_q^{\mathrm{fact}}, w_q^{\mathrm{syn}}, w_q^{\mathrm{ep}}\}, \tag{9}$$

where $\mathcal{W}_q$ assigns query-conditioned propagation strengths to the three edge types.

3.3.2 Query-Conditioned Propagation

We next specify edge weights in $\mathcal{W}_q$ and perform PPR over the resulting graph. Each fact edge $f = (s, r, o)$ is weighted by the semantic relevance of the complete fact to the query:

$$w_q^{\mathrm{fact}}(f) = \mathrm{sim}(q, f), \tag{10}$$

where $\mathrm{sim}(q, x)$ denotes the nonnegative cosine similarity between the embeddings of q and x. Conditioning on the complete fact allows both its relation semantics and entity context to influence propagation strength.

Entity–passage edges are similarly weighted by query–passage relevance:

$$w_q^{\mathrm{ep}}(v, d) = \mathrm{sim}(q, d). \tag{11}$$

To place synonym edges on the same query-dependent relevance scale as neighboring fact edges, we first compute the mean fact-edge weight incident to each entity:

$$\mu_q(v) = \begin{cases} \dfrac{1}{|\mathcal{E}^{\mathrm{fact}}(v)|} \displaystyle\sum_{e \in \mathcal{E}^{\mathrm{fact}}(v)} w_q^{\mathrm{fact}}(e), & |\mathcal{E}^{\mathrm{fact}}(v)| > 0 \\ 0, & \text{otherwise} \end{cases}, \tag{12}$$

where $\mathcal{E}^{\text{fact}}(v)$ is the set of fact edges connected to entity v. A synonym edge (u, v) is then assigned

$$w_q^{\text{syn}}(u, v) = \frac{\mu_q(u) + \mu_q(v)}{2}, \tag{13}$$

so that its propagation strength adapts to the query-dependent relevance of its local neighborhood.

With these query-conditioned edge weights, we construct the PPR restart distribution following prior work (Gutiérrez et al., 2025) by combining fact-induced entity seeds with dense passage seeds:

$$\boldsymbol{\rho}_q = \boldsymbol{s}_q^{\text{ent}} + \eta \boldsymbol{s}_q^{\text{pass}}, \tag{14}$$

where $\boldsymbol{s}_q^{\text{ent}}$ assigns relevance from the top query-relevant facts to their endpoint entities, $\boldsymbol{s}_q^{\text{pass}}$ contains normalized dense passage scores, and η weights passage seeds. The query–fact and query–passage similarities in Eqs. (10) and (11) reuse the underlying relevance scores used to construct these restart seeds before seed-specific normalization, introducing no additional embedding-based similarity computation. Query-aware PPR then computes the node relevance scores:

$$\boldsymbol{p}_q = (1 - \gamma)\boldsymbol{\rho}_q + \gamma \mathbf{P}_q^\top \boldsymbol{p}_q, \tag{15}$$

where $\mathbf{P}_q$ is the row-normalized transition matrix induced by $\mathcal{W}_q$ and γ is the damping factor. Thus, the current query determines not only where propagation restarts through $\boldsymbol{\rho}_q$, but also how relevance propagates across the graph through the query-conditioned transition matrix $\mathbf{P}_q$. Passage nodes are ranked by their resulting PPR relevance scores.

3.3.3 DEG-Frontier Reranking

At turn t, query-aware PPR over $\mathcal{G}_{q_t}$ assigns a relevance score $p_{q_t}(d)$ to each candidate passage d. We then use the DEG to favor passages that extend the most recently established evidence. The frontier coverage of candidate passage d is

$$c_t(d) = \frac{|\mathcal{F}_t \cap \text{Ent}(d)|}{\max(1, |\mathcal{F}_t|)}, \tag{16}$$

where $\mathcal{F}_t$ contains the subject and object entities from the latest non-empty DEG update, and $\text{Ent}(d)$ contains the corpus entities linked to d. The PPR passage score is then adjusted by

$$\widetilde{p}_t(d) = p_{q_t}(d) \times \begin{cases} 1, & d \in \mathcal{D}_t^{\text{cited}} \\ 1 + \lambda c_t(d), & d \notin \mathcal{D}_t^{\text{cited}} \end{cases}, \tag{17}$$

where $\mathcal{D}_t^{\text{cited}}$ contains all source passages the DEG has already cited, and λ controls the strength of DEG guidance. Already cited passages receive no boost, while uncited passages connected to the current evidence frontier move upward in the ranking. The top K passages under $\widetilde{p}_t$ form the next observation. During initialization, the DEG is empty, so retrieval reduces to query-aware PPR.

Additional method and theoretical details are provided in Appendix A.

4 Experiments

4.1 Experimental Setting

Evaluation Datasets. Following previous works (Gutiérrez et al., 2024; 2025), we conduct experiments on three multi-hop QA benchmarks: MuSiQue (Trivedi et al., 2022), 2WikiMultiHopQA (2Wiki) (Ho et al., 2020), and HotpotQA (Yang et al., 2018). We evaluate QA performance using Exact Match (EM) and F1, retrieval performance using Recall (R) and Complete Recall (CR), and DEG quality using accuracy- and faithfulness-based metrics.

Baselines. We compare against both single-turn and multi-turn methods. For single-turn comparison, we include two non-graph baselines: Direct Inference and Vanilla RAG (Lewis et al., 2020), and compare with representative GraphRAG methods, including RAPTOR (Sarthi et al., 2024), MS-GraphRAG (Edge et al., 2024a), LightRAG (Guo et al., 2025), HippoRAG (Gutiérrez et al., 2024), HippoRAG 2 (Gutiérrez et al., 2025), LinearRAG (Zhuang et al., 2026), QAFD-RAG (Zhou et al., 2026), and MemGraphRAG (Wu et al., 2026). For multi-turn comparison, we include the RL-based

Table 1: Performance comparison between CompassRAG and the baselines in terms of EM and F1. **Bold** denotes the best result, and underlined values denote the second-best result. Weak, Medium, and Heavy denote the three memory-folding variants.

Method	MuSiQue		2Wiki		HotpotQA		Average	
	EM	F1	EM	F1	EM	F1	EM	F1
Single-Turn Methods								
Non-Graph Methods								
Direct Inference	17.60	26.10	36.50	42.80	37.00	47.30	30.37	38.73
Vanilla RAG	34.70	45.70	57.50	61.50	62.80	75.30	51.67	60.83
GraphRAG Methods								
RAPTOR	20.70	28.90	47.30	52.10	56.80	69.50	41.60	50.17
MS-GraphRAG	27.30	38.50	51.40	58.60	55.20	68.60	44.63	55.23
LightRAG	0.50	1.60	9.40	11.60	2.00	2.40	3.97	5.20
HippoRAG	26.20	35.10	65.00	71.80	52.60	63.50	47.93	56.80
HippoRAG 2	37.20	48.60	65.00	71.00	62.70	75.50	54.97	65.03
LinearRAG	30.10	39.79	59.70	66.27	53.60	65.95	47.80	57.34
QAFD-RAG	36.80	48.40	64.80	71.85	60.40	73.33	54.00	64.53
MemGraphRAG	40.40	51.22	64.70	72.19	61.70	74.14	55.60	65.85
Multi-Turn Methods								
RL-based Methods								
R1-Searcher	29.30	37.60	41.70	49.00	46.60	56.70	39.20	47.77
Search-R1	32.00	40.80	51.80	58.90	51.00	62.00	44.93	53.90
GraphRAG-R1	11.42	20.20	29.61	38.32	28.40	39.40	23.14	32.64
Learning-Free Methods								
HippoRAG 2 + IRCoT	41.40	53.82	70.20	79.56	59.10	72.41	56.90	68.60
MemGraphRAG + IRCoT	43.80	55.06	70.30	79.44	58.70	72.08	57.60	68.86
HippoRAG 2 + Search-R1	34.70	47.22	71.50	80.04	56.10	70.01	54.10	65.76
MemGraphRAG + Search-R1	34.80	47.62	71.80	80.25	56.20	70.03	54.27	65.97
CompassRAG (Weak)	**46.30**	57.45	72.90	80.78	62.60	75.91	60.60	71.38
CompassRAG (Medium)	46.20	57.58	72.80	80.54	**63.20**	**76.56**	60.73	71.56
CompassRAG (Heavy)	45.90	**57.72**	**73.20**	**81.32**	**63.20**	76.46	**60.77**	**71.83**

search agents R1-Searcher (Song et al., 2025), Search-R1 (Jin et al., 2025), and GraphRAG-R1 (Yu et al., 2026). We further evaluate graph retrievers under two representative multi-turn interaction frameworks, IRCoT (Trivedi et al., 2023) and Search-R1 (Jin et al., 2025), by pairing each framework with HippoRAG 2 and MemGraphRAG. In these combinations, Search-R1 refers only to its multi-turn interaction framework rather than its released policy LLMs.

Implementation Details. For CompassRAG and other learning-free baselines, we use Llama-3.3-70B-Instruct (Grattafiori et al., 2024) for OpenIE and reasoning, while RL-based baselines use their released Qwen2.5-7B (Yang et al., 2025) policy LLMs. Search-R1 framework combinations use Llama-3.3-70B-Instruct for consistency with the other learning-free methods. We use NV-Embed-v2 (Lee et al., 2025) as the embedding model. All multi-turn methods are allowed up to five interaction rounds, with five passages returned per retrieval request. Hyperparameter settings and additional implementation details are provided in Appendix B.

4.2 Main Results

The main experimental results are presented in Table 1, and we have the following observations.

CompassRAG consistently outperforms all single-turn and multi-turn baselines. The best CompassRAG variant achieves the highest average performance across the three benchmarks, surpassing the strongest baseline by 3.17% in EM and 2.97% in F1. The improvements are also consistent across three benchmarks, demonstrating the effectiveness of CompassRAG for multi-hop QA.

Multi-turn interaction does not uniformly improve GraphRAG retrievers. Both multi-turn interaction frameworks improve HippoRAG 2 and MemGraphRAG on 2Wiki but degrade them on HotpotQA, while only IRCoT yields consistent average gains in both EM and F1. The performance differences are therefore related to both the interaction framework and the features of questions, which we further analyze from a retrieval perspective in §4.4.

Table 2: Ablation results of CompassRAG.

Method	Folding	MuSiQue		2Wiki		HotpotQA		Average	
		EM	F1	EM	F1	EM	F1	EM	F1
CompassRAG	Heavy	45.90	57.72	**73.20**	**81.32**	63.20	76.46	**60.77**	**71.83**
Evidence Memory Ablations									
w/o memory folding	None	**46.40**	**57.79**	72.50	80.64	62.90	76.17	60.60	71.53
w/o DEG-frontier reranking	Weak	45.40	56.50	72.10	79.90	62.60	75.83	60.03	70.74
	Medium	45.50	56.52	72.50	80.43	63.30	**76.81**	60.43	71.25
	Heavy	45.30	56.84	73.10	81.19	62.80	76.06	60.40	71.36
w/o DEG	–	45.00	57.69	71.60	80.03	61.70	74.85	59.43	70.86
Retrieval and Reasoning Ablations									
w/o multi-turn	–	37.50	48.29	68.00	74.76	**63.50**	76.20	56.33	66.42
w/o multi-turn & rel. weight.	–	37.20	48.38	67.60	74.57	62.50	75.69	55.77	66.21
w/o multi-turn & syn. adapt.	–	37.00	47.85	66.30	72.98	63.00	76.13	55.43	65.65

RL-trained compact models still exhibit a substantial gap from learning-free systems built on much larger LLMs. RL-based methods demonstrate that RL can equip 7B-scale models with autonomous multi-turn retrieval and reasoning capabilities. However, their performance remains below learning-free multi-turn methods using Llama-3.3-70B-Instruct, suggesting that policy optimization alone may not fully compensate for the capability gap introduced by model scale.

CompassRAG remains robust under increasingly aggressive memory folding. The weak, medium, and heavy variants achieve comparable performance, even though heavy folding removes all raw passage text. This suggests that the DEG preserves the evidence most relevant to subsequent reasoning, substantially reducing the need to retain previously retrieved passage text.

4.3 Ablation Study

We conduct ablation studies to examine the contributions of evidence memory and multi-turn query-aware graph retrieval, with results reported in Table 2.

Evidence Memory. Removing memory folding yields comparable performance to the full method despite retaining all passages, showing that CompassRAG can compress retrieval history with minimal loss. Removing DEG-frontier reranking degrades average performance across all three folding variants, showing that DEG provides useful guidance for acquiring subsequent evidence. Removing the DEG itself leads to a further performance decrease even when raw retrieval history is retained. This suggests that the DEG serves not only as a compact replacement for retrieval history, but also as a structured evidence representation that facilitates multi-turn reasoning.

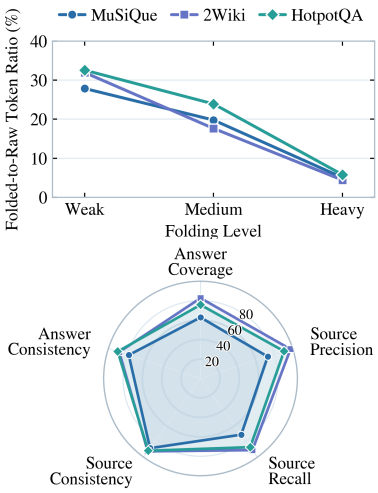

Figure 2: Folded-to-raw token ratios (top) and DEG quality (bottom) across datasets.

Retrieval and Reasoning. Removing multi-turn reasoning causes the largest performance drop, with particularly large decreases on MuSiQue and 2Wiki. On HotpotQA, however, the single-turn variant performs comparably to the multi-turn variants, consistent with the observation in §4.2. Notably, the single-turn variant still outperforms HippoRAG 2 and Mem-GraphRAG, indicating that the gains are not solely from multi-turn interaction. More specifically, removing query-aware graph retrieval reduces the retriever to the HippoRAG 2 formulation and lowers the average F1 by 1.39%, demonstrating the benefit of our retrieval design. We further ablate relation-aware fact weighting (rel. weight.) by replacing complete-fact relevance with query–entity similarity, and synonym adaptation (syn. adapt.) by reverting synonym edges to static semantic similarity between their endpoint entities as in HippoRAG 2. Both ablations reduce average EM and F1, confirming that relation semantics and query-adaptive synonym weighting each improve graph propagation.

4.4 Detailed Analysis

Memory Folding Analysis. We measure the effect of memory folding using the token-length ratio between each folded historical observation and its raw counterpart. As shown in the upper panel of Figure 2, this ratio decreases consistently from weak to medium and heavy folding across all three benchmarks, with heavy folding reducing historical observations to only about 5% of their original length. Together with the comparable QA performance of the three variants in Table 1, these results show that more aggressive folding can substantially reduce historical context without compromising answer quality.

DEG Quality Analysis. We evaluate DEG quality from both accuracy and faithfulness perspectives. Accuracy is measured by **Answer Coverage**, which checks whether the gold answer appears in the DEG, and **Source Precision/Recall**, which compares cited passages with gold supporting sources. Faithfulness is measured by **Source/Answer Consistency**, assessing whether retained evidence is supported by its cited passages and aligned with the final prediction. As shown in the lower panel of Figure 2, both accuracy and faithfulness remain consistently strong across datasets, with 2Wiki showing the strongest overall quality and MuSiQue exhibiting relatively lower evidence coverage.

Retrieval Efficiency Analysis. We compare query-aware graph retrieval with HippoRAG 2, the closest baseline with a similar graph retrieval pipeline, to quantify the overhead introduced by query-conditioned edge weighting and PPR. We report the key latency and the total latency. As shown in Table 3, reusing the query–fact similarities already computed for retrieval keeps query-conditioned edge weighting within 22.6 to 61.7 ms, while PPR latency remains lower than HippoRAG 2. Consequently, the relative difference in total retrieval latency is at most 0.6% across all datasets, demonstrating that query-aware propagation does not materially increase online retrieval latency.

Table 3: Mean retrieval latency (ms/query).

Dataset	HippoRAG 2		Query-Aware G. Ret.		
	PPR	Total	Weight	PPR	Total
MuSiQue	306.2	505.0	61.7	284.1	503.6
2Wiki	258.8	396.4	22.6	253.5	394.0
HotpotQA	305.3	488.2	56.6	284.2	487.3

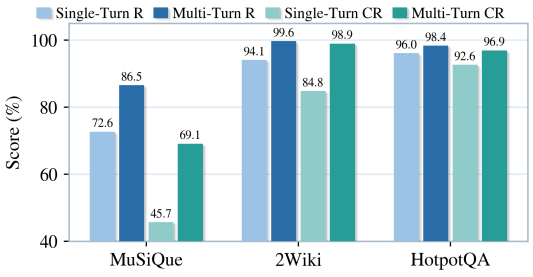

Figure 3: Retrieval performance across datasets.

Multi-Turn Retrieval Analysis. We compare the initial retrieval with the union of passages retrieved across all turns. As shown in Figure 3, multi-turn retrieval consistently improves both Recall (R) and Complete Recall (CR), with the largest gains on MuSiQue. In particular, MuSiQue and 2Wiki gain 23.4% and 14.1% in CR, respectively, showing that follow-up queries effectively recover evidence missed by the initial retrieval. By contrast, HotpotQA already achieves high initial coverage, and the redundant passages induced by multi-turn interaction may even hurt the QA performance. This trend aligns with the share of questions requiring at least three gold passages, which is 48.2% for MuSiQue, 23.5% for 2Wiki, and 0% for HotpotQA. Questions requiring evidence from more sources are harder to cover with a single graph retrieval, leaving greater scope for follow-up queries to recover missing evidence. Overall, multi-turn retrieval is most effective when the initial retrieval leaves substantial evidence gaps. We further examine this effect by stratifying questions according to the number of gold supporting passages in Appendix C.2.

5 Conclusions

In this work, we introduced CompassRAG, a "compass" for multi-turn GraphRAG that tracks evidence and guides retrieval. CompassRAG maintains a DEG as an explicit, source-aware working memory, while its query-aware relation graph conditions propagation on the evolving retrieval objective, allowing the retriever to prioritize edges that are relevant to the evidence currently being sought. Experiments on three multi-hop QA benchmarks demonstrate that CompassRAG consistently outperforms existing baselines while reducing reliance on raw retrieval history, providing an effective and traceable multi-turn GraphRAG solution.

AI USE STATEMENT

In this work, we used generative AI tools solely to edit the manuscript to improve language, clarity, and readability. We did not use generative AI tools to develop research ideas, design the methodology or experiments, or otherwise contribute to the technical content of this work. We have reviewed all AI-assisted edits and take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

To ensure reproducibility and facilitate further research, we provide comprehensive details of our methodology and experimental setup throughout the paper and appendices. §4.1 specifies the evaluated datasets and main experimental configurations, while Appendix B provides detailed dataset statistics, evaluation metrics, and all hyperparameter settings. Appendix A further provides the complete multi-turn inference procedure, Reasoner prompt template, evidence-update operators, and additional details of query-aware graph retrieval. All experiments are conducted on publicly available benchmarks, namely MuSiQue, 2WikiMultiHopQA, and HotpotQA. To further support reproducibility, we provide complete source code and datasets in the supplementary materials.

REFERENCES

Isabelle Augenstein, Timothy Baldwin, Meeyoung Cha, Tanmoy Chakraborty, Giovanni Luca Ciampaglia, David Corney, Renee DiResta, Emilio Ferrara, Scott Hale, Alon Halevy, Eduard Hovy, Heng Ji, Filippo Menczer, Ruben Miguez, Preslav Nakov, Dietram Scheufele, Shivam Sharma, and Giovanni Zagni. Factuality challenges in the era of large language models and opportunities for fact-checking. *Nature Machine Intelligence*, 6(8):852–863, 2024. URL https://doi.org/10.1038/s42256-024-00881-z.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024a. URL https://arxiv.org/abs/2404.16130.

Darren Edge, Ha Trinh, and Jonathan Larson. Lazygraphrag: Setting a new standard for quality and cost. Microsoft Research Blog, 2024b. URL https://www.microsoft.com/en-us/research/blog/lazygraphrag-setting-a-new-standard-for-quality-and-cost/.

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid

El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy,

Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 10746–10761, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.568. URL https://aclanthology.org/2025.findings-emnlp.568/.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: neurobiologically inspired long-term memory for large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pp. 59532–59569, 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=LWH8yn4HS2.

Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A. Rossi, Subhabrata Mukherjee, Xianfeng Tang, Qi He, Zhigang Hua, Bo Long, Tong Zhao, Neil Shah, Amin Javari, Yinglong Xia, and Jiliang Tang. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*, 2024.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020. URL https://aclanthology.org/2020.coling-main.580/.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2), January 2025. ISSN 1046-8188. doi: 10.1145/3703155. URL https://doi.org/10.1145/3703155.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025. URL https://arxiv.org/abs/2503.09516.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-embed: Improved techniques for training LLMs as generalist embedding models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=lgsyLSsDRe.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.276. URL https://aclanthology.org/2025.emnlp-main.276/.

Haoran Luo, Guanting Chen, Qika Lin, Yikai Guo, Fangzhi Xu, Zemin Kuang, Meina Song, Xiaobao Wu, Yifan Zhu, Luu Anh Tuan, et al. Graph-r1: Towards agentic graphrag framework via end-to-end reinforcement learning. *arXiv preprint arXiv:2507.21892*, 2025.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. Graph retrieval-augmented generation: A survey. *ACM Trans. Inf. Syst.*, 44(2), December 2025. ISSN 1046-8188. doi: 10.1145/3777378. URL https://doi.org/10.1145/3777378.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GN921JHCRw.

Aditi Singh, Abul Ehtesham, Saket Kumar, Tala Talaei Khoei, and Athanasios V Vasilakos. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*, 2025.

Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025. URL https://arxiv.org/abs/2503.05592.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. URL https://doi.org/10.1162/tacl_a_00475.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023. URL https://aclanthology.org/2023.acl-long.557/.

Chuanjie Wu, Zhishang Xiang, Yunbo Tang, Zerui Chen, Qinggang Zhang, and Jinsong Su. Memgraphrag: Memory-based multi-agent system for graph retrieval-augmented generation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5407–5418, 2026.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025. URL https://arxiv.org/abs/2412.15115.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018. URL https://aclanthology.org/D18-1259/.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Chuanyue Yu, Kuo Zhao, Yuhan Li, Heng Chang, Mingjian Feng, Xiangzhe Jiang, Yufei Sun, Jia Li, Yuzhi Zhang, Qingyun Sun, et al. Graphrag-r1: Graph retrieval-augmented generation with process-constrained reinforcement learning. In *Proceedings of the ACM Web Conference 2026*, pp. 1398–1409, 2026.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhong-Zhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita Velez, Yue Liao, Hongru WANG, Mengyue Yang, Heng Ji, Jun Wang, Shuicheng YAN, Philip Torr, and LEI BAI. The landscape of agentic reinforcement learning for LLMs: A survey. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=RY19y2RI1O. Survey Certification.

Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Hao Chen, Yilin Xiao, Chuang Zhou, Junnan Dong, Yi Chang, and Xiao Huang. A survey of graph retrieval-augmented generation for customized large language models. *arXiv preprint arXiv:2501.13958*, 2025.

Zhuoping Zhou, Davoud Ataee Tarzanagh, Sima Didari, Wenjun Hu, Baruch Gutow, Oxana Verkholyak, Masoud Faraki, Heng Hao, Hankyu Moon, and Seungjai Min. Query-aware flow diffusion for graph-based rag with retrieval guarantees. In *International Conference on Learning Representations*, volume 2026, pp. 53808–53845, 2026.

Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. Linearrag: Linear graph retrieval augmented generation on large-scale corpora. In *International Conference on Learning Representations*, volume 2026, pp. 147053–147075, 2026.

Algorithm 1 Multi-turn inference with CompassRAG

Input: question Q, corpus relation structure $\mathcal{R}$, and retrieval budget T
Output: answer $\hat{y}$

1: $\mathcal{M}_0 \leftarrow \emptyset,\ q_0 \leftarrow Q$ ▷ DEG initialization
2: $\mathcal{O}_0 \leftarrow \textsc{QueryAwareGraphRetrieval}(\mathcal{R}, q_0, \mathcal{M}_0)$
3: $\mathcal{H}_0 \leftarrow [Q, \mathcal{O}_0, \mathcal{M}_0],\ t \leftarrow 1$
4: **while** $t < T$ **do**
5: $\quad(\Delta_t, a_t) \leftarrow \textsc{Reasoner}(Q, \mathcal{H}_{t-1}, \mathcal{M}_{t-1})$
6: $\quad\mathcal{M}_t \leftarrow \textsc{UpdateDEG}(\mathcal{M}_{t-1}, \Delta_t)$ ▷ DEG update
7: $\quad$**if** <answer> $\in a_t$ **then**
8: $\quad\quad\hat{y} \leftarrow \textsc{ParseAnswer}(a_t)$
9: $\quad\quad$**return** $\hat{y}$
10: $\quad$**if** <search> $\in a_t$ **then**
11: $\quad\quad q_t \leftarrow \textsc{ParseSearch}(a_t)$
12: $\quad\quad\mathcal{O}_t \leftarrow \textsc{QueryAwareGraphRetrieval}(\mathcal{R}, q_t, \mathcal{M}_t)$
13: $\quad\quad\mathcal{H}_t \leftarrow \textsc{Fold}(\mathcal{H}_{t-1}, \mathcal{O}_t, \Delta_t, \mathcal{M}_t)$
14: $\quad t \leftarrow t + 1$
15: $\hat{y} \leftarrow \textsc{ForceAnswer}\big(Q, \mathcal{M}_{t-1}, \bigcup_{i=0}^{t-1} \mathcal{O}_i\big)$
16: **return** $\hat{y}$

A Additional Method Details

A.1 Multi-Turn Inference Procedure

As summarized in Algorithm 1, given a question Q, inference starts from an empty DEG with the original question $q_0 = Q$. CompassRAG first performs query-aware graph retrieval over $\mathcal{R}$ to obtain the initial observation $\mathcal{O}_0$ and initializes the reasoning history $\mathcal{H}_0$ (Lines 1–3). At each subsequent turn, a Reasoner LLM takes the question, current history, and accumulated DEG as input and produces a set of source-grounded evidence updates Δ_t together with an action a_t (Line 5). The evidence updates are first incorporated into $\mathcal{M}_t$ (Line 6), explicitly recording newly established facts before the model decides whether to answer or retrieve additional evidence. If a_t contains an <answer>, inference terminates, and the final answer is returned (Lines 7–9). Otherwise, an <search> specifies a focused query q_t for the remaining information need. Query-aware graph retrieval then uses q_t and the current DEG $\mathcal{M}_t$ to retrieve the next observation $\mathcal{O}_t$ (Lines 10–12). After retrieval, the reasoning history is folded according to the accumulated evidence to control context growth while preserving information useful for subsequent turns (Line 13). The process alternates between explicit evidence accumulation and targeted evidence acquisition until an answer is produced or the retrieval budget is exhausted. In the latter case, ForceAnswer generates the final output from the accumulated DEG and retrieved observations (Line 15).

A.2 Reasoner Prompt Template

Table 4 presents the Reasoner prompt we used. Rather than directly generating an entire evidence graph, the Reasoner incrementally maintains the DEG through two operators. The two placeholders following Rule 2 correspond to the operator definitions in Table 5 and the example in Listing 1, respectively.

A.3 Theoretical Analysis of Query-Aware Graph Retrieval

We analyze the two weighting mechanisms that make graph propagation query-aware. Relation-aware fact weighting determines which extracted facts should carry relevance for the current query, while mean-adaptive synonym weighting calibrates propagation strengths of synonym edges to the relevance of their neighboring facts.

A.3.1 Relation-Aware Fact Weighting

For analytical clarity, we focus on the transition distribution conditional on selecting a fact edge and temporarily omit synonym and entity–passage edges. This simplification is valid because the other

Table 4: Reasoner prompt template used by CompassRAG.

Reasoner Prompt Template

system
You are an evidence-driven reasoning agent solving multi-hop questions.

You maintain a Dynamic Evidence Graph (DEG), a set of (subject, relation, object) triples grounded in retrieved documents. It starts empty and grows as you retrieve.

Input structure
Each <observation>...</observation> block contains one or more retrieved documents. For every document you see:
[chunk-xxx]
<full passage text>

Loop each turn
1. Check the question and current DEG, identifying what is known and what is missing.
2. If the latest <observation>...</observation> contains useful new facts for answering the question, record them into the DEG via <update_evidence>...</update_evidence>.
3. Either issue a <search>...</search> for the missing piece, or provide an <answer>...</answer> if you already have enough evidence.

Response format (always in this order, omitting a block only if it is marked optional)
<think> concise reasoning: what is known, what is missing, next step </think>
<update_evidence> optional JSON array, see below </update_evidence>
<search> focused query for the missing information </search>
(or, instead of search)
<answer> direct final answer only, such as a name, date, number, or short phrase, with no explanation </answer>

Rules
1. <think>...</think> is always required. Exactly one of <search>...</search> and <answer>...</answer> must appear per response.
2. Use <update_evidence>...</update_evidence> to record facts from the latest <observation>...</observation> into the DEG. Read the passage text yourself and extract only the (subject, relation, object) facts useful for answering the question; never fabricate them.
{Evidence Update Operator Definitions}
{Evidence Update Operator Example}

user
Question: {question}
<evidence_graph>
Query: {question}
Evidence Triples (0): []
</evidence_graph>
<observation>
{initial retrieval results}
</observation>

assistant

Table 5: Definitions of the evidence-update operators used by the Reasoner.

Operator	Format and Description
ADD_TRIPLE	{"op": "ADD_TRIPLE", "subject": "...", "relation": "...", "object": "...", "doc_id": "chunk-xxx"} doc_id is the exact chunk ID shown in brackets in the observation.
MERGE_ENTITIES	{"op": "MERGE_ENTITIES", "primary": "...", "secondary": "..."} primary is the name to keep, and secondary is the name to remove. The secondary entity must already be in the DEG and is matched case-insensitively. All of its edges are moved to the primary entity.

Listing 1: Example of the evidence-update operators in the Reasoner prompt.

```
[
  {
    "op": "ADD_TRIPLE",
    "subject": "Marie Curie",
    "relation": "was born in",
    "object": "Poland",
    "doc_id": "chunk-a3f8b2"
  },
  {
    "op": "MERGE_ENTITIES",
    "primary": "Marie Curie",
    "secondary": "M. Curie"
  }
]
```

edge types affect only the total transition mass assigned to fact edges, not their relative allocation among the fact edges incident to the same entity.

To isolate the information encoded by a fact-edge weight, consider a query-independent connectivity weight and a query-dependent endpoint weight:

$$w^{\mathrm{conn}}(s, r, o) = c_{s,o}, \qquad w_q^{\mathrm{ent}}(s, r, o) = g_q(s, o), \tag{18}$$

where $c_{s,o}$ is independent of the query and g_q depends only on the endpoint entities. These two forms represent connectivity-based propagation as in HippoRAG 2 (Gutiérrez et al., 2025) and endpoint-based query weighting such as the query–entity alignment used by QAFD-RAG (Zhou et al., 2026), respectively. In contrast, CompassRAG conditions the edge weight on the complete fact through Equation (10).

Both alternatives in Equation (18) are invariant to the relation when the endpoints are fixed:

$$w^{\mathrm{conn}}(s, r_1, o) = w^{\mathrm{conn}}(s, r_2, o), \qquad w_q^{\mathrm{ent}}(s, r_1, o) = w_q^{\mathrm{ent}}(s, r_2, o). \tag{19}$$

They therefore cannot change their propagation preference when two facts differ only in relation semantics. Complete-fact weighting is not subject to this invariance, as (s, r_1, o) and (s, r_2, o) will receive different scores according to their semantic similarity with q. It can similarly distinguish outgoing edges whose endpoint entities are comparably relevant to the query but whose relations are not.

This additional selectivity can be quantified locally. Let $\mathcal{E}^{\mathrm{fact}}(u)$ contain the d fact edges incident to an entity u, and define its conditional fact-edge transition distribution as

$$\overline{P}_q(f \mid u) = \frac{w_q^{\mathrm{fact}}(f)}{\sum_{e \in \mathcal{E}^{\mathrm{fact}}(u)} w_q^{\mathrm{fact}}(e)}. \tag{20}$$

Suppose a query-relevant edge f^+ has weight at least $a > 0$, while every competing edge has weight at most $b < a$. Then

$$\overline{P}_q(f^+ \mid u) \geq \frac{a}{a + (d-1)b} > \frac{1}{d}. \tag{21}$$

Uniform connectivity assigns only $1/d$ of the fact-edge transition mass to each edge. Endpoint weighting can exceed this baseline when endpoint relevance separates the candidates, but it returns the same preference when their endpoint scores are equal, regardless of their relations. Complete-fact weighting can instead use a relevance margin in the relation text to reduce propagation through mismatched branches.

In our ablation study, complete-fact relevance is replaced by query–entity similarity while the remaining retrieval components are fixed. As shown in Table 2, its lower average performance (*w/o* rel. weight.) empirically supports the benefits of relation-aware fact weighting. Moreover, our single-turn variant in Table 2 outperforms both HippoRAG 2 and QAFD-RAG in Table 1, providing evidence for the benefit of complete-fact weighting relative to connectivity-based and endpoint-based propagation, respectively.

A.3.2 MEAN-ADAPTIVE SYNONYM WEIGHTING

The semantic similarity between two entities serves two distinct purposes in a relation graph: determining whether a synonym edge should exist and controlling how strongly relevance should propagate through that edge. We retain entity similarity for graph construction, but use the query-adaptive mean in Equation (13) as the propagation weight. This separates the semantic validity of a synonym connection from its relevance to the current query.

To examine the effect of this design, let $\mathcal{E}^{\text{fact}}(u)$ and $\mathcal{E}^{\text{syn}}(u)$ denote the fact and synonym edges incident to an entity u, with degrees $d_f(u) = |\mathcal{E}^{\text{fact}}(u)|$ and $d_s(u) = |\mathcal{E}^{\text{syn}}(u)|$, respectively. Its total fact-edge weight is

$$A_q(u) = \sum_{e \in \mathcal{E}^{\text{fact}}(u)} w_q^{\text{fact}}(e) = d_f(u)\mu_q(u). \tag{22}$$

Suppose that each synonym edge (u, v) instead retains the static endpoint similarity $\text{sim}(u, v)$. Among the fact and synonym edges incident to u, the conditional share of outgoing weight assigned to synonym edges is

$$\pi_q^{\text{sim}}(u) = \frac{\sum_{(u,v)\in\mathcal{E}^{\text{syn}}(u)} \text{sim}(u, v)}{d_f(u)\mu_q(u) + \sum_{(u,v)\in\mathcal{E}^{\text{syn}}(u)} \text{sim}(u, v)}. \tag{23}$$

By construction, a synonym edge is retained only when $\text{sim}(u, v) \geq \delta_{\text{syn}}$, where $\delta_{\text{syn}} = 0.8$ in our experiments. Its conditional share therefore satisfies

$$\pi_q^{\text{sim}}(u) \geq \frac{\delta_{\text{syn}} d_s(u)}{d_f(u)\mu_q(u) + \delta_{\text{syn}} d_s(u)}. \tag{24}$$

Consequently, static synonym weights are all at least δ_{syn}, whereas query–fact similarities rarely reach this level. This scale mismatch causes synonym edges to receive disproportionate transition mass, so relevance tends to circulate through synonym connections instead of propagating along evidence-bearing fact edges.

Mean-adaptive weighting removes this fixed high-weight baseline by deriving synonym weights from their neighboring query–fact similarities. The corresponding share becomes

$$\pi_q^{\text{mean}}(u) = \frac{d_s(u)\mu_q(u) + \sum_{(u,v)\in\mathcal{E}^{\text{syn}}(u)} \mu_q(v)}{2d_f(u)\mu_q(u) + d_s(u)\mu_q(u) + \sum_{(u,v)\in\mathcal{E}^{\text{syn}}(u)} \mu_q(v)}. \tag{25}$$

For two synonym edges (u, v_1) and (u, v_2) incident to the same entity u, their contributions to the transition mass of u satisfy

$$\frac{\tau_q((u, v_1) \mid u)}{\tau_q((u, v_2) \mid u)} = \frac{w_q^{\text{syn}}(u, v_1)}{w_q^{\text{syn}}(u, v_2)} = \frac{\mu_q(u) + \mu_q(v_1)}{\mu_q(u) + \mu_q(v_2)}. \tag{26}$$

where $\tau_q(e \mid u)$ denotes the transition mass contributed by edge e at u. Consequently, if $\mu_q(v_1) > \mu_q(v_2)$, then more transition mass is routed from u along the edge to v_1. Since a larger $\mu_q(v)$ indicates that the facts incident to v are more relevant to q on average, mean adaptation creates a relevance-guided local routing effect toward synonym neighbors connected to more relevant evidence. Static similarity may distinguish neighbors by semantic proximity, but it cannot adjust this preference according to their relevance to the current query. Moreover,

$$\min\{\mu_q(u), \mu_q(v)\} \leq w_q^{\text{syn}}(u, v) \leq \max\{\mu_q(u), \mu_q(v)\}, \tag{27}$$

which keeps each synonym edge on the relevance scale of its endpoints. An edge is suppressed when both neighborhoods are irrelevant. If only one endpoint has relevant incident facts, the edge remains active with a reduced weight, allowing limited propagation to the other endpoint. Thus, mean adaptation provides query-aware scale calibration without discarding the semantic connectivity established during graph construction. The consistent improvements over static similarity in Table 2 (*w/o* syn. adapt.) further validate the effectiveness of this query-adaptive weighting strategy.

B Implementation Details

B.1 Datasets

We evaluate all methods on the following three multi-hop QA datasets:

- **MuSiQue** (Trivedi et al., 2022) is constructed by composing connected single-hop questions in a bottom-up manner, containing questions requiring two to four reasoning hops.
- **2WikiMultiHopQA** (Ho et al., 2020) combines Wikipedia text with Wikidata relations and uses logical rules and question templates to ensure that answering requires multiple reasoning steps.
- **HotpotQA** (Yang et al., 2018) is a Wikipedia-based benchmark whose questions require locating and combining evidence from multiple documents, including bridge and comparison questions.

The statistics of the evaluation datasets are shown in Table 6.

Table 6: Statistics of the evaluation datasets.

Dataset	# Questions	# Passages	Avg. # Gold Passages
MuSiQue	1,000	11,656	2.65
2WikiMultiHopQA	1,000	6,119	2.47
HotpotQA	1,000	9,811	2.00

B.2 Evaluation Metrics

QA Performance. We use the standard normalized Exact Match (EM) and token-level F1 metrics. EM tests exact equality after normalization, while F1 is the harmonic mean of token precision and recall.

DEG Quality. We evaluate the final DEG along accuracy and faithfulness dimensions. All metrics are reported as percentages, and we exclude yes/no questions from the two answer-based metrics because a relational DEG may support a binary conclusion without representing "yes" or "no" as an entity.

For accuracy, given an evaluation set $\mathcal{Q}$, Answer Coverage (ACov) is defined as

$$\mathrm{ACov} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathbf{1}\big[\exists a \in \mathcal{A}_q,\, v \in \mathcal{V}_q^{\mathrm{ent}} : m_A(a, v) = 1\big]\,, \tag{28}$$

where $\mathcal{A}_q$ is the set of gold answer aliases, and $\mathcal{V}_q^{\mathrm{ent}}$ is the entity set in the final DEG. The matching function $m_A(x, z)$ returns one when the normalized token sequence of x occurs contiguously in z.

Source Precision (SP) measures how many passages cited by the final DEG are gold supporting passages:

$$\mathrm{SP} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{|\mathcal{C}_q \cap \mathcal{D}_q^{\mathrm{gold}}|}{|\mathcal{C}_q|}, \tag{29}$$

where $\mathcal{C}_q$ denotes the unique source passages cited by relation edges in the final DEG and $\mathcal{D}_q^{\mathrm{gold}}$ denotes the gold supporting passages. Conversely, Source Recall (SR) measures how completely the DEG covers the gold sources:

$$\mathrm{SR} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{|\mathcal{C}_q \cap \mathcal{D}_q^{\mathrm{gold}}|}{|\mathcal{D}_q^{\mathrm{gold}}|}. \tag{30}$$

For faithfulness, Source Consistency (SC) measures whether each DEG entity can be found in a source attached to one of its incident relation edges:

$$\mathrm{SC} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{1}{|\mathcal{V}_q^{\mathrm{ent}}|} \sum_{v \in \mathcal{V}_q^{\mathrm{ent}}} \mathbf{1}[\exists d \in \mathcal{D}_q(v) : m_S(v, d) = 1]\,, \tag{31}$$

where $\mathcal{D}_q(v)$ contains the source passages associated with the relation edges incident to v, and m_S is the normalized contiguous-token matching function used for entity–source matching. Finally, Answer Consistency (ACons) measures whether the predicted answer is represented by a DEG entity:

$$\mathrm{ACons} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathbf{1}\left[\exists v \in \mathcal{V}_q^{\mathrm{ent}} : m_A(\hat{y}_q, v) = 1\right], \tag{32}$$

where $\hat{y}_q$ is the predicted answer. Both m_A and m_S require a contiguous whole-token match.

Retrieval Performance. We evaluate retrieval quality using Recall and Complete Recall. For each question q, let $\mathcal{D}_q^{\mathrm{retr}}$ denote the set of unique passages retrieved across all retrieval rounds and $\mathcal{D}_q^{\mathrm{gold}}$ the gold supporting passages. Recall (R) measures the fraction of gold supporting passages successfully retrieved:

$$\mathrm{R} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{|\mathcal{D}_q^{\mathrm{retr}} \cap \mathcal{D}_q^{\mathrm{gold}}|}{|\mathcal{D}_q^{\mathrm{gold}}|}, \tag{33}$$

while Complete Recall (CR) measures the fraction of questions for which all gold supporting passages are retrieved:

$$\mathrm{CR} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathbf{1}\left[\mathcal{D}_q^{\mathrm{gold}} \subseteq \mathcal{D}_q^{\mathrm{retr}}\right]. \tag{34}$$

B.3 Baselines

We compare CompassRAG with the following single-turn and multi-turn baselines.

Single-Turn Methods. The single-turn baselines include conventional non-graph methods as well as representative GraphRAG approaches, all of which answer the question without iterative reasoner-driven query reformulation.

- **Direct Inference** directly prompts the LLM to answer the question without retrieving any external evidence.
- **Vanilla RAG** (Lewis et al., 2020) follows the standard RAG pipeline by retrieving passages with dense semantic similarity and providing the retrieved passages to the LLM for answer generation.
- **RAPTOR** (Sarthi et al., 2024) recursively embeds, clusters, and summarizes text chunks to construct a tree with progressively coarser levels of abstraction. At inference time, it retrieves relevant nodes from this hierarchy, including original chunks and higher-level summaries.
- **MS-GraphRAG** (Edge et al., 2024a) uses an LLM to construct an entity–relation graph, detects hierarchical communities, and generates a summary report for each community. Its global-search pipeline generates partial responses from relevant community reports and aggregates them into the final response.
- **LightRAG** (Guo et al., 2025) indexes LLM-extracted entities and relations together with their vector representations. Its dual-level retrieval matches low-level query keywords to entities and high-level keywords to relations, and gathers associated graph context and source passages for generation.
- **HippoRAG** (Gutiérrez et al., 2024) constructs a schemaless knowledge graph from OpenIE triples and connects semantically similar phrase nodes with synonym edges. Query named entities are linked to graph nodes to initialize Personalized PageRank (PPR), whose propagated node scores are aggregated over their supporting passages for retrieval.
- **HippoRAG 2** (Gutiérrez et al., 2025) extends HippoRAG by introducing passage nodes and linking them to phrase nodes extracted from each passage. It filters query-relevant triples through LLM-based recognition memory and initializes PPR with both fact-induced phrase seeds and dense passage seeds before ranking passage nodes.
- **LinearRAG** (Zhuang et al., 2026) avoids explicit relation extraction and constructs a relation-free Tri-Graph over entities, sentences, and passages using lightweight entity extraction and semantic linking. Retrieval first activates relevant entities through local semantic bridging and then applies PPR over the entity–passage graph for global importance aggregation and passage ranking.

- **QAFD-RAG** (Zhou et al., 2026) performs query-aware graph retrieval through flow diffusion over a knowledge graph. It constructs a passage–entity graph and dynamically reweights graph edges according to the semantic alignment between the query and endpoint nodes. It then propagates flow from query-relevant seeds through the reweighted graph and ranks passages according to the accumulated flow scores.
- **MemGraphRAG** (Wu et al., 2026) organizes corpus knowledge into a three-layer memory of schemas, facts, and source passages, with a multi-agent indexing process for knowledge extraction and conflict resolution. At retrieval time, it retrieves candidates across the memory layers, projects them onto a hierarchical graph for query-aware node initialization, and applies PPR to rank relevant passages.

Multi-Turn Methods. These baselines interleave reasoning, retrieval, and answer generation, including RL-trained search policies and learning-free reasoning workflows instantiated with GraphRAG retrievers.

- **R1-Searcher** (Song et al., 2025) trains a policy LLM to autonomously invoke external retrieval during multi-step reasoning through a two-stage outcome-based RL procedure.
- **Search-R1** (Jin et al., 2025) trains a policy LLM over multi-turn rollouts that interleave reasoning, search actions, retrieved information, and final-answer generation, using outcome-based RL with retrieved-token masking.
- **GraphRAG-R1** (Yu et al., 2026) trains a policy LLM to perform multi-turn graph retrieval and reasoning through process-constrained RL, using process-level rewards to encourage effective retrieval.
- **IRCoT with GraphRAG Retrievers** (Trivedi et al., 2023; Gutiérrez et al., 2025; Wu et al., 2026) instantiates the learning-free IRCoT workflow with either HippoRAG 2 or MemGraphRAG as the retrieval backend. Starting from the original question, the reasoner alternates between generating an intermediate reasoning step and using the newly generated step to guide the next graph retrieval.
- **Search-R1-style with GraphRAG Retrievers** (Jin et al., 2025; Gutiérrez et al., 2025; Wu et al., 2026) adopts the multi-turn interaction framework of Search-R1 with either HippoRAG 2 or MemGraphRAG as the retrieval backend. To ensure a fair comparison, these variants use only the Search-R1 interaction workflow rather than its learned RL policy, with Llama-3.3-70B-Instruct (Grattafiori et al., 2024) serving as the reasoner throughout the interaction.

B.4 Hyperparameter Settings

Following HippoRAG 2, we set the PPR damping factor γ to 0.5, the passage-seed weight η to 0.05, the synonym threshold δ_{syn} to 0.8 and the decoding temperature to 0. For DEG-frontier reranking, when the DEG is non-empty, we over-retrieve 10 additional passages to form a 15-candidate pool and set the reranking coefficient $\lambda = 2.0$, selected using 1,000 examples from the MuSiQue training set. All multi-turn methods use at most five interaction rounds and return five passages for each retrieval request.

C Additional Experiments

C.1 Sensitivity to DEG-Frontier Reranking

We evaluate the sensitivity of the DEG-frontier reranking to its coefficient λ, which controls the strength of the frontier-coverage boost in Equation (17). We vary λ from 0.1 to 5.0 while keeping all other settings fixed. As shown in Figure 4, performance remains stable throughout the evaluated range. Across all benchmarks, the variation is at most 0.6% EM and 0.9% F1. The selected value $\lambda = 2.0$ obtains the best EM and F1 on MuSiQue and 2Wiki, while close to the best result on HotpotQA. These results indicate that DEG-frontier reranking is robust to its coefficient and does not rely on a narrowly tuned value.

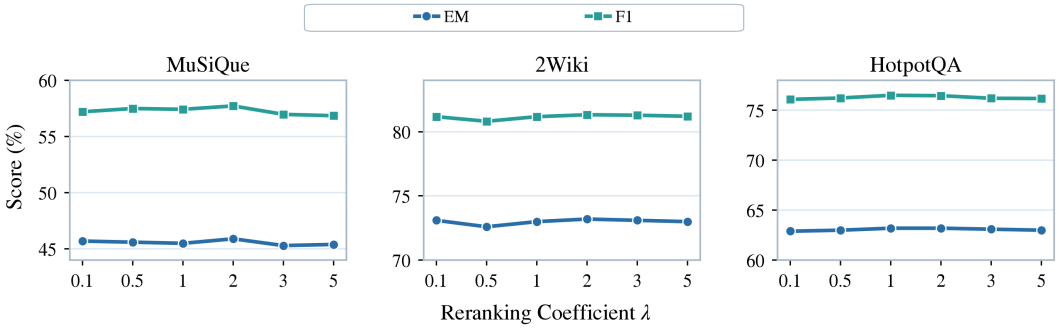

Figure 4: Sensitivity of QA performance to the DEG-frontier reranking coefficient λ.

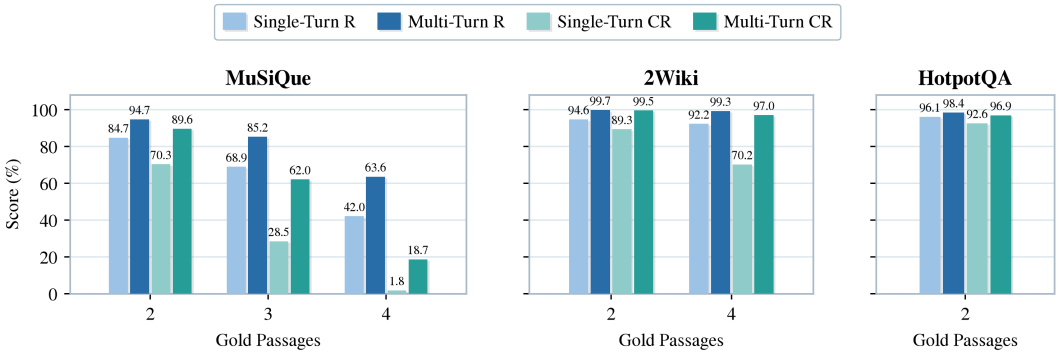

Figure 5: Single-turn and multi-turn retrieval performance across benchmarks, grouped by the number of gold supporting passages.

C.2 Further Analysis of Multi-Turn Retrieval

Extending the analysis in Figure 3, we stratify each benchmark by the number of gold passages required by a question. Single-turn retrieval uses the initial observation, while multi-turn retrieval uses the deduplicated union of passages retrieved across all turns.

MuSiQue provides the clearest evidence-chain trend. As the chain length increases from two to four, single-turn R falls from 84.7% to 42.0%, while single-turn CR falls more sharply from 70.3% to 1.8%. Multi-turn retrieval consistently recovers part of this missing evidence, and its R improvement grows from 10.0% to 21.6%. A similar distinction appears in 2Wiki. Single-turn R is high for both the two- and four-passage groups, at 94.6% and 92.2%, but single-turn CR decreases from 89.3% to 70.2%. Thus, individual gold passages are often retrievable, while obtaining the complete four-passage set with one query is substantially harder. Multi-turn retrieval raises CR to 99.5% and 97.0% for the two groups, showing that focused follow-up queries are particularly effective at completing the larger evidence set. HotpotQA contains only two-passage questions and already obtains 96.1% R and 92.6% CR from the initial retrieval. With little missing gold evidence left to recover, subsequent turns mainly introduce non-gold passages, adding noise that can distract the Reasoner and hurt QA performance.

Overall, the stratified results show that multi-turn retrieval is most beneficial when answering requires a larger set of passages that a single query is unlikely to retrieve completely. When the original query already covers most of a small evidence set, as in HotpotQA, additional retrieval may introduce noise rather than useful evidence.

C.3 Effect of Initial Retrieval

We examine whether retrieving with the original question before the first Reasoner call provides a useful initialization. In the variant without initial retrieval, the Reasoner must first formulate a

Table 7: Effect of initial retrieval on QA performance.

Folding	Initial Retrieval	MuSiQue		2Wiki		HotpotQA		Average	
		EM	F1	EM	F1	EM	F1	EM	F1
Weak	Yes	**46.30**	**57.45**	**72.90**	80.78	62.60	75.91	**60.60**	71.38
	No	44.80	56.20	72.70	**81.36**	**64.00**	**76.69**	60.50	**71.42**
Medium	Yes	**46.20**	**57.58**	**72.80**	80.54	63.20	76.56	**60.73**	**71.56**
	No	44.90	55.66	72.10	**80.93**	**64.50**	**76.98**	60.50	71.19
Heavy	Yes	**45.90**	**57.72**	**73.20**	**81.32**	63.20	**76.46**	**60.77**	**71.83**
	No	45.20	56.22	71.90	80.70	**63.40**	76.14	60.17	71.02

Table 8: Average number of Reasoner interaction turns per question.

Folding	Initial Retrieval	MuSiQue	2Wiki	HotpotQA	Average	Turn Increase
Weak	Yes	**2.33**	**1.53**	**1.40**	**1.75**	–
	No	3.54	2.77	2.53	2.95	68.6% ↑
Medium	Yes	**2.35**	**1.55**	**1.40**	**1.77**	–
	No	3.54	2.78	2.55	2.96	67.2% ↑
Heavy	Yes	**2.42**	**1.53**	**1.42**	**1.79**	–
	No	3.56	2.77	2.54	2.96	65.4% ↑

search query without observing any external evidence. The total retrieval budget remains unchanged between the two settings. As shown in Table 7, initial retrieval consistently benefits MuSiQue, whose longer compositional questions leave more room for the original question to provide a useful starting point. It also improves the average performance under medium and heavy folding, while the two weak-folding variants remain comparable on average. On HotpotQA, where retrieval with the original question already has high coverage, allowing the Reasoner to formulate the first query can occasionally yield slightly higher performance.

The efficiency benefit is more consistent. As reported in Table 8, initial retrieval reduces the average number of Reasoner turns from approximately 3.0 to 1.8 across all folding levels. The final column reports the relative increase caused by removing this initialization, which ranges from 65.4% to 68.6%. By supplying evidence before the first Reasoner call, initial retrieval removes the otherwise evidence-free query-formulation turn and allows the Reasoner to begin directly from retrieved information. It therefore reduces interaction cost while improving average QA performance in most settings.

Table 9: Single-turn and multi-turn answers to the same question.

Dataset	Question	Gold	Single-turn	Multi-turn
MuSiQue	How many episodes are in season 5 of the series with The Bag or the Bat?	12	The information is not available for "Ray Donovan" ×	12 ✓

C.4 Case Study

We use one MuSiQue example to illustrate two aspects of CompassRAG. First, Tables 9 and 10 compare the single-turn result with the multi-turn trajectory, showing how a focused follow-up query completes a missing reasoning step. Second, Table 11 compares how weak, medium, and heavy folding represent the initial retrieval after it has been processed in Turn 1. The trajectory displays the DEG after every turn for clarity, although the Reasoner receives only its latest state during actual inference. For readability, we abbreviate the two evidence-bearing passage identifiers as `chunk-A` and `chunk-B`.

Table 10: Multi-turn interaction example of CompassRAG.

Question. How many episodes are in season 5 of the series with The Bag or the Bat?
Gold answer. 12.

[Initialization]
<observation>
Retrieved Documents (5):
[chunk-A] **The Bag or the Bat.** "The Bag or the Bat" is the pilot episode of the Showtime original series "Ray Donovan", and premiered on June 30, 2013. The series premiere was directed by Allen Coulter and written by series creator Ann Biderman. Prior to the premiere television airing, the episode was uploaded to YouTube by Showtime and was previewed over 150,000 times.
[4 irrelevant passages omitted]
</observation>
<evidence_graph>
Query: How many episodes are in season 5 of the series with The Bag or the Bat?
Evidence Triples (0): []
</evidence_graph>

[Turn 1]
<think> *The question asks for the number of episodes in season 5 of the series with "The Bag or the Bat". From the observation, we can identify that "The Bag or the Bat" is the pilot episode of the series "Ray Donovan". However, the number of episodes in season 5 of "Ray Donovan" is not mentioned in the provided documents. We need to search for this information.* </think>
<update_evidence>
[
{"op": "ADD_TRIPLE", "subject": "The Bag or the Bat", "relation": "is pilot episode of", "object": "Ray Donovan", "doc_id": "chunk-A"}
]
</update_evidence>
<search> Ray Donovan season 5 episode count </search>
<observation>
Retrieved Documents (5):
[chunk-B] **List of Ray Donovan episodes.** Season Episodes Originally aired First aired Last aired 12 June 30, 2013 (2013 - 06 - 30) September 22, 2013 (2013 - 09 - 22) 12 July 13, 2014 (2014 - 07 - 13) September 28, 2014 (2014 - 09 - 28) 12 July 12, 2015 (2015 - 07 - 12) September 27, 2015 (2015 - 09 - 27) 12 June 26, 2016 (2016 - 06 - 26) September 18, 2016 (2016 - 09 - 18) 5 12 August 6, 2017 (2017 - 08 - 06) October 29, 2017 (2017 - 10 - 29)
[chunk-A] **The Bag or the Bat.** ...
[3 irrelevant passages omitted]
</observation>
<evidence_graph>
Query: How many episodes are in season 5 of the series with The Bag or the Bat?
Evidence Triples (1): [(The Bag or the Bat, is pilot episode of, Ray Donovan)]
</evidence_graph>

[Turn 2]
<think> *We have the evidence that "The Bag or the Bat" is the pilot episode of the series "Ray Donovan". From the provided documents, we can see that season 5 of "Ray Donovan" has 12 episodes, as listed in the table in chunk-B. We can now answer the question.* </think>
<update_evidence>
[
{"op": "ADD_TRIPLE", "subject": "Ray Donovan", "relation": "has number of episodes in season 5", "object": "12", "doc_id": "chunk-B"}
]
</update_evidence>
<answer> 12 </answer>
<evidence_graph>
Query: How many episodes are in season 5 of the series with The Bag or the Bat?
Evidence Triples (2): [(The Bag or the Bat, is pilot episode of, Ray Donovan), (Ray Donovan, has number of episodes in season 5, 12)]
</evidence_graph>

Multi-Turn Evidence Acquisition. The initial retrieval identifies Ray Donovan as the bridge entity but does not provide its fifth-season episode count. As shown in Table 10, the Reasoner records this relation in the DEG and converts the unresolved step into a focused query. The next observation supplies the missing value, allowing the Reasoner to extend the DEG and answer correctly. In contrast, the single-turn variant only observes the first half of the evidence chain among its top-ranked passages and reports that the requested information is unavailable.

Memory Folding. After Turn 1, the initial retrieval has been processed and becomes historical context. Across all three folding variants, the Reasoner adds the same bridge fact, (`The Bag or the Bat`, `is pilot episode of`, `Ray Donovan`), to the latest DEG and issues the same follow-up query. Their only difference is how the processed initial observation is represented in the next turn. As shown in Table 11, weak folding removes the four uncited passages but retains the cited passage in full, medium folding keeps only its DEG-relevant sentence, and heavy folding replaces all passage text with a short retrieval trace. The evidence needed for subsequent reasoning remains available through the latest DEG in every variant.

Table 11: The initial observation after Turn 1 under the three memory-folding levels.

Folding Level	Folded Historical Observation
Weak	<observation> Retrieved Documents (1): [chunk-A] **The Bag or the Bat.** "The Bag or the Bat" is the pilot episode of the Showtime original series "Ray Donovan", and premiered on June 30, 2013. The series premiere was directed by Allen Coulter and written by series creator Ann Biderman. Prior to the premiere television airing, the episode was uploaded to YouTube by Showtime and was previewed over 150,000 times. </observation>
Medium	<observation> Retrieved Documents (1): [chunk-A] **The Bag or the Bat.** "The Bag or the Bat" is the pilot episode of the Showtime original series "Ray Donovan", and premiered on June 30, 2013. </observation>
Heavy	<observation> [Searched "How many episodes are in season 5 of the series with The Bag or the Bat?" — 1 relevant doc(s) found, evidence added to DEG] </observation>