LI-RAG: LATENT ITERATIVE RAG FOR COMPLEX QA

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented generation (RAG) grounds large language models (LLMs) in retrieved passages, but for multi-hop questions a single round of retrieval often misses the later evidence, which cannot be located from the question alone. Existing multi-round methods either call an LLM at every retrieval step, so that latency grows with the number of hops, or train the retriever on annotated evidence chains, which are expensive to collect. We propose LI-RAG (Latent Iterative RAG), which trains a multi-round query encoder on the decomposition reasoning paths of an LLM, supervising each round's query embedding with the sub-question the LLM asks at that step and the evidence it selects. At inference, retrieval takes only a few encoder forward passes and no LLM call. Trained on 4,000 questions, LI-RAG outperforms multi-round retrievers trained on the gold evidence of the same questions on four multi-hop benchmarks, performs on par with or better than a retriever trained on 46 times as much annotated data, and remains competitive with a method that calls an LLM at every retrieval step.

1 INTRODUCTION

Retrieval-augmented generation (RAG) grounds the answers of a large language model (LLM) in passages retrieved from a corpus (Lewis et al., 2020). In its standard dense form, the question is embedded once, a precomputed passage index is searched once (Karpukhin et al., 2020), and the LLM is called once to read the retrieved passages. This single round is what makes RAG inexpensive, and it suffices when the evidence can be located from the question alone. Multi-hop questions violate this assumption: their evidence is a chain of passages, and a later passage in the chain may concern an entity that the question never mentions and that only an earlier passage identifies. Following HotpotQA (Yang et al., 2018), we call an entity that a later hop is about and the question does not name the **bridge entity** of the chain. A query computed from the question alone can reach a passage about a bridge entity only through incidental term overlap or an association memorised by the encoder, because the information needed to form that query is in the corpus, not in the question (Qi et al., 2019).

Existing multi-hop systems address this limitation in three ways. The first places an LLM inside the retrieval loop: the LLM decomposes the question into sub-questions, interleaves retrieval with its own reasoning, or is trained with reinforcement learning to issue search queries (Press et al., 2023; Trivedi et al., 2023; Yao et al., 2023; Jin et al., 2025; Song et al., 2025). Because the LLM reads each hop before it writes the next query, it can name the bridge entity, but every hop costs a sequential LLM generation, and the cost grows with the length of the chain. The second is graph-based: HippoRAG and HippoRAG 2 (Gutiérrez et al., 2024; 2025) run Personalized PageRank over a knowledge graph of the corpus, seeded with the nodes matched to the question, and so reach passages linked to the question through shared entities without a query for each hop. An LLM extracts the graph from every passage, however, and HippoRAG 2 makes 23,302 LLM calls to index the MuSiQue corpus alone. The third removes the LLM from retrieval and trains the retriever to take the next hop itself, conditioning each query on the passages retrieved so far (Xiong et al., 2021; Khattab et al., 2021; Erker et al., 2026; Li et al., 2026; Sorokin et al., 2026). These retrievers make no LLM call during retrieval, but most are supervised only with the evidence passages of each question, typically the annotated chains of a multi-hop benchmark; the few exceptions turn LLM-written intermediate steps into training data for a model that composes a text query at every hop (Zhuang et al., 2024), or into privileged information for distilling a retriever that still runs a summarizer at every hop (Xia et al., 2025). Supervision by evidence passages sets the query

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.

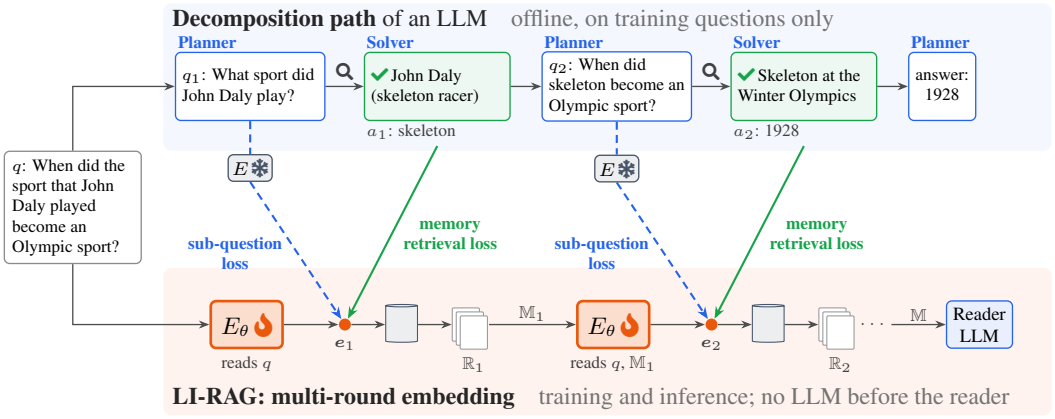


Figure 1: Overview of LI-RAG. Top: the decomposition path of an LLM on a training question, produced once, offline: in each round a planner writes a sub-question q_i, the frozen encoder retrieves candidates for it (magnifier), and a solver marks the evidence (ticks) and gives a sub-answer a_i. Bottom: the fine-tuned encoder E_θ reads the question and the passages found so far and produces the query vector $\boldsymbol{e}_i$ of round i; in training, the sub-question loss draws $\boldsymbol{e}_i$ toward $E(q_i)$ and the memory retrieval loss toward the marked passages. At inference the path is not used. Snowflakes mark frozen components and the flame the fine-tuned encoder.

encoder a target that it cannot fully compute from its inputs, a difficulty known from imitating a teacher that acts on privileged information (Weihs et al., 2021). The embedding of the next passage encodes that passage's content, including facts that neither the question nor the passages retrieved so far contain, and the encoder can approach it only as far as its pre-trained knowledge of that content allows (Reichman & Heck, 2024). On the training questions the remainder can be fitted by memorising the training passages, which transfers poorly to questions about other entities (Lewis et al., 2021; Sciavolino et al., 2021).

We propose LI-RAG (Latent Iterative RAG), a multi-round retriever whose query encoder is trained on the reasoning decomposition path of an LLM. Retrieval runs for a few rounds that differ only in the query embedding (figure 1): a fine-tuned embedding model reads the question together with the passages retrieved so far and produces the vector that the next round searches with, while the passage index stays frozen and the reader remains the only LLM call. The decomposition path comes from an LLM loop run once, offline, over the training questions: in each round a planner writes a sub-question, and a solver marks which of the retrieved passages carry the evidence. We train the encoder to follow this path in embedding space, drawing its query vector at each round toward the embedding of the planner's sub-question and toward the passages the solver marked. Unlike the next passage, the sub-question is a target computable from the encoder's inputs: the planner is instructed to name only entities that the question or an earlier round established, so producing its embedding requires composing information the encoder already receives rather than knowing the content of the passage it leads to. The sub-question thus supervises what each round should look for, and the marked passages what it should retrieve. Figure 2 shows that the trained encoder follows the path on held-out questions: its first- and second-round query vectors lie close to the embeddings of the first and second annotated sub-questions, which play no part in training, whereas both rounds of the untrained encoder lie between the two.

We evaluate LI-RAG on four multi-hop and two single-hop question-answering benchmarks, with the same corpora, reader and LLM judge for every method. Our contributions are as follows.

- A training method that teaches the query encoder to follow the decomposition path of an LLM in embedding space. At each round, a sub-question loss draws the query embedding toward the embedding of the next sub-question on the path, a target the encoder can compose from the question and the passages found so far, rather than only toward the next evidence passage, whose embedding encodes facts absent from those inputs; a memory retrieval loss on the passages the LLM marked as evidence complements it.

- By training on the decomposition paths of a small question set, LI-RAG outperforms multi-round retrievers trained on equivalent gold evidence, and matches or exceeds baselines trained on 46x more annotated data. Furthermore, LI-RAG relies solely on a few encoder forward passes without costly LLM calls. Across four multi-hop benchmarks, this lightweight approach dominates all methods of comparable cost while remaining competitive with computationally expensive, LLM-driven retrieval methods.

2 Related Work

Most multi-hop RAG systems let an LLM decide what to retrieve next, either by decomposing the question into sub-questions (Press et al., 2023; Khot et al., 2023; Nahid & Rafiei, 2026), by interleaving retrieval with the LLM's own generation (Trivedi et al., 2023; Yao et al., 2023; Shao et al., 2023; Jiang et al., 2023; Asai et al., 2024), or by training the LLM to issue search calls with reinforcement learning or on sampled retrieval chains (Jin et al., 2025; Song et al., 2025; Chen et al., 2025; Wang et al., 2025). Every hop then waits for an LLM generation. Adaptive-RAG (Jeong et al., 2024) bypasses the loop for easy questions, and LatentRAG and LAnR (Zheng & Worring, 2026; T et al., 2026) replace the written sub-query with LLM hidden states, but each hop still costs an LLM pass. Graph- and tree-structured indices move the LLM cost to indexing instead, by extracting knowledge graphs, entity and community summaries, or summary trees from every passage (Gutiérrez et al., 2024; 2025; Edge et al., 2025; Guo et al., 2025; Sarthi et al., 2024; Luo et al., 2025). LI-RAG uses an LLM loop only offline, as the teacher of its query encoder, and at inference needs neither a generation per hop nor an index built by LLM extraction.

A second line of work trains the retriever to take the next hop itself. Early systems updated the query with a recurrent unit, generated a text query per hop, or followed hyperlinks (Das et al., 2019; Qi et al., 2019; Asai et al., 2020). Multi-hop dense retrieval (Xiong et al., 2021) re-encodes the question together with the retrieved passage as the next query, and later work extends it with condensed evidence, end-to-end training of the beam over chains, pre-training on pseudo multi-hop questions, or a 7B encoder trained jointly with generation (Khattab et al., 2021; Zhang et al., 2024; Seonwoo et al., 2021; Erker et al., 2026). Other systems carry the state between hops as a pooled passage vector (Li et al., 2026), a text query composed from labelled tokens (Zhuang et al., 2024), an LLM hidden state (Lin et al., 2025), or an embedding trained as a value function with reinforcement learning (Sorokin et al., 2026). Most of these methods are trained on gold evidence passages, typically the annotated chains of a multi-hop benchmark; DistDR (Zhao et al., 2021a) is an exception that learns from question–answer pairs alone. LI-RAG shares with multi-hop dense retrieval and GRITHopper the choice to read the retrieved text with a large encoder, but it is supervised by the sub-questions and evidence judgements of an LLM decomposition loop rather than by gold chains or answer labels, and it trains only the query side.

LLMs also serve as a source of supervision for retrievers. They write hypothetical documents to embed (Gao et al., 2023), train small query rewriters and decomposers (Ma et al., 2023; Xu et al., 2025; Juneja et al., 2023), generate queries and relabel positives for embedding training (Dai et al., 2023; Wang et al., 2024; Shao et al., 2025; Lee et al., 2024), and supervise retrievers through answer likelihood or pseudo-labels (Izacard et al., 2023; Shi et al., 2024; Lee et al., 2025b); these either keep a text generator at inference or learn single-shot relevance rather than a policy over rounds. Retrievers fine-tuned on search-agent trajectories (Zhou et al., 2026; Chen et al., 2026b;a) still rely on the agent to write every query, and multi-step RAG distilled into small language models (Kang et al., 2025; Lee et al., 2025c) still generates text. The closest precedents for our sub-question loss are ConvDR (Yu et al., 2021), which trains a conversational query encoder to match a frozen encoder's embedding of a human-written rewrite, and DiSCo (Lupart et al., 2025), which uses LLM-written rewrites as the teacher of a sparse conversational retriever. LI-RAG carries this idea across hops: the target is the frozen embedding of an LLM planner's sub-question, the positives are the passages its solver marked, and nothing is generated at inference.

3 Approach

LI-RAG retrieves the evidence for a multi-hop question in a few dense retrieval rounds that differ only in the query embedding, and, as in earlier LLM-free multi-hop retrievers (Zhuang et al., 2024;

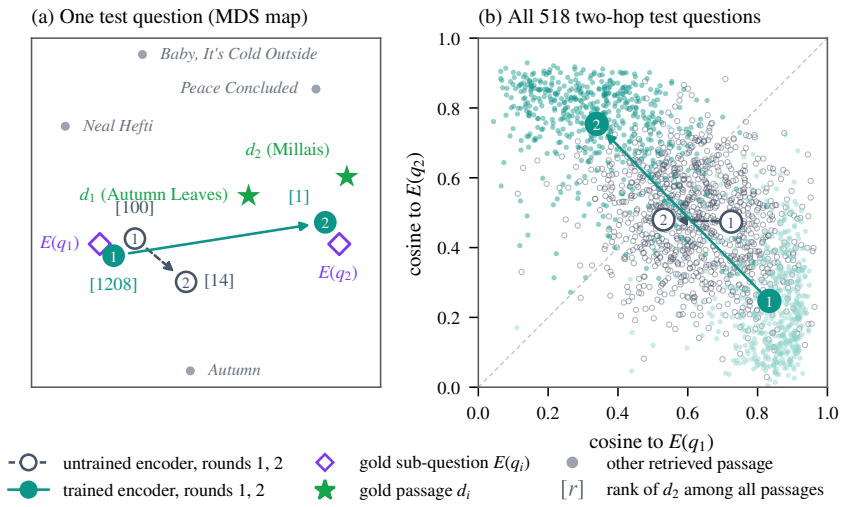


Figure 2: Query vectors of the retrieval loop on MuSiQue, for LI-RAG and for the untrained encoder, whose round 1 is single-round retrieval, $E(q)$; q_1 and q_2 are the annotated sub-questions and d_1 and d_2 the gold passages of the two hops (Appendix A). *(a)* The test question *What date did the creator of Autumn Leaves die?*, by multidimensional scaling of cosine distances. *(b)* Cosine similarity to $E(q_1)$ and $E(q_2)$ for the 518 two-hop test questions; numbered markers are means.

Li et al., 2026; Sorokin et al., 2026), calls no language model during retrieval. Each round is trained on the corresponding round of a decomposition path that an LLM produced offline (figure 1), and at inference the loop runs without it.

3.1 Setting and the retrieval loop

Let $\mathbb{C} = \{d_1, \ldots, d_N\}$ be a corpus of passages and let E be a frozen text encoder that maps a string to a unit vector in $\mathbb{R}^D$. The passage index $\boldsymbol{d}_j = E(d_j)$ is computed once. Single-round dense retrieval (Karpukhin et al., 2020) embeds a question q under a retrieval instruction and returns the k passages with the highest cosine similarity to $E(q)$; such a query does not in general reach a passage about a bridge entity (section 1).

LI-RAG instead runs R rounds and lets the query change between them. Let $\mathbb{M}_i$ denote the ordered list of passages retrieved in rounds 1 to i, with $\mathbb{M}_0 = \emptyset$. Round i computes

$$\boldsymbol{e}_i = E_\theta(q, \mathbb{M}_{i-1}), \tag{1}$$

$$\mathbb{R}_i = \underset{j \in \mathbb{C}}{\text{top-}k}\ \boldsymbol{e}_i^\top \boldsymbol{d}_j, \tag{2}$$

$$\mathbb{M}_i = \mathbb{M}_{i-1} \oplus \mathbb{R}_i, \tag{3}$$

where E_θ is the adapted encoder, top-k returns the k highest-scoring passages in rank order, and $\oplus$ appends the new passages to the list, skipping any that are already present. After the last round the evidence set $\mathbb{M} = \mathbb{M}_R$ holds at most Rk passages. The reader LLM receives the top Rk passages of the **pooled ranking**, which orders the ten highest-scoring passages of every round by their best rank in any round, ties going to the earlier round. Its head is $\mathbb{M}$, whose passages are exactly those in the top k of some round; when later rounds re-retrieve passages already found, so that $|\mathbb{M}| < Rk$, the next-ranked candidates fill the remaining positions. The passage index is never touched: equation 2 is the same matrix product as single-round retrieval.

3.2 Guidance from an LLM decomposition loop

The supervision for E_θ comes from a decomposition loop in which an LLM does, in text, what the embedder is asked to do in embedding space.

The loop. Two roles share one frozen LLM, in the style of iterative decomposition methods (Press et al., 2023; Trivedi et al., 2023). The **planner** is shown the original question and the sub-questions answered so far, $[q, (q_1, a_1), \ldots, (q_{i-1}, a_{i-1})]$, and emits either the next sub-question q_i or the final answer. It is instructed to ask for exactly one fact per sub-question and to mention only entities that are named in q or established by an earlier sub-answer, so that each q_i is a query a dense retriever can act on; it never sees a passage. The sub-question is embedded by the frozen encoder under the retrieval instruction, and the k_T passages closest to $E(q_i)$ form the round's candidate list $\mathbb{R}_i^T$. The **solver** is shown q_i and the numbered candidates and returns a short sub-answer a_i together with the indices of the candidates it needed, $\mathbb{U}_i \subseteq \mathbb{R}_i^T$, and is told that the list contains distractors. The planner can only ask at round one and only answer after R_T retrieval rounds. The rounds of the loop form the **decomposition path** of a question, and its **trace** is the record of that path: the sequence $(q_i, \mathbb{R}_i^T, \mathbb{U}_i, a_i)_{i=1}^n$ of its $n \leq R_T$ rounds together with the final answer.

We run the loop in its dense form, in which every round ranks passages by cosine similarity alone, because that is the retriever the student has: the vector $E(q_i)$ the teacher searched with is then exactly a vector a student round can produce. Traces are built over the training split of each benchmark, on corpora of a size comparable to the evaluation corpora.

Three signals per round. A trace round gives the student three signals (section 3.4). The sub-question q_i, written with the entities the previous rounds established, has a frozen embedding $E(q_i)$ known to retrieve the next hop, and is the target of the sub-question loss. The useful set $\mathbb{U}_i$ separates the candidates that carried the fact from those that merely matched the query, and gives the positives of the memory retrieval loss. The candidate list $\mathbb{R}_i^T$, with its rejected passages $\mathbb{R}_i^T \setminus \mathbb{U}_i$, records the evidence and distractors that a real round shows the next one, and supplies the student's inputs. Gold evidence names the passages, and some benchmarks add their order and a sub-question per hop, but none records the candidates a particular retriever meets.

3.3 The next-hop embedder

E_θ is NV-Embed-v2 (Lee et al., 2025a), a decoder-based embedding model with a latent-attention pooling layer and $D = 4096$, carrying a low-rank adapter (Hu et al., 2022) on the query, key, value and output projections of every decoder layer (rank 16, 13.6M trainable parameters against a 7.85B base). The pooling layer and the passage index are frozen. NV-Embed prefixes its input with an instruction whose tokens are excluded from pooling, and we use two instructions, one per kind of input. Round one embeds the bare question under the retrieval instruction of the single-round baseline (Gutiérrez et al., 2025), *Given a question, retrieve relevant documents that best answer the question.* Every later round embeds the question followed by the passages found so far, numbered in the order they were found, under a next-hop instruction, *Given a question and the passages found so far, retrieve documents that provide the next fact needed to answer the question.*

Two properties follow. First, a freshly initialised adapter is the identity, so round one of the untrained model is exactly single-round dense retrieval, the floor against which every trained model is read. Second, the state passed between rounds is text: the previous query vector is not fed back, since the model can recompute it from q and θ, and what a round adds is what was *found*. Reading the retrieved text rather than pooled vectors follows multi-hop dense retrieval (Xiong et al., 2021) and GRITHopper (Erker et al., 2026), and bets that the entity a later hop needs is legible in an earlier passage.

3.4 Training from decomposition traces

Examples. Each trace round becomes one training example. The input of the example for round i is the question and passages of rounds 1 to $i-1$; its target is the teacher's sub-question q_i; its positives are the useful set $\mathbb{U}_i$. The example for round one is the bare question with target q_1, so training also moves the first query toward the teacher's first hop rather than the whole question.

Inputs. The passages an input shows are drawn from the trace, not from the student's own retrieval, which makes the procedure a form of teacher forcing. At inference each earlier round contributes its top k in rank order, evidence and distractors alike, and the inputs reproduce that shape: for each earlier round $l < i$ the input contains the useful passages $\mathbb{U}_l$ (at most k, best rank first) and m_l

passages sampled uniformly from the rejected ones $\mathbb{R}_l^T \setminus \mathbb{U}_l$, with m_l itself drawn uniformly from $\{0, \ldots, k-|\mathbb{U}_l|\}$; the passages of a round are listed in the teacher's rank order, and a passage already shown for an earlier round is not repeated.

Objective. Let $\boldsymbol{e} = E_\theta(q, \tilde{\mathbb{M}}_{i-1})$ be the embedding of an input built as above. It is pulled toward two targets. The **sub-question loss** distils the teacher's sub-question into the query embedding: its target is the frozen encoder's embedding of the sub-question under the retrieval instruction, $\boldsymbol{t}_i = E(q_i)$, the vector the teacher's round searched with:

$$\mathcal{L}_{\text{sub}} = 1 - \boldsymbol{e}^\top \boldsymbol{t}_i. \tag{4}$$

The **memory retrieval loss** scores the retrieval step that adds passages to $\mathbb{M}$. It is an InfoNCE loss (van den Oord et al., 2019) whose positives are the round's useful passages and whose candidates are every passage of the corpus the question was drawn from:

$$\mathcal{L}_{\text{mem}} = -\log \frac{\sum_{j \in \mathbb{U}_i} \exp(\boldsymbol{e}^\top \boldsymbol{d}_j / \tau)}{\sum_{j=1}^{N} \exp(\boldsymbol{e}^\top \boldsymbol{d}_j / \tau)}, \tag{5}$$

with temperature $\tau = 0.05$. Because the denominator ranges over the full index, the hard negatives of the round, the candidates the solver rejected, are included by construction and no in-batch negatives are needed. Rounds in which the solver marked nothing useful, about 12% of the rounds in the traces we train on, contribute only the sub-question loss. The two losses play different roles: the sub-question loss alone cannot take the student past the teacher's query, whereas the memory retrieval loss optimises the retrieval that is scored and can reward a query that finds evidence the teacher's did not. The training loss is $\mathcal{L} = \lambda_{\text{s}}\mathcal{L}_{\text{sub}} + \lambda_{\text{m}}\mathcal{L}_{\text{mem}}$, with $\lambda_{\text{s}} = \lambda_{\text{m}} = 1$ unless stated. Gradients reach only the query side, so the index built by the frozen encoder is shared by the baseline, the teacher and every checkpoint of the student.

3.5 Inference and cost

At inference the loop of equations 1–3 runs for a fixed number of rounds R with k passages per round, batched over questions. Both constants are small. In the teacher's traces about 90% of the passages the solver marked useful sit within the top three of their round and about 95% within the top five, so $k = 3$ keeps most of the evidence a round surfaces while limiting what the reader must read; with $R = 3$ the reader sees nine passages. Retrieval costs R encoder forward passes and R matrix products per question and no LLM call, whereas the decomposition loop makes two LLM calls per round plus one to answer. Appendix D gives a training-free rule for leaving the loop before R rounds; the results in section 4 use the fixed number.

4 Evaluation & Discussion

4.1 Experimental setup

Benchmarks. We evaluate on four multi-hop benchmarks, MuSiQue (Trivedi et al., 2022) (two to four hops), HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020) (2Wiki) and MultiHop-RAG (Tang & Yang, 2024), and on two single-hop benchmarks, NQ (Kwiatkowski et al., 2019) and PopQA (Mallen et al., 2023). For the five Wikipedia benchmarks we use, unchanged, the 1,000-question test sets and passage corpora released with HippoRAG 2 (Gutiérrez et al., 2025), which hold 6,119 to 11,656 passages; for MuSiQue, HotpotQA and 2Wiki the corpus is the deduplicated union of the candidate paragraphs, gold and distractor, of all its questions. MultiHop-RAG splits 609 news articles into 6,055 chunks of 256 tokens; we use its 2,255 queries with gold evidence, two to four chunks from different articles.

Training. The decomposition loop of section 3.2 (Qwen3.5-35B-A3B (Qwen Team, 2026) as planner and solver, $k_T = 10$, $R_T = 4$) is run over 1,000-question shards of the official training splits, each with a corpus built as above. The main model is trained on the traces of 4,000 MuSiQue questions, and a second model adds 3,000 HotpotQA questions. The checkpoint, the untrained model included, is selected by the answer accuracy of the inference loop on 500 in-domain development questions per dataset, disjoint from the test sets, rather than by loss or recall, since a round that

Table 1: Answer accuracy (%).

	Multi-hop				Single-hop	
Method	MuSiQue	HotpotQA	2Wiki	MultiHop-RAG	NQ	PopQA
HippoRAG 2	<u>52.6</u>	<u>85.4</u>	<u>74.0</u>	<u>72.5</u>	<u>71.4</u>	**72.5**
Single-round dense (NV-Embed-v2)	47.4	84.0	59.3	72.3	71.0	72.2
GRITHopper	52.3	83.0	68.6	70.1	69.4	71.0
TreeHop	45.9	84.6	58.5	72.3	70.6	<u>72.4</u>
L-RAG	21.3	55.3	38.0	52.2	57.2	68.2
Q-RAG	39.7	81.4	71.9	61.5	58.3	62.4
IRCoT	57.6	85.5	84.9	76.1	71.1	72.4
LI-RAG (ours)	**58.5**	**86.1**	**82.6**	**77.3**	**72.1**	<u>72.4</u>

appends a distractor can raise recall while lowering the answer rate. Appendix A gives the training data (Table 4) and hyperparameters.

Baselines. *Single-round dense* retrieval uses the frozen NV-Embed-v2 (Lee et al., 2025a), the backbone of LI-RAG, and *HippoRAG 2* (Gutiérrez et al., 2025) is run with the same LLM and embedding model. *GRITHopper* (Erker et al., 2026) re-encodes the question with the retrieved passages using GritLM-7B (Muennighoff et al., 2025); besides its released checkpoint, GRITHopper-7B*, we retrain its recipe on the same 4,000 MuSiQue training questions as LI-RAG, with the supporting paragraphs annotated in MuSiQue as positives. *TreeHop* (Li et al., 2026) updates the query embedding with a gating network; as its released model is built on BGE-M3 (Chen et al., 2024), we retrain its gates in the same way on NV-Embed-v2 embeddings. GRITHopper and TreeHop run three hops of three passages, as LI-RAG does. *L-RAG* (Lin et al., 2025) maps the hidden state of an LLM into a Contriever (Izacard et al., 2022) space for a second hop, and *Q-RAG* (Sorokin et al., 2026) selects one passage per step with embedders trained by reinforcement learning; we report its best released checkpoint on each benchmark. *IRCoT* (Trivedi et al., 2023) alternates retrieval with chain-of-thought generation, taking each sentence the LLM generates as the next query; we run it with Qwen3.5-35B-A3B and NV-Embed-v2 in place of Codex and BM25. As it calls the LLM at every retrieval step, IRCoT is reported in grey and not ranked. Appendix A gives each configuration.

Protocol and metrics. All methods share the corpora, reader, judge and metric code. The reader is Qwen3.5-35B-A3B with the one-shot chain-of-thought prompt of HippoRAG 2, and **answer accuracy** is the percentage of answers accepted by the yes-or-no judge of MemGAS (Xu et al., 2026). As evidence sets differ in size across methods, every reader except that of L-RAG (Appendix A) receives the top nine passages of the method's pooled ranking (section 3.1), the top ten candidates of every round fused by best rank. R@10 is the recall of the gold passages in the top ten of this ranking, and **full-chain recall** (FCR@10) the fraction of questions with all of them there. Two reader-and-judge passes over identical passage lists differed by up to 1.6 points, so we treat differences below about two points as ties; every run is a single seed. In every table, bold marks the best and underline the second-best value among the ranked methods.

4.2 Main results

Answer accuracy. Among the ranked methods, LI-RAG is the most accurate on all four multi-hop benchmarks (Table 1), ahead of the next best, HippoRAG 2, by 5.9, 0.7, 8.6 and 4.8 points on MuSiQue, HotpotQA, 2Wiki and MultiHop-RAG, and of single-round retrieval by 2.1 to 23.3 points; HotpotQA is a near tie, and on the single-hop sets LI-RAG stays within 1.1 points of both baselines. The retrievers trained on gold chains fall short: L-RAG and Q-RAG are below single-round retrieval on MuSiQue and HotpotQA; TreeHop, retrained on the encoder and questions of LI-RAG, stays within 1.5 points of single-round retrieval on every benchmark; and GRITHopper, retrained on the same questions as LI-RAG, trails it by 3.1 to 14.0 points on the multi-hop sets.

Table 2: Retrieval on each method's pooled ranking (exceptions in Appendix A).

	MuSiQue		HotpotQA		2Wiki		MultiHop-RAG	
Method	R@10	FCR@10	R@10	FCR@10	R@10	FCR@10	R@10	FCR@10
HippoRAG 2	0.839	0.622	0.985	0.971	0.947	0.852	0.568	0.277
Single-round dense (NV-Embed-v2)	0.778	0.504	0.974	0.949	0.806	0.578	0.592	0.286
GRITHopper	0.848	0.673	0.955	0.921	0.917	0.729	0.439	0.192
TreeHop	0.781	0.511	0.974	0.949	0.806	0.578	0.594	0.288
L-RAG	0.524	0.189	0.859	0.739	0.792	0.543	0.199	0.025
Q-RAG	0.651	0.415	0.940	0.914	0.939	0.838	0.298	0.095
IRCoT	0.902	0.756	0.995	0.990	0.994	0.986	0.612	0.317
LI-RAG (ours)	**0.899**	**0.740**	**0.994**	**0.989**	**0.990**	**0.970**	**0.604**	**0.314**

Table 3: LI-RAG against the released GRITHopper-7B*, trained with gold evidence on 46 times as many questions and claims, HotpotQA and 2Wiki among them (Table 4).

	Multi-hop				Single-hop	
Method	MuSiQue	HotpotQA	2Wiki	MultiHop-RAG	NQ	PopQA
	Acc. / R@10					
Single-round dense (NV-Embed-v2)	47.4 / 0.778	84.0 / 0.974	59.3 / 0.806	72.3 / 0.592	71.0 / 0.970	72.2 / 0.563
Single-round dense (GritLM-7B)	45.0 / 0.753	83.4 / 0.970	58.7 / 0.801	74.2 / 0.592	71.1 / 0.963	71.7 / 0.538
GRITHopper-7B*	56.8 / 0.888	**86.7 / 0.996**	71.9 / 0.920	76.0 / 0.557	71.9 / 0.935	70.2 / 0.707
LI-RAG on NV-Embed-v2	**58.5 / 0.899**	86.1 / 0.994	**82.6 / 0.990**	**77.3** / 0.604	**72.1 / 0.971**	72.4 / **0.814**
LI-RAG on GritLM-7B	57.6 / 0.883	85.8 / 0.992	**82.6** / 0.986	76.0 / **0.606**	70.8 / 0.964	**73.0** / 0.723

Retrieval. The pooled rankings follow the same ordering (Table 2), with larger gains in full-chain recall: on MuSiQue and 2Wiki, LI-RAG reaches FCR@10 of 0.740 and 0.970, against 0.673 and 0.852 for the next-best ranked method and 0.504 and 0.578 for single-round retrieval. The accuracy margins track these retrieval margins. On MultiHop-RAG, whose chains span articles that share their entities with many other chunks, every method's FCR@10 is below 0.32.

Against IRCoT. IRCoT is not comparable in cost: before its reader is called, it makes 2.3 to 4.3 LLM calls per question, with 18,000 to 34,000 prompt tokens, where LI-RAG makes none. LI-RAG is nonetheless within 1.2 points of it in answer accuracy on five of the six benchmarks, a tie under our two-point criterion, and 2.3 points below on 2Wiki, where IRCoT makes the most calls (Table 1). In retrieval, the pooled ranking of IRCoT, which fuses 3.3 to 4.3 retrieval steps where LI-RAG fuses three, is ahead by at most 0.008 in R@10 and 0.016 in FCR@10 (Table 2).

4.3 Trace supervision against gold-chain supervision

LI-RAG, a 13.6M-parameter adapter trained on MuSiQue traces, is ahead of the released GRITHopper-7B* by 1.7 points on MuSiQue and 10.7 on 2Wiki and within about two points elsewhere (Table 3), with higher pooled recall on every benchmark except HotpotQA. Trained instead on GritLM-7B, the base of GRITHopper-7B*, whose frozen embeddings answer 0.6 to 2.4 points fewer questions than NV-Embed-v2 in single-round retrieval on the Wikipedia multi-hop sets, LI-RAG stays within 1.3 points of its NV-Embed-v2 version on every benchmark and keeps its 10.7-point lead over GRITHopper-7B* on 2Wiki. With the equal-data result of Table 1, this indicates that traces are the more effective supervision at equal data, with the caveat that our retraining of GRITHopper uses smaller batches than the original. Figure 3 places every method by the size of its

training set: LI-RAG is more accurate than every trained retriever, including one trained on 46 times as much data, and is matched only by IRCoT, which calls the LLM at every retrieval step.

4.4 Analysis

By number of hops. On MuSiQue, LI-RAG leads at every hop count (Table 5 in Appendix B), and its accuracy margin over HippoRAG 2 grows from 2.4 points on two-hop questions to 10.1 and 9.0 on three- and four-hop questions.

Query geometry. Besides lying near $E(q_2)$ (figure 2), the second-round vector of LI-RAG retrieves the second hop's passage more reliably. On the question of panel (a), single-round retrieval ranks the passage about the bridge entity, John Everett Millais, 100th among the 11,656 passages of the corpus, and the second round of LI-RAG ranks it first. Across the 518 two-hop test questions, the second-round vector ranks the second gold passage in the top three for 85.9% of them, against 50.0% for the question embedding and 49.2% for the second round of the untrained encoder.

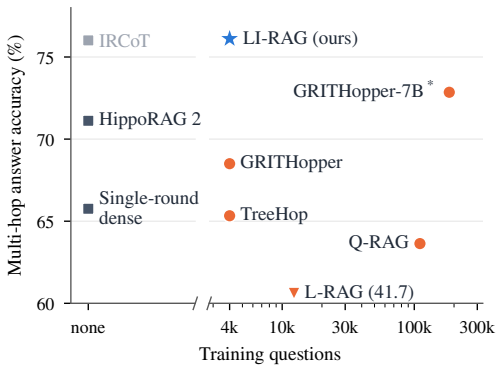


Figure 3: Mean multi-hop answer accuracy against the number of training questions, backbone pre-training excluded. Squares mark training-free methods, grey with LLM calls during retrieval.

Ablations. Without training, the FCR@10 gain of the loop over single-round retrieval is less than a third of that of LI-RAG on MuSiQue and 2Wiki, and negative on MultiHop-RAG (Appendix C). Adding HotpotQA or 2Wiki traces costs 3.8 to 5.5 points on 2Wiki, even with 2Wiki in the training data, for a reason we have not identified, and is otherwise within noise. Without the memory retrieval loss, FCR@10 falls by 0.053 on MultiHop-RAG and by at most 0.008 elsewhere, although the selected epochs also differ. Fine-tuning every decoder weight is worse than the adapter, by 2.9 and 5.7 points on MuSiQue and 2Wiki at a learning rate of 2×10^{-6}, consistent with the query encoder leaving the space of the frozen index.

Discussion and limitations. The gains concentrate where a later round is needed, on multi-hop benchmarks, in full-chain recall and on deeper questions, and traces from one dataset transfer to five others, including a news corpus. The traces took about 26,000 LLM calls, made once, whereas HippoRAG 2 makes 23,302 to index the MuSiQue corpus alone. Four-hop accuracy remains low for every method, at most 26.5% on MuSiQue, and a larger k or a longer loop is not evaluated; the corpora hold at most 11,656 passages and the traces come from one LLM, so web-scale corpora and weaker teachers are untested; and because $\mathbb{M}$ is unfiltered, the reader must tolerate its distractors.

5 Conclusion

We present LI-RAG, a multi-round retriever whose query encoder is trained on the decomposition path of an LLM: a sub-question loss draws each round's query vector toward the frozen embedding of the LLM's sub-question and a memory retrieval loss toward the passages its solver marked, so training uses no gold evidence chain and the reader is the only LLM call at inference. Trained on the traces of 4,000 MuSiQue questions, LI-RAG matched or exceeded every evaluated method of comparable cost on all four multi-hop benchmarks and stayed within noise of single-round retrieval on the two single-hop benchmarks. It was more accurate than GRITHopper retrained on the same questions with gold chains on every multi-hop benchmark, and matched the released GRITHopper-7B*, trained on 46 times as many gold-annotated questions and claims, on MuSiQue while exceeding it on 2Wiki. These results suggest that the decomposition behaviour of an LLM can be distilled into the query encoder, removing the LLM from the retrieval loop without annotated chains.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods and generate synthetic data sets. We have not used generative AI tools to develop theoretical models, propose hypotheses, design research methodology or experiments, clean or reformat datasets, or interpret results; mathematical claims and proofs, translation, and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools to create and edit code, draft and edit parts of the paper, and identify relevant literature. We have reviewed all AI-assisted work: the code was tested, and every reference and reported number was checked against its source. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only publicly released benchmarks built from Wikipedia and news articles, and involves no human subjects and no collection of personal data. The training traces are generated by an LLM over the training splits of these benchmarks, so they and the trained encoder may inherit biases of that LLM and of the underlying corpora. Like other retrieval methods, our work retrieves whatever its corpus contains and does not itself filter harmful or incorrect content.

REPRODUCIBILITY STATEMENT

Section 3 specifies the retrieval loop, the decomposition loop that produces the training traces, the adapted encoder and the training losses. Section 4.1 describes the benchmarks, whose test sets and passage corpora are those released with HippoRAG 2, the construction of the training data from the official training splits, the checkpoint selection and the evaluation protocol, including the variation measured between reader-and-judge passes. Appendix A gives the hyperparameters, the composition of the training data (Table 4), the configuration of every baseline and the details of the reader, judge and metrics. The encoder and the LLM are publicly available models, NV-Embed-v2 and Qwen3.5-35B-A3B. The implementation is currently undergoing internal legal review; we intend to release the code publicly once the required approval is obtained.

REFERENCES

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgVHkrYDH.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

Haodong Chen, Shuai Wang, Yu Yin, Shengyao Zhuang, Guido Zuccon, and Teerapong Leelanupab. Iter: Interaction-aware retrieval for agentic search, 2026a. URL https://arxiv.org/abs/2608.27912.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2318–2335, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.137. URL https://aclanthology.org/2024.findings-acl.137/.

Mingyang Chen, Linzhuang Sun, Tianpeng Li, Haoze Sun, ZhouYijie, Chenzheng Zhu, Haofen Wang, Jeff Pan, Wen Zhang, Huajun Chen, Fan Yang, Zenan Zhou, and weipeng chen. Research: Learning to reason with search for llms via reinforcement learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 85287–85307. Curran Associates, Inc., 2025. doi: 10.52202/

085713-2858. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/7b3a0d3a0864b66f229c0a9c84f9e950-Paper-Conference.pdf.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Jimmy Lin, Akari Asai, and Victor Zhong. AgentIR: Reasoning-aware retrieval for deep research agents. In *Third Conference on Language Modeling*, 2026b. URL https://openreview.net/forum?id=rn8kjxilhF.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. Promptagator: Few-shot dense retrieval from 8 examples. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=gmL46YMpu2J.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. Multi-step retriever-reader interaction for scalable open-domain question answering. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HkfPSh05K7.

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, 2025. URL https://arxiv.org/abs/2404.16130.

Justus-Jonas Erker, Nils Reimers, and Iryna Gurevych. GRITHopper: Decomposition-free multi-hop dense retrieval. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 77–94, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.5. URL https://aclanthology.org/2026.eacl-long.5/.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1762–1777, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.99. URL https://aclanthology.org/2023.acl-long.99/.

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 A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 59532–59569. Curran Associates, Inc., 2024. doi: 10.52202/079017-1902. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/6ddc001d07ca4f319af96a3024f6dbd1-Paper-Conference.pdf.

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 Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21497–21515. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=jKN1pXi7b0.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251): 1–43, 2023. URL http://jmlr.org/papers/v24/23-0037.html.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7036–7050, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.389. URL https://aclanthology.org/2024.naacl-long.389/.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.495. URL https://aclanthology.org/2023.emnlp-main.495/.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Rwhi91ideu.

Gurusha Juneja, Subhabrata Dutta, Soumen Chakrabarti, Sunny Manchanda, and Tanmoy Chakraborty. Small language models fine-tuned to coordinate larger language models improve complex reasoning. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3675–3691, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.225. URL https://aclanthology.org/2023.emnlp-main.225/.

Minki Kang, Jongwon Jeong, Seanie Lee, Jaewoong Cho, and Sung Ju Hwang. Distilling llm agent into small models with retrieval and code tools. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 106501–106538. Curran Associates, Inc., 2025. doi: 10.52202/085713-3553. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/99263f9bf46874e2b8b7f104b5063864-Paper-Conference.pdf.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

Omar Khattab, Christopher Potts, and Matei Zaharia. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 27670–27682. Curran Associates, Inc., 2021. URL

https://proceedings.neurips.cc/paper_files/paper/2021/file/e8b1cbd05f6e6a358a81dee52493dd06-Paper.pdf.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=_nGgzQjzaRy.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pp. 611–626, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702297. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

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 Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 79310–79333, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/c4bf73386022473a652a18941e9ea6f8-Paper-Conference.pdf.

Dosung Lee, Wonjun Oh, Boyoung Kim, Minyoung Kim, Joonsuk Park, and Paul Hongsuck Seo. ReSCORE: Label-free iterative retriever training for multi-hop question answering with relevance-consistency supervision. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 341–359, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.16. URL https://aclanthology.org/2025.acl-long.16/.

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernandez Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftekhar Naim. Gecko: Versatile text embeddings distilled from large language models, 2024. URL https://arxiv.org/abs/2403.20327.

Kyumin Lee, Minjin Jeon, Sanghwan Jang, and Hwanjo Yu. StepER: Step-wise knowledge distillation for enhancing reasoning ability in multi-step retrieval-augmented language 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. 29501–29523, Suzhou, China, November 2025c. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1500. URL https://aclanthology.org/2025.emnlp-main.1500/.

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. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. Question and answer test-train overlap in open-domain question answering datasets. In Paola Merlo, Jorg Tiedemann, and Reut

Tsarfaty (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1000–1008, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.86. URL https://aclanthology.org/2021.eacl-main.86/.

Zhonghao Li, Kunpeng Zhang, Jinghuai Ou, Shuliang Liu, and Xuming Hu. Treehop: Efficient embedding-level query rewriter, 2026. URL https://arxiv.org/abs/2504.20114.

Jiaen Lin, Jingyu Liu, and Yingbo Liu. Optimizing multi-hop document retrieval through intermediate representations. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15798–15809, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.816. URL https://aclanthology.org/2025.findings-acl.816/.

Linhao Luo, Zicheng Zhao, Reza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. Gfm-rag: Graph foundation model for retrieval augmented generation. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 36371–36405. Curran Associates, Inc., 2025. doi: 10.52202/085713-1222. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/33ca0b1102b54c191a9a45a05adafaf4-Paper-Conference.pdf.

Simon Lupart, Mohammad Aliannejadi, and Evangelos Kanoulas. Disco: Llm knowledge distillation for efficient sparse retrieval in conversational search. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, pp. 9–19, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715921. doi: 10.1145/3726302.3729966. URL https://doi.org/10.1145/3726302.3729966.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5303–5315, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.322. URL https://aclanthology.org/2023.emnlp-main.322/.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

Niklas Muennighoff, Hongjin SU, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 45544–45613, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/70cf215430492f7d34830a24e744b3f1-Paper-Conference.pdf.

Md Mahadi Hasan Nahid and Davood Rafiei. Prism: Agentic retrieval with llms for multi-hop question answering, 2026. URL https://arxiv.org/abs/2510.14278.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378/.

Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D. Manning. Answering complex open-domain questions through iterative query generation. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2590–2602, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1261. URL https://aclanthology.org/D19-1261/.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Benjamin Reichman and Larry Heck. Dense passage retrieval: Is it retrieving? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13540–13553, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.791. URL https://aclanthology.org/2024.findings-emnlp.791/.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher Manning. Raptor: Recursive abstractive processing for tree-organized retrieval. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 32628–32649, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8a2acd174940dbca361a6398a4f9df91-Paper-Conference.pdf.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. Simple entity-centric questions challenge dense retrievers. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6138–6148, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.496. URL https://aclanthology.org/2021.emnlp-main.496/.

Yeon Seonwoo, Sang-Woo Lee, Ji-Hoon Kim, Jung-Woo Ha, and Alice Oh. Weakly supervised pre-training for multi-hop retriever. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 694–704, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.62. URL https://aclanthology.org/2021.findings-acl.62/.

Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen tau Yih, Pang Wei Koh, and Luke Zettlemoyer. ReasonIR: Training retrievers for reasoning tasks. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=kkBCNLMbGj.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9248–9274, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.620. URL https://aclanthology.org/2023.findings-emnlp.620/.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8371–8384, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.463. URL https://aclanthology.org/2024.naacl-long.463/.

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, 2025. URL https://arxiv.org/abs/2503.05592.

Artyom Sorokin, Nazar Buzun, Aleksandr Anokhin, Egor KONSTANTINOVICH VEDERNIKOV, Petr Anokhin, Mikhail Burtsev, and Evgeny Burnaev. Q-RAG: Long context multi-step retrieval via value-based embedder training. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=MS9nWFY7LG.

Ha Lan N. T, Minh-Anh Nguyen, and Dung D. Le. Latent abstraction for retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2604.17866.

Yixuan Tang and Yi Yang. Multihop-RAG: Benchmarking retrieval-augmented generation for multi-hop queries. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=t4eB3zYWBK.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.557. URL https://aclanthology.org/2023.acl-long.557/.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. URL https://arxiv.org/abs/1807.03748.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11897–11916, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.642. URL https://aclanthology.org/2024.acl-long.642/.

Liang Wang, Haonan Chen, Nan Yang, Xiaolong Huang, Zhicheng Dou, and Furu Wei. Chain-of-retrieval augmented generation. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 59888–59915. Curran Associates, Inc., 2025. doi: 10.52202/085713-2003. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/566aad4fac1acf17fd1ae8c3aef75326-Paper-Conference.pdf.

Luca Weihs, Unnat Jain, Iou-Jen Liu, Jordi Salvador, Svetlana Lazebnik, Aniruddha Kembhavi, and Alex Schwing. Bridging the imitation gap by adaptive insubordination. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 19134–19146. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/9fc664916bce863561527f06a96f5ff3-Paper.pdf.

Zehua Xia, Yuyang Wu, Yiyun Xia, and Cam Tu Nguyen. Momentum posterior regularization for multi-hop dense retrieval. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 8255–8271, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.550/.

Wenhan Xiong, Xiang Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Scott Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oguz. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=EMHoBG0avc1.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, and Tong Xu. From single to

multi-granularity: Toward long-term memory association and selection of conversational agents. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 91581–91617, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/93e444b4a1a64815631c3c74977ee200-Paper-Conference.pdf.

Ran Xu, Wenqi Shi, Yuchen Zhuang, Yue Yu, Joyce C. Ho, Haoyu Wang, and Carl Yang. Collab-RAG: Boosting retrieval-augmented generation for complex question answering via white-box and black-box LLM collaboration. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=CODs4jSGhN.

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 Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R 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.

Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pp. 829–838, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380379. doi: 10.1145/3404835.3462856. URL https://doi.org/10.1145/3404835.3462856.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Liu Yong, and Shen Huang. End-to-end beam retrieval for multi-hop question answering. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1718–1731, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.96. URL https://aclanthology.org/2024.naacl-long.96/.

Chen Zhao, Chenyan Xiong, Jordan Boyd-Graber, and Hal Daumé III. Distantly-supervised dense retrieval enables open-domain question answering without evidence annotation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9612–9622, Online and Punta Cana, Dominican Republic, November 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.756. URL https://aclanthology.org/2021.emnlp-main.756/.

Chen Zhao, Chenyan Xiong, Jordan Boyd-Graber, and Hal Daumé III. Multi-step reasoning over unstructured text with beam dense retrieval. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4635–4641, Online, June 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.368. URL https://aclanthology.org/2021.naacl-main.368/.

Yijia Zheng and Marcel Worring. Latentrag: Latent reasoning and retrieval for efficient agentic rag, 2026. URL https://arxiv.org/abs/2605.06285.

Yuqi Zhou, Sunhao Dai, Changle Qu, Liang Pang, Jun Xu, and Ji-Rong Wen. Learning to retrieve from agent trajectories. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '26, pp. 2544–2555, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400725999. doi: 10.1145/3805712.3809675. URL https://doi.org/10.1145/3805712.3809675.

Table 4: Training data. LI-RAG uses the decomposition traces of 1,000-question shards of the official training splits; the GRITHopper-7B* column is as reported by Erker et al. (2026).

	LI-RAG MuSiQue traces	LI-RAG MuSiQue + HotpotQA traces	GRITHopper-7B*
Training sets	MuSiQue	MuSiQue, HotpotQA	MuSiQue, HotpotQA, 2Wiki, HoVer, EX-FEVER
Questions or claims	4,000	7,000	184,070
one-hop	0	0	5,034
two-hop	1,600	4,600	127,846
three-hop	1,600	1,600	15,328
four-hop	800	800	35,862
Gold evidence used	no	no	yes

Ziyuan Zhuang, Zhiyang Zhang, Sitao Cheng, Fangkai Yang, Jia Liu, Shujian Huang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. EfficientRAG: Efficient retriever for multi-hop question answering. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3392–3411, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.199. URL https://aclanthology.org/2024.emnlp-main.199/.

A Experimental details

Implementation. The adapter of section 3.3 has rank 16, $\alpha = 32$ (a scaling factor of 2) and dropout 0.05 on the query, key, value and output projections of every decoder layer. It is trained with AdamW at a learning rate of 10^{-4} without weight decay, with a linear warm-up over the first 5% of the steps followed by cosine decay, a batch of 16 examples, gradient clipping at norm 1, three epochs and bfloat16 arithmetic; the number of distractors shown for each earlier round is redrawn every time an example is visited. Inputs to the adapted encoder are truncated at 2,048 tokens. The passage vectors are precomputed in float16 and the index is never rebuilt. On one H100 an epoch over the 23,000 examples of the three-dataset mix takes about 60 minutes, and an epoch of the GritLM-7B run of Table 3 about 29 minutes. At inference the loop runs three rounds of three with the pool of ten per round, batched over questions. The planner and solver of the decomposition loop answer under a JSON schema enforced by constrained decoding, so the solver's indices arrive as integers and the planner's admissible actions are those of section 3.2. The traces of the 4,000 MuSiQue questions comprise 10,886 rounds, and those of the MuSiQue + HotpotQA model 16,182; every trace is kept whether or not the teacher answered correctly, and no training run uses NQ, PopQA or MultiHop-RAG. Model selection scores every epoch, the untrained epoch included, by the answer accuracy of the inference loop on the development questions; the MuSiQue-traces model of the main tables is its second epoch.

Baselines. HippoRAG 2 is run from its released code and uses the same LLM for its open information extraction and recognition-memory filter and the same embedding model as LI-RAG, at its default retrieval settings, and its ranking is read at nine passages rather than its default five. GRITHopper is run for a fixed three hop rounds, appending three passages per hop as LI-RAG does, and its reader, like every other, reads the top nine of the pooled ranking. No training code for GRITHopper was released, so the retrained model follows the paper and the released training arguments: the two stages with distractor and mined negatives, the contrastive and generative objectives, the annotated supporting paragraphs of MuSiQue as positives, five epochs, and the checkpoint chosen by development hit rate; the contrastive and generative batches are 256 and 32 in place of 2,048 and 256, in the paper's ratio, for this data size. TreeHop is retrained with its released architecture, loss and optimizer on NV-Embed-v2 embeddings of the 4,000 MuSiQue questions, each question a tree of its annotated supporting paragraphs ordered by the annotated decomposition, with negatives drawn from the passages of the training shards, at a learning rate of 2×10^{-5} for eight epochs; the fifth is selected by its own recall statistic on the development questions. The output projections of

its gates are initialized at zero, so that before training every generated query equals the question embedding; from its default initialization, recall at nine passages on the development questions stayed below that of single-round retrieval. It runs for three hops of three passages with its redundancy and layer-wise pruning, and its pooled ranking fuses the passages its hops retain by best rank and continues with its first-hop ranking. L-RAG releases no checkpoint, so we train one model on each of MuSiQue, HotpotQA and 2Wiki with its released code and the gold documents of the training split, 12,288 two-hop questions for MuSiQue. It reads the question and two first-hop Contriever passages with the LLM of its released recipe and retrieves eight second-hop passages, and its reader receives these passages, up to ten, rather than the top nine of a pooled ranking. The answer accuracies of Table 1 are those of the model trained on MuSiQue on every benchmark, and the retrieval scores of Table 2 are those of the in-domain model on MuSiQue, HotpotQA and 2Wiki, on the pooled ranking of the top ten passages of both hops. Q-RAG is run from its three released checkpoints for nine steps, with the gold-based stop of its released evaluation removed, and the best checkpoint is reported on each benchmark; on the four multi-hop benchmarks this is the GTE checkpoint, which its released configuration trains on the full HotpotQA and MuSiQue training splits, 110,385 questions in all (figure 3); its retrieval scores in Table 2 on MuSiQue, HotpotQA and 2Wiki are read from the same checkpoint at six steps. IRCoT is run with its released controller, completion prompt and demonstrations, with the MuSiQue demonstrations on MultiHop-RAG, NQ and PopQA, for which it provides none, packed to 8,000 GPT-2 tokens, with greedy reasoning calls of at most 300 tokens that stop at a newline, its caps of ten reasoning sentences and fifteen paragraphs, and six passages per retrieval step. In the reasoning calls the newline and thinking-block tokens of Qwen3.5-35B-A3B are suppressed, so that the model continues the reasoning line rather than opening a thinking block. Its pooled ranking fuses the top ten passages of every retrieval step.

Reader, judge and metrics. The reader and the judge are both Qwen3.5-35B-A3B, served with vLLM (Kwon et al., 2023) with its thinking mode disabled and a context length of 16,384 tokens. Both decode greedily (temperature 0), with at most 512 generated tokens for the reader and 32 for the judge. The reader receives the one-shot chain-of-thought prompt of HippoRAG 2 (figure 4), with each retrieved passage given as its title and text, and the answer is the text after *Answer:*, or the whole reply if it contains none. The judge receives the prompt of MemGAS (figure 5) with the question, one reference answer and the reader's answer; when a question has several acceptable answers, the reference shown is the one with the highest token-level F1 with that answer, and a verdict is counted as correct only if it contains `[[yes]]` and not `[[no]]`. Exact match and string recall are computed as well and not reported: the reader answers in phrases rather than in the gold string, so exact match is low for every method and separates them less than the judge does. In the pooled ranking, ties between rounds are resolved toward the earlier round. HippoRAG 2 is scored on its own ranking, and TreeHop, which records no candidates beyond the passages its hops retain, on the pooled ranking described above.

Embedding geometry. Figure 2 compares LI-RAG, the NV-Embed-v2 model trained on MuSiQue traces (Table 1), with the untrained encoder on the MuSiQue test set. The sub-questions q_1 and q_2 are the two steps of the annotated MuSiQue decomposition of each question, with the answer of the first step substituted into the second, embedded under the retrieval instruction; for 246 of the 518 two-hop questions q_1 is in the relation form of MuSiQue (for example, `Autumn Leaves >> creator`) and is embedded as written. In panel (a), q_2 is *What date did John Everett Millais die?*, d_1 and d_2 are *Autumn Leaves (painting)* and *John Everett Millais*, and the other points are the passages the two loops retrieved in rounds 1 and 2. The map is computed by metric multidimensional scaling of cosine distances, following the per-hop visualization of Zhao et al. (2021b), and is approximate (Spearman correlation 0.90 between distances on the map and cosine distances), and the rank of d_2 beside each query vector is computed among all 11,656 passages in the full embedding space.

B Results by number of hops

Table 5 breaks the MuSiQue results of section 4.4 down by the number of hops.

Table 5: MuSiQue by number of hops (518, 316 and 166 questions): R@10 of the pooled ranking and answer accuracy (Acc.) at nine passages.

	2-hop		3-hop		4-hop	
Method	R@10	Acc.	R@10	Acc.	R@10	Acc.
HippoRAG 2	0.929	65.8	0.815	49.4	0.602	17.5
Single-round dense (NV-Embed-v2, top 9)	0.843	56.0	0.783	48.1	0.568	19.3
GRITHopper	0.918	64.3	0.875	53.8	0.578	12.1
LI-RAG (MuSiQue traces)	**0.966**	**68.2**	**0.903**	**59.5**	**0.682**	**26.5**

Table 6: Answer accuracy of training variants of LI-RAG at nine passages. The 10,000-question run trains on 6,000 MuSiQue and 4,000 2Wiki questions; the doubly indented row modifies the run above it, and the full fine-tunes train every decoder weight on the MuSiQue traces.

	Multi-hop				Single-hop	
Training	MuSiQue	HotpotQA	2Wiki	MultiHop-RAG	NQ	PopQA
Single-round dense (NV-Embed-v2)	47.4	84.0	59.3	72.3	71.0	72.2
LI-RAG, MuSiQue traces (4,000)	**58.5**	86.1	**82.6**	77.3	**72.1**	72.4
+ HotpotQA traces (7,000)	58.2	85.9	78.8	77.0	71.2	72.4
MuSiQue + 2Wiki traces (10,000)	58.4	**86.5**	77.1	**78.1**	71.6	72.7
adapter rank 64	56.6	86.3	77.8	76.5	71.7	**73.8**
full fine-tune, lr 2×10^{-6}	55.6	85.8	76.9	75.9	71.6	72.5
full fine-tune, lr 10^{-5}	47.6	84.5	72.5	72.2	69.4	71.5

C Training ablations

Table 6 gives the answer accuracy of the training variants of LI-RAG discussed in section 4.4. Table 7 gives the retrieval of their pooled rankings and of three further variants whose answer accuracy at nine passages was not measured: the untrained loop, a run on MuSiQue, HotpotQA and 2Wiki traces, and the same run with the sub-question loss alone.

D Early stopping

The number of rounds R of section 3.5 is a cap rather than a fixed length. Each round's forward pass also records two training-free signals, the cosine between $\boldsymbol{e}_i$ and $\boldsymbol{e}_{i-1}$ and the number of passages in $\mathbb{R}_i$ that were already shown to the model, from which a rule for leaving the loop early can be derived; the second is the redundancy signal TreeHop (Li et al., 2026) prunes on, used here as a per-question exit rather than to prune branches. Since round i depends only on rounds 1 to $i-1$, a run recorded at the cap contains every shorter run of the same questions, so such a rule can be evaluated by replaying the record without the model. Replaying the MuSiQue + HotpotQA model with the overlap stop, which exits on the second signal, under a cap of five rounds leaves the accuracy of its evidence set within noise of three fixed rounds on every Wikipedia benchmark while keeping 1.4 to 2.7 rounds per question (Table 8). The round that triggers the exit is computed and then discarded, so a question that exits before the cap costs one forward pass more than the rounds it keeps.

E Reader and judge prompts

Figure 4 gives the reader prompt and figure 5 the judge prompt, both used unchanged on every benchmark; bold labels mark the chat roles and italic text in angle brackets the fields filled in per question.

Table 7: R@10 (R) and FCR@10 (FCR) on the pooled ranking of the training variants of Table 6 and of three further variants. The run on HotpotQA and 2Wiki traces adds 3,000 questions of each to the 4,000 MuSiQue questions, and the doubly indented rows modify the run above them.

	MuSiQue		HotpotQA		2Wiki		MultiHop-RAG	
Training	R	FCR	R	FCR	R	FCR	R	FCR
Single-round dense (NV-Embed-v2)	0.778	0.504	0.974	0.949	0.806	0.578	0.592	0.286
Untrained loop	0.804	0.565	0.987	0.975	0.880	0.701	0.559	0.243
LI-RAG, MuSiQue traces (4,000)	0.899	0.740	**0.994**	0.989	**0.990**	**0.970**	0.604	0.314
+ HotpotQA traces (7,000)	0.892	0.726	0.992	0.986	0.971	0.898	0.612	0.328
+ HotpotQA + 2Wiki traces (10,000)	0.895	0.734	0.993	0.987	0.953	0.815	0.635	0.350
sub-question loss only	0.891	0.728	0.989	0.979	0.949	0.807	0.588	0.297
MuSiQue + 2Wiki traces (10,000)	**0.904**	**0.756**	**0.994**	**0.990**	0.960	0.844	**0.638**	**0.352**
adapter rank 64	0.885	0.716	0.990	0.982	0.960	0.845	0.603	0.309
full fine-tune, lr 2×10^{-6}	0.862	0.672	0.989	0.980	0.963	0.877	0.568	0.272
full fine-tune, lr 10^{-5}	0.787	0.567	0.968	0.940	0.935	0.810	0.525	0.228

Table 8: Early stopping of the MuSiQue + HotpotQA model, replayed from one five-round record per benchmark (1,000 questions each). Acc. is the answer accuracy when the reader receives the evidence set $\mathbb{M}$ alone rather than the top nine passages of the pooled ranking; $|\mathbb{M}|$ and Rounds are means per question.

	Three rounds		Five rounds		Overlap stop, cap five		
Benchmark	$\lvert\mathbb{M}\rvert$	Acc.	$\lvert\mathbb{M}\rvert$	Acc.	Rounds	$\lvert\mathbb{M}\rvert$	Acc.
MuSiQue	6.1	57.3	6.4	56.3	2.7	6.4	56.9
HotpotQA	4.5	84.9	4.5	84.8	1.9	4.5	85.6
2Wiki	5.6	79.5	5.7	80.6	2.4	5.7	79.7
NQ	3.4	69.9	3.4	70.5	1.4	3.4	70.6
PopQA	3.7	72.0	3.7	71.5	1.6	3.7	72.6

```
[system]
As an advanced reading comprehension assistant, your task is to analyze text passages and
  corresponding questions meticulously. Your response start after "Thought: ", where you
  will methodically break down the reasoning process, illustrating how you arrive at
  conclusions. Conclude with "Answer: " to present a concise, definitive response, devoid of
  additional elaborations.
[user]
Wikipedia Title: The Last Horse
The Last Horse (Spanish:El último caballo) is a 1950 Spanish comedy film directed by Edgar
  Neville starring Fernando Fernán Gómez.
Wikipedia Title: Southampton
The University of Southampton, which was founded in 1862 and received its Royal Charter as a
  university in 1952, has over 22,000 students. The university is ranked in the top 100
  research universities in the world in the Academic Ranking of World Universities 2010. In
  2010, the THES - QS World University Rankings positioned the University of Southampton in
  the top 80 universities in the world. The university considers itself one of the top 5
  research universities in the UK. The university has a global reputation for research into
  engineering sciences, oceanography, chemistry, cancer sciences, sound and vibration
  research, computer science and electronics, optoelectronics and textile conservation at
  the Textile Conservation Centre (which is due to close in October 2009.) It is also home
  to the National Oceanography Centre, Southampton (NOCS), the focus of Natural Environment
  Research Council-funded marine research.
Wikipedia Title: Stanton Township, Champaign County, Illinois
Stanton Township is a township in Champaign County, Illinois, USA. As of the 2010 census, its
  population was 505 and it contained 202 housing units.
Wikipedia Title: Neville A. Stanton
Neville A. Stanton is a British Professor of Human Factors and Ergonomics at the University
  of Southampton. Prof Stanton is a Chartered Engineer (C.Eng), Chartered Psychologist
  (C.Psychol) and Chartered Ergonomist (C.ErgHF). He has written and edited over a forty
  books and over three hundered peer-reviewed journal papers on applications of the subject.
  Stanton is a Fellow of the British Psychological Society, a Fellow of The Institute of
  Ergonomics and Human Factors and a member of the Institution of Engineering and
  Technology. He has been published in academic journals including "Nature". He has also
  helped organisations design new human-machine interfaces, such as the Adaptive Cruise
  Control system for Jaguar Cars.
Wikipedia Title: Finding Nemo
Finding Nemo Theatrical release poster Directed by Andrew Stanton Produced by Graham Walters
  Screenplay by Andrew Stanton Bob Peterson David Reynolds Story by Andrew Stanton Starring
  Albert Brooks Ellen DeGeneres Alexander Gould Willem Dafoe Music by Thomas Newman
  Cinematography Sharon Calahan Jeremy Lasky Edited by David Ian Salter Production company
  Walt Disney Pictures Pixar Animation Studios Distributed by Buena Vista Pictures
  Distribution Release date May 30, 2003 (2003 - 05 - 30) Running time 100 minutes Country
  United States Language English Budget $$94 million Box office $$940.3 million

Question: When was Neville A. Stanton's employer founded?
Thought:
[assistant]
The employer of Neville A. Stanton is University of Southampton. The University of
  Southampton was founded in 1862.
Answer: 1862.
[user]
Wikipedia Title: <title of passage 1>
<text of passage 1>

Wikipedia Title: <title of passage 2>
<text of passage 2>

...

Question: <question>
Thought:
```

Figure 4: Reader prompt, the one-shot chain-of-thought prompt of HippoRAG 2 (Gutiérrez et al., 2025), as chat messages. The last user message lists the passages to be read, followed by the question.

[user]
I will give you a question, a reference answer, and a response from a model. Please answer [[yes]] if the response contains the reference answer. Otherwise, answer [[no]].
If the response is equivalent to the correct answer or contains all the intermediate steps to get the reference answer, you should also answer [[yes]]. If the response only contains a subset of the information required by the answer, answer [[no]].

[User Question]
<question>

[The Start of Reference Answer]
<reference answer>
[The End of Reference Answer]

[The Start of Model's Response]
<reader's answer>
[The End of Model's Response]

Is the model response correct? Answer [[yes]] or [[no]] only.

Figure 5: Judge prompt of MemGAS (Xu et al., 2026), sent as a single user message.