S^{2}LMR: SUPERVISED SEQUENTIAL LONG-CONTEXT MULTI-STEP RETRIEVER

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-step retrieval over long contexts is a key capability for retrieval-augmented generation, but strong existing approaches are costly and complex to train. Supervised retrievers are far simpler and cheaper, yet they fall well behind in this setting. We show that this gap stems not from supervised training itself, but from three misalignments between training and inference: easy training negatives, noisy retrieval prefixes, and position-agnostic scoring. Guided by this analysis, we introduce S^{2}LMR, a simple yet effective supervised sequential retriever that models multi-step retrieval as the evolution of an explicit textual state encoded by a single shared Transformer. On long-context reasoning, multi-hop QA, and needle-in-a-haystack benchmarks with contexts from 4K to 10M tokens, S^{2}LMR matches or outperforms state-of-the-art methods and far surpasses prior supervised retrievers, while training over 20$\times$ faster and remaining stable across a wide range of hyperparameters.

1 INTRODUCTION

Retrieval-Augmented Generation (RAG) has become a standard paradigm for augmenting large language models (LLMs) with external knowledge, improving factuality without requiring large knowledge sources to be directly included in the model context (Lewis et al., 2020; Guu et al., 2020; Izacard & Grave, 2021; Gao et al., 2023). Most existing RAG studies focus on single-step retrieval, which is often sufficient for relatively simple tasks such as Needle-in-a-Haystack (Hsieh et al., 2024). However, single-step retrieval is insufficient for emerging scenarios that require multiple rounds of interaction with the context, such as multi-hop question answering (Yang et al., 2018; Trivedi et al., 2022) and reasoning over millions of tokens (Kuratov et al., 2024). These settings require the retriever to progressively gather relevant evidence across multiple steps.

There are several types of approaches for multi-step retrieval. Graph-based approaches build structured graphs over retrieved information to connect evidence across documents, but often incur additional overhead from graph construction and traversal (Li et al., 2024; Mao et al., 2021). Some works use LLM agents to guide multi-step retrieval through generated reasoning or instructions (Anokhin et al., 2024; Singh et al., 2025). However, repeated LLM calls increase inference cost, while noisy evidence can misguide later retrieval steps. Another line of work fine-tunes LLMs to interact with retrieval tools (Jin et al., 2025; Song et al., 2025), but fine-tuning LLMs can be expensive and difficult to scale.

More recently, Sorokin et al. (2026) proposed a more resource-efficient approach, Q-RAG, that uses reinforcement learning (RL) to fine-tune the embedder model, achieving state-of-the-art results on multiple long-context multi-step benchmarks with contexts of up to 10 million tokens. However, compared with conventional supervised retrieval using bi-encoders or cross-encoders (Xiong et al., 2020; Zhang et al., 2024), this approach introduces additional RL components, such as value and target networks, as well as temporal-difference updates, increasing implementation complexity and introducing additional hyperparameters. This raises an intriguing but non-trivial question:

Can we achieve effective multi-step retrieval over million-token contexts using only a simple supervised learning framework?

Through extensive analysis, we identify three misalignments that explain why supervised retrievers fall far behind recent methods such as Q-RAG: (i) training negatives are easy, unrelated passages,

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.

whereas test-time distractors come from the same context as the gold passage; (ii) later hops are trained on the model's own, sometimes wrong, retrieval prefixes, which injects noise into their supervision; and (iii) the scorer ignores where a candidate lies relative to the passages already retrieved, even though this position strongly predicts the next relevant passage.

Motivated by these observations, we propose the **S**upervised **S**equential **L**ong-Context **M**ulti-Step **R**etriever (S^2LMR), a supervised bi-encoder designed to align training with long-context multi-step retrieval. S^2LMR addresses the identified mismatches through same-context negatives, gold-prefix next-evidence supervision, and history-adaptive relative positions, using a single shared Transformer encoder with an explicit *textual state* that evolves across retrieval hops. Across (ultra)-long context benchmarks and multi-hop QA, S^2LMR matches or outperforms the state-of-the-art RL-based retriever Q-RAG (Sorokin et al., 2026). Our main contributions are as follows:

- We propose S^2LMR, a supervised sequential retriever built on a single shared Transformer encoder and a simple contrastive supervision.
- On long-context commonsense reasoning, multi-hop QA, and needle-in-a-haystack tasks with contexts from 4K to 10M tokens, S^2LMR matches or outperforms the state-of-the-art Q-RAG. It also substantially outperforms Beam Retriever, the strongest existing supervised retriever. On BabiLong QA3, S^2LMR keeps fact EM above 0.95 at up to 10M tokens, while Beam Retriever reaches only 0.2–0.3.
- We further show that S^2LMR offers substantial advantages in training efficiency and stability. For example, on BabiLong QA3 at 10M tokens, S^2LMR reaches its peak performance in only 2.25 GPU hours, compared with 54.4 GPU hours for Q-RAG and more than 400 GPU hours for Beam Retriever. Moreover, compared with Q-RAG, S^2LMR exhibits more stable training dynamics and is substantially more robust to a wide range of hyperparameter choices, including learning rate and batch size.

2 Related Works

Multi-step retrieval. Multi-hop retrieval methods (Das et al., 2019; Qi et al., 2019) extend single-step dense retrieval (Karpukhin et al., 2020; Izacard et al., 2021) by conditioning each retrieval decision on previously retrieved evidence. MDR (Xiong et al., 2020) uses a bi-encoder and iteratively augments the query with retrieved passages, while BeamRetriever (Zhang et al., 2024) combines a cross-encoder with beam search for stronger multi-hop retrieval at higher inference cost. Q-RAG (Sorokin et al., 2026) further studies embedder-based multi-step retrieval in long-context settings. Another line of work uses LLM agents to interleave retrieval with generated reasoning (Yao et al., 2022; Trivedi et al., 2023; Shao et al., 2023; Asai et al., 2024; Singh et al., 2025; Jin et al., 2025; Song et al., 2025). These methods provide flexible reasoning but require repeated LLM inference across retrieval steps. S^2LMR focuses instead on more efficient sequential retrieval with a lightweight shared encoder.

Retrieval for long-context tasks. Long-context benchmarks such as RULER (Hsieh et al., 2024) and BabiLong (Kuratov et al., 2024) evaluate reasoning over contexts ranging from thousands to millions of tokens. Existing approaches improve long-context processing through architectural changes, including recurrent memory (Bulatov et al., 2022; Behrouz et al., 2026; 2025; Rodkin et al., 2024), state-space models (Gu & Dao, 2023), and positional encoding adaptation (Shang et al., 2025). An alternative is to first retrieve a small subset of relevant content from the long context (Jiang et al., 2024a;b). Q-RAG (Sorokin et al., 2026) follows this direction and shows that learned chunk embeddings can support multi-step retrieval over very long contexts. S^2LMR builds on this retrieval-based paradigm while introducing history-adaptive retrieval states and candidate positions for simple supervised multi-step retrieval.

Supervised training for dense retrieval. Dense retrievers are commonly trained with contrastive objectives (Oord et al., 2018; Karpukhin et al., 2020). DPR (Karpukhin et al., 2020) trains a bi-encoder using in-batch negatives together with hard negatives from term-based retrieval, and multi-hop methods such as MDR (Xiong et al., 2020) largely follow the same training strategy. Such objectives were originally developed for open-domain retrieval, where negatives often come from other queries or documents. In long-context multi-step retrieval, however, the main competitors at inference are chunks from the same context, and the correct target can also depend on previously

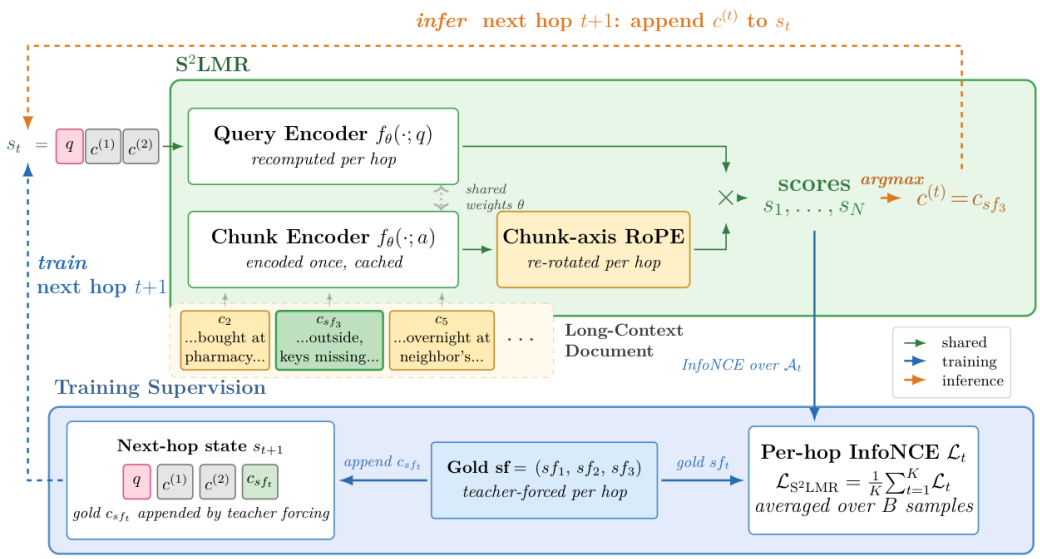


Figure 1: Overview of S^{2}LMR, which performs sequential multi-hop retrieval with a shared bi-encoder. The initial state s_0 contains the query q, and all chunks of the long context are embedded and cached at the start. At each hop t, S^{2}LMR embeds the current state s_t, re-rotates cached chunk embeddings according to the retrieval history, and scores candidate chunks by their inner products with the state embedding. During training (blue), gold chunks provide per-hop supervision; during inference (orange), S^{2}LMR greedily selects and appends the highest-scoring chunk.

retrieved evidence. Q-RAG (Sorokin et al., 2026) instead formulates retrieval as a Markov decision process and trains with temporal-difference learning (Mnih et al., 2015). In contrast, S^{2}LMR retains simple supervised contrastive training while aligning the candidate set, retrieval state, and relative positions with those encountered during inference.

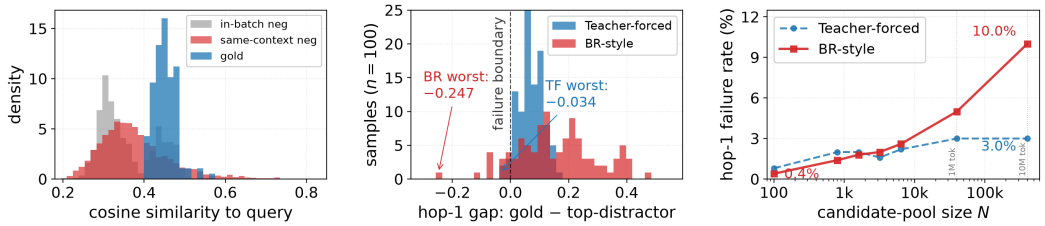
3 Methods

3.1 Revisiting Supervised RAG for Long-Context Multi-Step Retrieval

Supervised retrievers remain attractive for multi-step retrieval because they are efficient and can be trained directly from retrieval labels. Two representative approaches are MDR (Xiong et al., 2020) and Beam Retriever (BR) (Zhang et al., 2024). MDR performs sequential dense retrieval with a bi-encoder, while BR scores each new candidate together with the question and previously retrieved passages. These methods were mainly developed for open-domain or relatively small candidate sets. Long-context retrieval is different: the model must search over many passages from the same context, where distractors can be highly similar to the correct evidence.

To test how well existing designs transfer to this setting, we first train a bi-encoder with the standard DPR/MDR recipe on the challenging long-context BabiLong QA3 benchmark (Kuratov et al., 2024). It achieves only 1.6% fact exact match (fact-EM), substantially below recent multi-step retrievers. We identify three issues that become especially important as the context and candidate pool grow.

Issue 1: Training negatives do not match hard same-context distractors. MDR mainly learns from negatives associated with other queries, which are often easier than distractors from the same context. In long-context retrieval, however, incorrect passages may mention the same entities, describe related events, and use very similar language. BR uses candidates from the same example, but its original training setting contains only 10–20 candidates, far fewer than the thousands of chunks that may appear in long contexts. Existing training therefore provides limited supervision for distinguishing the gold passage from many highly similar distractors. Figure 2a illustrates this problem: on our 128 evaluation examples, an MDR-style model ranks at least one same-context distractor above the correct first-hop passage in every case.



(a) (b) (c)

Figure 2: Evidence for Issues 1 and 2. (a) Hop-1 cosine distributions under MDR-style training: gold vs. same-context and in-batch negatives. (b) Distribution of the hop-1 gold-vs-distractor cosine gap at ns=6.4k. (c) Hop-1 failure rate as the candidate pool grows to ∼10M tokens.

Table 1: Per-hop retrieval failure of a fine-tuned chunk-position-agnostic bi-encoder on BabiLong QA3 at N=100. $\text{margin}_t = \cos(q_t, \text{gold})_{\max} - \cos(q_t, \text{distractor})_{\max}$. See Appendix C.3.

Metric	hop-1	hop-2	hop-3	K=3 upper bound
$P(\text{margin} \le 0)$	0.66	0.53	0.80	—
Per-hop pick success	0.34	0.47	0.20	**3.1%**

Issue 2: Predicted prefixes make multi-hop supervision noisy. BR trains later hops using passages selected by earlier hops. If an earlier prediction is wrong, later hops are trained on an incorrect retrieval history. This makes the supervision noisier and can reduce the score gap between the gold passage and strong distractors. On 100 held-out QA3 examples with 400k candidates each (∼10M tokens), BR-style training gives a **negative** gold-vs-distractor margin on 12% of examples, with a worst margin of −0.247, while a *matched model trained* with ground-truth prefixes flips on only 4% of examples, with a worst value of −0.034 (Fig. 2b). The problem becomes more serious as the candidate pool grows: from 100 to 400k passages (∼10M tokens), the hop-1 failure rate rises from 0.4% to 10% and end-to-end EM drops from 0.994 to 0.880, while the matched control remains below 3% (Fig. 2c).

Issue 3: Retrieval scores ignore the chunk position. Neither MDR nor BR explicitly uses where a candidate appears in the document. This loses useful information in long-context tasks, where relevant facts are often positionally related. On BabiLong QA3, the next relevant fact lies within 25% of the previous pick's position in 90% of examples, with a median offset of only 5%. For example, after retrieving a fact at chunk 48 for the question "Where was the apple before the bathroom?", the word "before" suggests that the next relevant fact should appear earlier in the document. A position-agnostic retriever cannot directly use this signal. Table 1 shows the consequence: even after fine-tuning, the model often assigns similar scores to the gold passage and its strongest distractor, and per-hop success remains only 20%–47%, limiting three-hop retrieval EM to at most 3.1%.

3.2 Supervised Sequential Long-Context Multi-Step Retriever

Motivated by Sec. 3.1, we propose S^{2}LMR, a supervised sequential retriever for long-context multi-step retrieval. It aligns training with inference along three complementary dimensions: the candidate set used for contrastive supervision, the sequential retrieval state conditioned on previous evidence, and the history-adaptive relative positions used to score candidates.

S^{2}LMR uses a shared encoder $f_\theta : \mathcal{X} \to \mathbb{R}^d$ to represent both retrieval states and candidate chunks. Given a text input x, we mean-pool its final-layer hidden states over non-padding tokens and apply ℓ_2 normalization to obtain $f_\theta(x)$. Each example consists of a query q and a context divided into N chunks $c_1, \ldots, c_N$, indexed by $\mathcal{C} = \{1, \ldots, N\}$.

Sequential retrieval state. S^{2}LMR retrieves one chunk per hop, and each hop is conditioned on the chunks retrieved before it. Let $c^{(t)} = c_{i_t}$ be the chunk retrieved at hop t, where i_t is its document index, and let π be the permutation of $\{1, \ldots, t-1\}$ with $i_{\pi_1} < \cdots < i_{\pi_{t-1}}$. The state at hop t is

$$s_t = \left[q,\ c^{(\pi_1)},\ \ldots,\ c^{(\pi_{t-1})} \right], \tag{1}$$

with $s_1 = q$, where $[\cdot]$ denotes concatenation with a fixed separator token such as `[SEP]`. The permutation only determines how the retrieved chunks are arranged in the input. Chunks are still retrieved one at a time, but the encoder always reads them in document order.

During training, S^{2}LMR uses the gold retrieval prefix $c^{(1)}, \ldots, c^{(t-1)} = (c_{i_1^\star}, \ldots, c_{i_{t-1}^\star})$ to construct s_t. Rather than conditioning later-hop supervision on noisy model-generated retrieval prefixes, S^{2}LMR therefore reformulates multi-hop retrieval as a sequence of conditionally supervised retrieval decisions, where each hop directly supervises the next evidence-selection decision given the preceding gold evidence. Fig. 3b shows the effect. The performance gap between beam-expanded supervision and ours widens as context length increases.

The gold indices $(i_1^\star, \ldots, i_K^\star)$ follow the annotated reasoning chain when available (MuSiQue, BabiLong) and document order otherwise (HotpotQA). Under this ordering, each hop has a single designated target, while later gold chunks remain candidates rather than being simultaneously treated as positive. This explicitly trains the model to distinguish which evidence should be retrieved next, rather than only which evidence belongs to the final support set.

History-adaptive candidate positions. A candidate's position should describe where it lies relative to the evidence retrieved so far, not where it lies in the document. Let $i_{(1)} < \cdots < i_{(t-1)}$ be the sorted document indices of the chunks retrieved before hop t, with $i_{(0)} = 0$ and $i_{(t)} = N + 1$. These indices divide the remaining candidates into t segments $S_k = \{\, j \mid i_{(k)} < j < i_{(k+1)} \,\}$ for $k = 0, \ldots, t-1$, one for each gap between consecutive retrieved chunks. For $j \in S_k$ we set

$$p_j^{(t)} = k\Delta + 0.9\,\Delta\,\frac{j - i_{(k)}}{i_{(k+1)} - i_{(k)}}, \tag{2}$$

with a fixed step size $\Delta > 0$. Segment k occupies $[k\Delta, (k + 0.9)\Delta)$, and the remaining 0.1Δ separates it from segment $k+1$. Two candidates on opposite sides of a retrieved chunk are therefore assigned clearly different positions, even if they are adjacent in the document.

The relative chunk axis is taken similarly to that of Q-RAG (Sorokin et al., 2026). The main difference is that S^{2}LMR recomputes the positions after every hop from the current history, using gold chunks during training and retrieved chunks at inference. This has two consequences. First, a position now tells the model which two pieces of retrieved evidence a candidate lies between and how far along that gap it sits, which is the information the next hop needs. Second, $p_j^{(t)} < t\Delta \le K\Delta$ regardless of document length, so the model never encounters a position at test time that it did not see during training. Absolute chunk indices do not have this property because they grow with N.

Fig. 3c compares three positional schemes: no positional encoding, static (absolute) RoPE, and our history-adaptive relative positions. Removing positional encoding entirely performs poorly across all context lengths. Static RoPE is competitive near the training length but degrades sharply beyond it, whereas ours remains accurate even at the longest context length we evaluated.

Full-context contrastive supervision. Given s_t and $p_j^{(t)}$, the score of candidate c_j is given by

$$\mathrm{score}(s_t, c_j) = \big\langle\, f_\theta(s_t),\ \mathrm{RoPE}\big(f_\theta(c_j),\, p_j^{(t)}\big)\,\big\rangle, \tag{3}$$

where RoPE (Su et al., 2024) is applied only to the candidate chunk embedding, while the query-state embedding $f_\theta(s_t)$ is kept at position zero. Since $p_j^{(t)}$ already measures the candidate's position relative to the retrieved history represented by s_t, this makes the score depend directly on the history-adaptive candidate position.

At training hop t, the candidate set contains every chunk not yet retrieved, $\mathcal{A}_t = \mathcal{C} \setminus \{i_1^\star, \ldots, i_{t-1}^\star\}$, and the loss is a softmax over this full set, averaged over hops:

$$\mathcal{L}_t = -\log \frac{\exp\big(\mathrm{score}(s_t, c_{i_t^\star})/\tau\big)}{\sum_{j \in \mathcal{A}_t} \exp\big(\mathrm{score}(s_t, c_j)/\tau\big)}, \qquad \mathcal{L}_{\mathrm{S^2LMR}} = \frac{1}{K}\sum_{t=1}^{K} \mathcal{L}_t, \tag{4}$$

with temperature τ. We use neither sampled nor in-batch negatives. The difficult negatives in multi-hop retrieval are chunks from the same context that mention the right entities but are not the next step in the chain, and they only enter the loss if the entire context is in the denominator. Fig. 3a shows the effect. Replacing same-context negatives with in-batch negatives lowers Fact EM from 0.99 to 0.28, and Table 6 shows that most of S^{2}LMR's margin over distractors comes from same-context negatives. Because the gold sequence fixes the training state at every hop, all K decisions receive a direct gradient, and no trajectory sampling or search is needed during training.

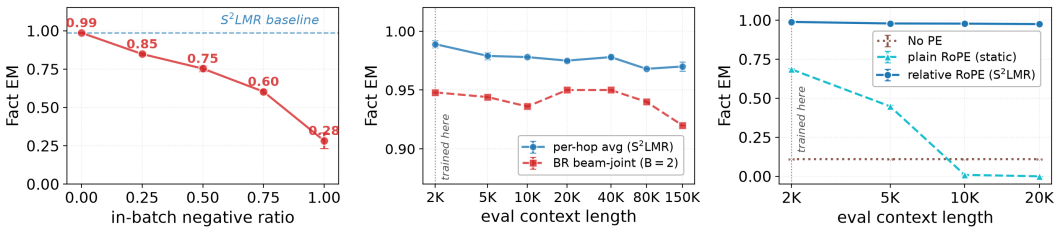


Figure 3: Verification of the three S^{2}LMR designs on BabiLong QA3, measured by Fact EM: **(a)** negative sampling with varying fractions of in-batch negatives; **(b)** teacher-forced versus beam-expanded per-hop supervision (B=2); and **(c)** re-anchored, static absolute, and no chunk-axis positional encoding. Vertical dashed lines indicate the training context length (2K).

Inference. At test time, the gold history is replaced by the model's own previous retrievals, and the state construction, position rule, and scoring function are unchanged. Chunk embeddings $\{f_\theta(c_j)\}_{j=1}^{N}$ are computed once and cached. At hop t, the candidate set is $\mathcal{A}_t = \mathcal{C} \setminus \{i_1, \ldots, i_{t-1}\}$. S^{2}LMR selects

$$i_t = \arg\max_{j \in \mathcal{A}_t} \text{score}(s_t, c_j),$$

appends c_{i_t} to the history, and recomputes the candidate positions. Algorithm 1 in the appendix gives the full procedure.

4 Experiments

4.1 Experimental Setup

We evaluate S^{2}LMR on both long-context retrieval and open-domain multi-hop retrieval, covering the primary settings of multi-step dense retrieval. Long-context evaluation is conducted on **RULER** (Hsieh et al., 2024), which contains synthetic retrieval tasks with context lengths ranging from 4K to 1M tokens, including both single- and multi-step retrieval, and on **BabiLong** (Kuratov et al., 2024), which evaluates multi-hop reasoning over contexts of up to millions of tokens. Open-domain evaluation is conducted on **HotpotQA** (Yang et al., 2018) and **MuSiQue** (Trivedi et al., 2022), with typical context lengths of 1–2K and 2–3K tokens, respectively.

Following prior work, retrieval quality on HotpotQA and MuSiQue is evaluated using Supporting Fact F1/EM together with final answer F1/EM. For RULER and BabiLong, we report Fact EM across retrieval subtasks and context lengths. Unless otherwise specified, we use identical retrieval budgets and decoding procedures where applicable. Detailed training hyperparameters are listed in Appendix D. In the result tables, ∘ marks *reported* baselines whose scores are taken directly from the original papers, while ✓ marks *reproduced* baselines that we fine-tuned and/or evaluated using released code or publicly available checkpoints.

4.2 Retrieval Performance

1) Ultra-long Context Reasoning. We compare S^{2}LMR with the state-of-the-art long-context retrieval methods including Q-RAG (Sorokin et al., 2026), Beam Retriever (Zhang et al., 2024), ARMT (Rodkin et al., 2024), RMT (Bulatov et al., 2022), Mamba2 (Dao & Gu, 2024), and other long-context language models and LLM-based agents (Achiam et al., 2023; Grattafiori et al., 2024) reported in prior work. Figure 4 reports average performance across all QA tasks. S^{2}LMR maintains strong performance across the entire context range, remaining above 95% even at 10M tokens and closely matching Q-RAG throughout. It also substantially outperforms existing long-context architectures, including recurrent transformers and state-space models.

We further examine QA3, the most challenging BabiLong task requiring three-step reasoning. As shown in Figure 4, S^{2}LMR again maintains strong performance as context length increases, closely matching Q-RAG across the full range. Together, these results show that a supervised bi-encoder can effectively support multi-step retrieval and reasoning even in extremely long contexts.

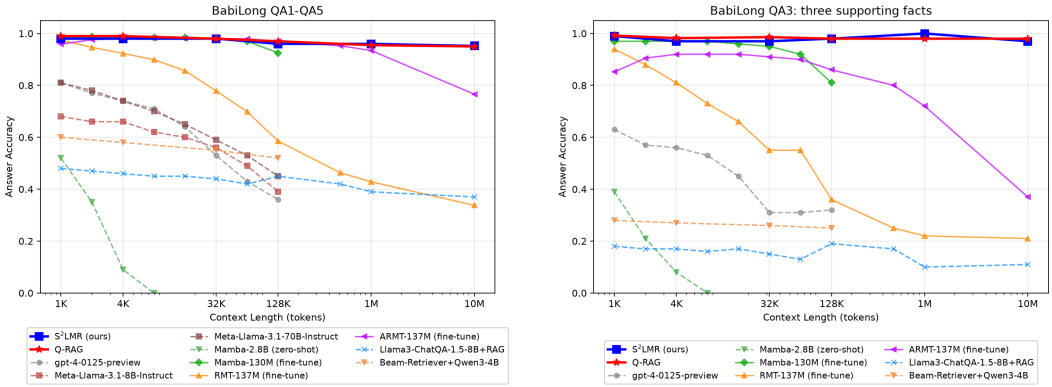


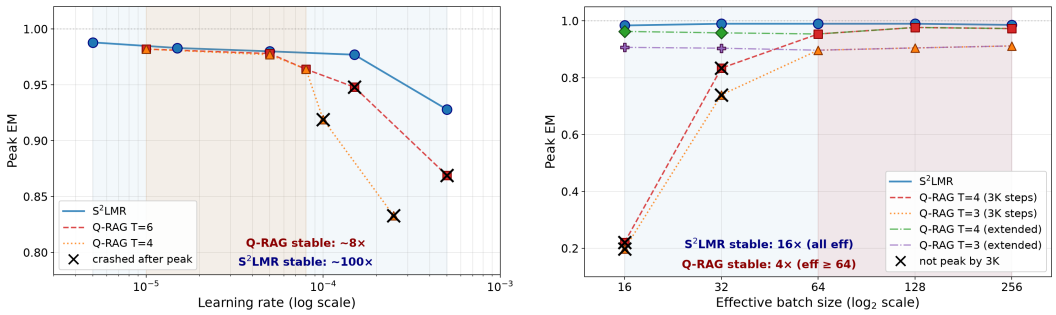
Figure 4: Comparison of answer accuracy on the long-context benchmark BabiLong. Solid lines denote methods fine-tuned on the BabiLong, while dashed lines denote zero-shot methods. a) Average performance across tasks Q1–QA5. b) Performance on the hardest task, QA3, which requires the longest reasoning chain and temporal awareness.

Table 2: Results on the RULER benchmark, evaluating long-context retrieval performance across various context lengths. **S** (Single-needle): Find one value for one key. **MK** (Multi-keys): Find one value for one key among many. **MV** (Multi-values): Find all values for one key. **MQ** (Multi-query): Answer multiple questions over the context. **MH QA**: open domain multi-hop question answering. **SH QA**: single-hop question answering.

		S			MK			MV	MQ	NIAH	QA	
Len	Methods	1-st	2-nd	3-rd	1-st	2-nd	3-rd			Avg.	SH	MH
4K	°Titans	98.4	99.8	89.4	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	°Atlas	99.2	100	90.6	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	°Mamba2-Hybrid	100	100	95.7	89.5	95.5	96	97.9	97.6	96.5	56.5	48.8
	°LongRoPE2-8B	100	100	99	100	100	97	99.7	100	99.7	**79**	60
	✓Beam-Retriever	100	98	98	98	98	97	98	99	98.5	29.0	39.0
	✓Q-RAG	100	100	100	100	100	100	100	100	100	55	**64**
	S²LMR (Ours)	100	100	100	100	100	100	100	100	100	56	**64**
16K	°Titans	96.2	80.2	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	°Atlas	97	84	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	°Mamba2-Hybrid	100	100	81.5	92	92.2	83	89.8	90.2	91.1	48.8	44
	°LongRoPE2-8B	100	100	99	99	100	98	95	98.2	98.8	**69**	58
	✓Beam-Retriever	100	100	97	96.5	96	95	80	98	95.3	24.0	35.0
	✓Q-RAG	100	100	100	100	100	100	100	100	100	53	60
	S²LMR (Ours)	100	100	100	100	100	100	100	100	100	51	**62**
32K	°Mamba2-Hybrid	100	100	96.7	84	76.5	81.5	84.3	80.9	88.0	41.8	38.5
	°LongRoPE2-8B	100	100	99	96	91	94	96.5	97	96.7	**56**	50
	✓Q-RAG	100	100	100	100	100	100	100	100	100	52	**63**
	S²LMR (Ours)	100	100	100	100	100	100	100	100	100	50	**63**
128K	°LongRoPE2-8B	100	100	99	96	91	94	96.5	97	96.7	**56**	50
	✓Q-RAG	100	100	100	100	100	100	100	100	100	50	61
	S²LMR (Ours)	100	100	100	100	100	100	100	100	100	47	**62**
1M	✓Q-RAG	100	100	100	98.5	99.0	100	100	100	99.7	**50**	56
	S²LMR (Ours)	100	100	100	**100**	99.0	100	100	100	**99.8**	46	**59**

2) Long-context retrieval. Table 2 reports results on RULER across context lengths from 4K to 1M tokens. The baselines include Beam Retriever (Zhang et al., 2024), Titans (Behrouz et al., 2026), Atlas (Behrouz et al., 2025), Mamba2 (Dao & Gu, 2024), LongRoPE2 (Shang et al., 2025), and Q-RAG (Sorokin et al., 2026).

S²LMR consistently matches or exceeds Q-RAG on nearly every retrieval task across the full context range. On single-needle, multi-key, multi-value, and multi-query retrieval, both methods achieve near-perfect accuracy, while S²LMR also remains competitive with Q-RAG on long-context question answering and substantially outperforms previous long-context architectures, including recur-



(a) Learning-rate robustness.

(b) Batch-size robustness.

Figure 5: Hyperparameter robustness on BabiLong QA3. Shaded regions indicate the stable training range; black $\times$ marks Q-function collapse.

rent transformers and state-space models. These results demonstrate that S^{2}LMR remains effective across diverse retrieval tasks and context lengths, including contexts of up to 1M tokens.

3) Open-domain multi-hop retrieval. Table 3 compares S^{2}LMR with multiple strong baselines on HotPotQA, including Beam Retriever (Zhang et al., 2024), Q-RAG (Sorokin et al., 2026), Search-R1 (Jin et al., 2025), RAG-RL (Huang et al., 2025), GraphReader (Li et al., 2024), and a conventional single-step RAG baseline (Lewis et al., 2020). S^{2}LMR achieves the best downstream answer performance (Ans F1/EM) and consistently outperforms Q-RAG across all four metrics, while remaining competitive with Beam Retriever on supporting-fact retrieval. Full MuSiQue results are reported in Appendix Table 10. S^{2}LMR remains competitive with both Beam Retriever and Q-RAG and achieves the best average answer EM across the two benchmarks. Notably, S^{2}LMR fine-tunes only the dense retriever rather than the language model, yet achieves comparable downstream performance. These results show that S^{2}LMR generalizes beyond long-context retrieval to standard short-context multi-hop QA.

Table 3: Comparison on HotPotQA (1-2K tokens). Bold/underline mark best/2nd.

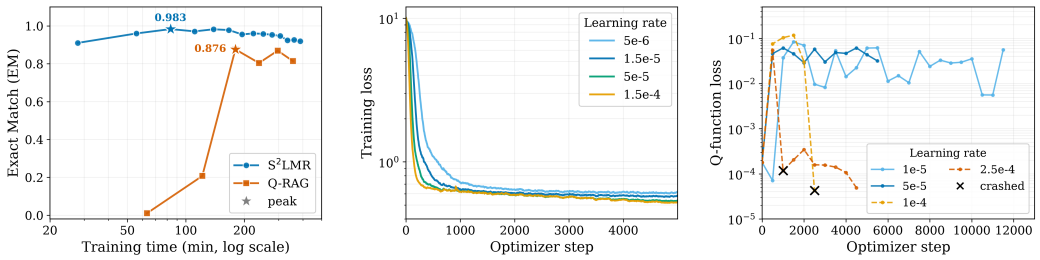
Methods	Fact F1	Fact EM	Ans F1	Ans EM
Finetuned				
✓Q-RAG	0.92	0.87	0.76	0.59
°Beam-Retriever	**0.97**	**0.94**	<u>0.77</u>	<u>0.61</u>
°Search-R1	0.81	0.66	0.65	0.52
°RAG-RL	0.82	–	0.69	–
S^{2}LMR (Ours)	<u>0.95</u>	<u>0.90</u>	**0.78**	**0.62**
Zero Shot				
°Multi-step RAG	0.73	0.54	0.65	0.50
°GraphReader	–	–	0.46	0.24
°Single step RAG	–	–	0.53	0.39

4.3 Training Stability

Training efficiency and stability directly affect the practical cost and reproducibility of retriever training. We compare S^{2}LMR with Q-RAG on BabiLong QA3 from two complementary perspectives: (1) robustness to hyperparameter choices and (2) stability of training dynamics.

1) Hyperparameter robustness. Figure 5 sweeps S^{2}LMR and Q-RAG over learning rate (panel a) and effective batch size (panel b), while keeping the trainable retriever backbone fixed. ***Learning-rate robustness.*** S^{2}LMR maintains near-peak performance across a learning-rate range spanning nearly two orders of magnitude ($\approx 100\times$). Q-RAG remains stable only within a much narrower $\approx 8\times$ range, with its Q-function collapsing at higher learning rates (crashes marked by $\times$). Thus, S^{2}LMR provides a usable learning-rate range more than an order of magnitude wider than Q-RAG's. ***Batch-size robustness.*** At a fixed 3K-step training budget, S^{2}LMR reaches near-peak EM across a $16\times$ batch-size range, from 16 to 256. Under the same budget, Q-RAG has a usable range of only $4\times$: at batch sizes below 64, it fails to reach peak performance within the standard budget and requires several times more optimizer steps to escape its initial low-EM phase. This additional step requirement further increases its practical training cost.

Beyond robustness to individual hyperparameters, S^{2}LMR also has fewer method-specific hyperparameters to tune. Q-RAG's value-based formulation introduces additional TD-related choices, including the discount factor, target-network update schedule, and value-network learning rate, which are absent from the supervised bi-encoder formulation.



(a) Peak-arrival speed. (b) S^{2}LMR training loss. (c) Q-RAG qf-loss.

Figure 6: Training stability on BabiLong QA3. (a) Peak EM vs. training time. (b) S^{2}LMR training loss across learning rates. (c) Q-RAG Q-function loss across learning rates.

2) Training dynamics. Figure 6 compares the training dynamics of S^{2}LMR and Q-RAG at the same learning rate, using Q-RAG's reported best setting. ***Peak-arrival speed.*** S^{2}LMR reaches its peak EM of 0.983 within ~ 83 minutes of wall-clock training, whereas Q-RAG requires ~ 180 minutes to reach a peak of 0.876 (Figure 6a). S^{2}LMR converges $2.2\times$ faster and to $+0.11$ higher final EM. ***Loss stability.*** Across the learning rates within each method's stable range, S^{2}LMR's training loss decreases smoothly, whereas Q-RAG's Q-function loss is substantially more volatile and diverges at larger learning rates (Figure 6b,c). The smoother optimization of S^{2}LMR also makes its performance less sensitive to the learning rate within the stable range, reducing the need for repeated restarts or fine-grained tuning.

4.4 Training Efficiency

We compare the training and inference costs of S^{2}LMR and representative baselines on a single NVIDIA RTX A6000 GPU (Table 4). For retrieval-based methods, we report retrieval time and add 0.44 s for Qwen3-4B answer generation; end-to-end LLM methods report total inference time. "OOM" denotes out-of-memory runs, and "limit" denotes runs exceeding the model's maximum context length. For methods without reported training compute, we estimate training time from reported FLOPs or training data size and calibrate it on our hardware. Full results across 1K/32K/128K/1M/10M-token contexts and additional baselines are provided in Appendix C.5. On BabiLong QA3, S^{2}LMR requires only 2.25 hours of training, compared with 54.4 hours for our Q-RAG reproduction, corresponding to a roughly $24\times$ lower training cost. A similar cost gap is observed on RULER. Compared with methods that fine-tune long-context language models, which can require tens to thousands of GPU-hours, S^{2}LMR achieves comparable or stronger retrieval performance at substantially lower training cost.

Table 4: Time cost on BabiLong QA3.

Method	**Train (h)**	**Infer. (s/sample)**			
		1K	128K	1M	10M
Retrieval-only (+0.44s Qwen3-4B)					
S^{2}LMR	**2.25**	0.20	3.20	27.8	279
Q-RAG	54.4	0.20	3.42	29.0	318
Beam Retriever	∼450	1.2	155	∼1200	∼12000
End-to-end fine-tuned					
RMT-137M	∼20	0.21	3.15	24.5	253
Llama3-8B+RAG	>40	0.65	1.23	7.82	64.8
Mamba-130M	>400	4.21	OOM	OOM	OOM
End-to-end zero-shot					
Mamba-2.8B	0	8.81	OOM	OOM	OOM
Llama-3.1-8B	0	0.61	49	limit	limit

5 Conclusion

We introduced S^{2}LMR, a simple supervised bi-encoder for long-context multi-step retrieval. We showed that supervised retrievers fall behind in this setting not because supervision is too weak, but because their training is misaligned with how they are used. By correcting these misalignments, S^{2}LMR matches or outperforms the strongest RL-based retriever at up to 10M tokens, while training substantially faster and remaining robust to hyperparameters. Future work includes learning when to stop retrieving instead of using a fixed number of hops, reducing the reliance on gold supporting-fact annotations through weaker supervision, and extending S^{2}LMR to heterogeneous retrieval corpora with evidence distributed across different sources.

AI USE STATEMENT

In this work, we used generative AI tools for assistance with writing, editing, and refining the organization and presentation of the manuscript, including improving clarity, concision, and technical exposition. We have not used generative AI tools to generate or modify experimental data or make final scientific claims. We have reviewed all AI-assisted work and verified the accuracy and consistency of the resulting text, claims, and reported results against our experiments, analyses, and cited sources. We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

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

Petr Anokhin, Nikita Semenov, Artyom Sorokin, Dmitry Evseev, Andrey Kravchenko, Mikhail Burtsev, and Evgeny Burnaev. Arigraph: Learning knowledge graph world models with episodic memory for llm agents. *arXiv preprint arXiv:2407.04363*, 2024.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

Ali Behrouz, Zeman Li, Praneeth Kacham, Majid Daliri, Yuan Deng, Peilin Zhong, Meisam Razaviyayn, and Vahab Mirrokni. Atlas: Learning to optimally memorize the context at test time. *arXiv preprint arXiv:2505.23735*, 2025.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *Advances in Neural Information Processing Systems*, 38:113506–113543, 2026.

Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. Recurrent memory transformer. *Advances in neural information processing systems*, 35:11079–11091, 2022.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. Multi-step retriever-reader interaction for scalable open-domain question answering. *arXiv preprint arXiv:1905.05733*, 2019.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

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

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Jerry Huang, Siddarth Madala, Risham Sidhu, Cheng Niu, Hao Peng, Julia Hockenmaier, and Tong Zhang. Rag-rl: Advancing retrieval-augmented generation via rl and curriculum learning. *arXiv preprint arXiv:2503.12759*, 2025.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pp. 874–880, 2021.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, 2024a.

Ziyan Jiang, Xueguang Ma, and Wenhu Chen. Longrag: Enhancing retrieval-augmented generation with long-context llms. *arXiv preprint arXiv:2406.15319*, 2024b.

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.

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 *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 6769–6781, 2020.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *Advances in Neural Information Processing Systems*, 37:106519–106554, 2024.

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

Shilong Li, Yancheng He, Hangyu Guo, Xingyuan Bu, Ge Bai, Jie Liu, Jiaheng Liu, Xingwei Qu, Yangguang Li, Wanli Ouyang, et al. Graphreader: Building graph-based agent to enhance long-context abilities of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 12758–12786, 2024.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4089–4100, 2021.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D Manning. Answering complex open-domain questions through iterative query generation. In *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, 2019.

Ivan Rodkin, Yuri Kuratov, Aydar Bulatov, and Mikhail Burtsev. Associative recurrent memory transformer. *arXiv preprint arXiv:2407.04841*, 2024.

Ning Shang, Li Lyna Zhang, Siyuan Wang, Gaokai Zhang, Gilsinia Lopez, Fan Yang, Weizhu Chen, and Mao Yang. Longrope2: Near-lossless llm context window scaling. *arXiv preprint arXiv:2502.20082*, 2025.

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 *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9248–9274, 2023.

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.

Artyom Sorokin, Nazar Buzun, Aleksandr Anokhin, Egor Vedernikov, Petr Anokhin, Mikhail Burtsev, and Evgeny Burnaev. Q-rag: Long context multi-step retrieval via value-based embedder training. In *International Conference on Learning Representations*, volume 2026, pp. 32481–32506, 2026.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

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.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 10014–10037, 2023.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, et al. Answering complex open-domain questions with multi-hop dense retrieval. *arXiv preprint arXiv:2009.12756*, 2020.

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.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Liu Yong, and Shen Huang. End-to-end beam retrieval for multi-hop question answering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1718–1731, 2024.

Algorithm 1 S^{2}LMR: Supervised Sequential Long-Context Multi-Step Retriever (one training update)

1: **Hyperparameters:** batch size B, hops K, temperature τ, RoPE step Δ, learning rate η.
2: **Model:** shared bi-encoder f_θ; chunk-axis $\mathrm{RoPE}(\cdot, p)$.
3: Sample batch $\big\{\big(q^{(b)}, \{c_j^{(b)}\}_{j=1}^N, (i_1^{\star,(b)}, \ldots, i_K^{\star,(b)})\big)\big\}_{b=1}^B$
4: **for** $b = 1, \ldots, B$ **do**
5: $\quad h_j \leftarrow f_\theta(c_j^{(b)})$ for $j = 1, \ldots, N$ ▷ encode candidates once
6: $\quad \mathcal{S}_1 \leftarrow \varnothing$
7: $\quad$ **for** $t = 1, \ldots, K$ **do**
8: $\quad\quad$ build s_t via Eq. 1 with $c^{(k)} = c^{(b)}_{i_k^{\star,(b)}}$ for $k < t$ ▷ teacher-forced state
9: $\quad\quad$ compute $(p_j^{(t)})_{j=1}^N$ from $\mathcal{S}_t$ via Eq. 2
10: $\quad\quad \mathcal{A}_t \leftarrow \{1, \ldots, N\} \setminus \mathcal{S}_t$
11: $\quad\quad \tilde{h}_j^{(t)} \leftarrow \mathrm{RoPE}(h_j,\, p_j^{(t)})$ for all $j \in \mathcal{A}_t$
12: $\quad\quad L_t^{(b)} \leftarrow -\log \dfrac{\exp\big(\langle f_\theta(s_t), \tilde{h}^{(t)}_{i_t^{\star,(b)}}\rangle/\tau\big)}{\sum_{j\in\mathcal{A}_t}\exp\big(\langle f_\theta(s_t), \tilde{h}_j^{(t)}\rangle/\tau\big)}$ ▷ Eq. 4
13: $\quad\quad \mathcal{S}_{t+1} \leftarrow \mathcal{S}_t \cup \{i_t^{\star,(b)}\}$
14: $\quad$ **end for**
15: $\quad L^{(b)} \leftarrow \frac{1}{K}\sum_{t=1}^K L_t^{(b)}$
16: **end for**
17: $\mathcal{L} \leftarrow \frac{1}{B}\sum_{b=1}^B L^{(b)};\ \theta \leftarrow \theta - \eta\,\nabla_\theta \mathcal{L}$

A S^{2}LMR TRAINING PROCEDURE

Algorithm 1 makes the training loop of Sec. 3.2 explicit for one batched update. Candidate chunks are encoded once per example (line 5); at every hop the query state s_t is rebuilt from the gold prefix (line 8), the chunk-axis positions $p_j^{(t)}$ are recomputed from the already-selected indices $\mathcal{S}_t$ (line 9), and candidate embeddings are re-rotated accordingly (line 10). The per-hop InfoNCE loss (line 12) instantiates Eq. 4 on $\mathcal{A}_t = \{1, \ldots, N\} \setminus \mathcal{S}_t$, and the example loss averages over the K hops before the batch gradient step.

B CONNECTION BETWEEN VALUE-BASED RL AND SUPERVISED RETRIEVAL

This section formalizes the connection between the value-based RL formulation of Q-RAG and direct supervised learning for multi-step retrieval. We show that under a particular limiting regime, namely $T = K$, $\lambda = 1$, and $\alpha \to 0$, the Q-RAG learning signal reduces to binary supervision over sampled state–action pairs. This does not imply that the Q-RAG and S^{2}LMR objectives are mathematically identical, rather, it shows that the distinctive components of value-based RL become unnecessary for constructing the learning signal in this regime.

B.1 PROBLEM SETUP

Consider a multi-step retrieval problem with K ground-truth supporting facts. At retrieval step t, the state consists of the query and the supporting facts retrieved at previous steps. S^{2}LMR constructs the state s_t by teacher forcing (Eq. 1 with $c^{(k)} = c_{i_k^\star}$ for $k < t$). For brevity, let

$$\tilde{h}_j^{(t)} \;=\; \mathrm{RoPE}\big(f_\theta(c_j),\, p_j^{(t)}\big)$$

denote the rotated candidate embedding at hop t, so that $\mathrm{score}(s_t, c_j) = \langle f_\theta(s_t),\, \tilde{h}_j^{(t)}\rangle$ (Eq. 3). The per-hop InfoNCE objective from Eq. 4 can then be written as

$$\mathcal{L}_t^{\mathrm{S^2LMR}} = -\log \frac{\exp\big(\langle f_\theta(s_t),\, \tilde{h}^{(t)}_{i_t^\star}\rangle/\tau\big)}{\sum_{j\in\mathcal{A}_t}\exp\big(\langle f_\theta(s_t),\, \tilde{h}_j^{(t)}\rangle/\tau\big)}, \tag{5}$$

where τ is the contrastive temperature and $\mathcal{A}_t = \mathcal{C} \setminus \{i_1^\star, \ldots, i_{t-1}^\star\}$ is the remaining candidate set at hop t.

Q-RAG instead defines a value function over state–action pairs,

$$Q(s_t, c_j) \;=\; \langle \phi(s_t),\, \psi(c_j) \rangle,$$

and uses a Boltzmann policy

$$\pi_\alpha(c_j \mid s_t) \;=\; \frac{\exp(Q(s_t, c_j)/\alpha)}{\sum_{j' \in \mathcal{A}_t} \exp(Q(s_t, c_{j'})/\alpha)}, \tag{6}$$

where α controls the degree of stochastic exploration. The value function is trained by minimizing the squared error to a λ-return:

$$\mathcal{L}^{\text{Q-RAG}} \;=\; \mathbb{E}\Big[\big(Q(s_t, c_{i_t}) - G_t^\lambda\big)^2\Big].$$

For the multi-hop QA setting considered here, the terminal reward is constructed from the ground-truth supporting-fact indices:

$$r_T \;=\; \mathbb{1}[\{i_1^\star, \ldots, i_K^\star\} \subseteq \{i_1, \ldots, i_T\}].$$

Thus, a trajectory receives positive reward when the retrieved set covers all annotated supporting facts.

B.2 Effect of the Retrieval Horizon

We first consider the limiting case in which the retrieval horizon equals the number of supporting facts:

$$T \;=\; K.$$

Assume that additional retrieval beyond the K required supporting facts is penalized, as in the setting considered in our analysis. Since a successful trajectory must contain all K supporting facts and contains exactly K retrieval steps, a successful trajectory cannot contain an additional non-supporting passage. Therefore,

$$r_T \;=\; \begin{cases} 1, & \{i_1, \ldots, i_K\} = \{i_1^\star, \ldots, i_K^\star\}, \\ 0, & \text{otherwise.} \end{cases}$$

This condition removes the ambiguity caused by extra retrieved passages. In the general set-based reward formulation, a trajectory may receive reward 1 even if it contains irrelevant passages in addition to all supporting facts. When $T = K$, this ambiguity disappears because there is no additional retrieval step available after covering the K supporting facts.

B.3 Effect of the λ-Return

Next, consider the case

$$\lambda \;=\; 1.$$

With only a terminal reward, the λ-return reduces to the Monte Carlo return:

$$G_t^\lambda \;\overset{\lambda=1}{=}\; G_t^{\text{MC}} \;=\; r_T,$$

up to the discounting convention used by the implementation. In the undiscounted setting considered here, this gives $G_t^\lambda \in \{0, 1\}$.

Consequently, the Q-RAG objective becomes

$$\mathcal{L}^{\text{Q-RAG}} \;=\; \mathbb{E}\Big[\big(Q(s_t, c_{i_t}) - y_t\big)^2\Big], \qquad y_t = r_T \in \{0, 1\}.$$

Thus, the target no longer depends on a bootstrapped estimate of a future value. A state–action pair on a successful trajectory receives target 1, whereas a state–action pair on a failed trajectory receives target 0. This removes the bootstrapping component that distinguishes the general temporal-difference formulation.

B.4 Deterministic Policy Limit

Finally, consider the limit

$$\alpha \;\to\; 0.$$

For the Boltzmann policy in Eq. 6, the probability mass concentrates on actions with the maximum value:

$$\lim_{\alpha \to 0} \pi_\alpha(c_j \mid s_t) \;=\; \begin{cases} 1, & c_j \in \arg\max_{c_{j'}} Q(s_t, c_{j'}), \\ 0, & \text{otherwise}, \end{cases}$$

with an arbitrary tie-breaking rule when multiple actions attain the same maximum. Hence, in this limit, sampling becomes equivalent to greedy action selection:

$$i_t \;=\; \arg\max_{j \in \mathcal{A}_t} Q(s_t, c_j).$$

The stochastic exploration component of the policy is therefore removed. Importantly, this limit alone does not guarantee a successful trajectory; rather, once the policy has converged to a successful deterministic policy, the resulting rollouts repeatedly expose state–action pairs that lead to successful retrieval.

B.5 Resulting Learning Signal

Combining $T = K$, $\lambda = 1$, and $\alpha \to 0$, the terminal outcome provides a binary target for each sampled state–action pair:

$$y_t \;=\; \begin{cases} 1, & (s_t, c_{i_t}) \text{ on a successful trajectory}, \\ 0, & (s_t, c_{i_t}) \text{ on a failed trajectory}. \end{cases}$$

The Q-RAG objective consequently takes the form

$$\mathcal{L}^{\text{Q-RAG}} \;=\; \mathbb{E}\Big[\big(Q(s_t, c_{i_t}) - y_t\big)^2\Big], \qquad y_t \in \{0, 1\}.$$

The corresponding gradient with respect to the parameters of Q is

$$\nabla \mathcal{L}^{\text{Q-RAG}} \;=\; 2\big(Q(s_t, c_{i_t}) - y_t\big)\, \nabla Q(s_t, c_{i_t}).$$

Therefore, successful state–action pairs are pushed toward value 1, while state–action pairs observed on failed trajectories are pushed toward value 0. In this limiting regime, the Q-RAG learning signal can thus be viewed as a binary supervised regression objective over sampled trajectory actions.

B.6 Relation to S^{2}LMR

S^{2}LMR instead uses the contrastive objective in Eq. 5. Its gradient with respect to the query-state embedding is

$$\nabla_{f_\theta(s_t)} \mathcal{L}_t^{\text{S}^2\text{LMR}} \;=\; -\frac{1}{\tau}\left(\tilde{h}_{i_t^\star}^{(t)} \;-\; \sum_{j \in \mathcal{A}_t} p_\tau(j \mid s_t)\, \tilde{h}_j^{(t)}\right),$$

where

$$p_\tau(j \mid s_t) \;=\; \frac{\exp(\langle f_\theta(s_t),\, \tilde{h}_j^{(t)} \rangle / \tau)}{\sum_{j' \in \mathcal{A}_t} \exp(\langle f_\theta(s_t),\, \tilde{h}_{j'}^{(t)} \rangle / \tau)}.$$

Thus, S^{2}LMR increases the score of the ground-truth supporting passage while decreasing the scores of competing candidates, with the magnitude of the negative signal determined by their current softmax probabilities.

The two objectives are therefore not mathematically identical. Their difference is summarized in Table 5.

The key connection is therefore not an exact equivalence between the two loss functions. Rather, under the limiting regime considered above, the Q-RAG formulation loses the main components that distinguish value-based RL from direct supervised learning: extra-step credit assignment, bootstrapping, and stochastic exploration. What remains is a supervised learning signal that encourages relevant actions to receive higher scores than sampled irrelevant actions. S^{2}LMR directly provides this signal through fixed per-hop supervision and contrasts each positive passage against all in-context candidates.

Table 5: Comparison between S^{2}LMR and the limiting form of Q-RAG.

Property	S^{2}LMR	Q-RAG limiting case
Learning signal	Fixed per-hop supervision	Sampled trajectory outcome
Objective	InfoNCE	Binary MSE
Positive signal	Ground-truth supporting passage	Action on successful trajectory
Negative signal	All in-context candidates	Sampled actions on failed trajectories
Bootstrapping	None	None ($\lambda = 1$)
Exploration	None	None in the limit ($\alpha \to 0$)
State construction	Teacher forcing	Policy-generated trajectory

Table 6: Hop-1 median cosine similarities on 128 BabiLong QA3 samples (RoPE-rotated). $\text{margin}_{\text{sc}} = \text{sim(gold)} - \text{sim(sc-neg)}$; $\text{margin}_{\text{ib}} = \text{sim(gold)} - \text{sim(ib-neg)}$. MDR training widens the in-batch margin more than the same-context margin (its own objective converges on the wrong distribution). S^{2}LMR training widens the same-context margin most, matching what inference requires. All three checkpoints use identical encoder/tokenizer/RoPE.

Checkpoint	sim(gold)	sim(sc-neg)	sim(ib-neg)	$\text{margin}_{\text{sc}}$	$\text{margin}_{\text{ib}}$
Contriever (zero-shot)	0.29	0.23	0.23	+0.06	+0.06
MDR-repro (in-batch trained)	0.45	0.36	0.34	+0.09	**+0.11**
S^{2}LMR (same-context trained)	0.67	0.46	0.54	**+0.21**	+0.13

This analysis motivates our empirical comparison: when supporting-fact supervision is available, value-based RL introduces additional optimization machinery beyond the direct supervised signal, while it remains an empirical question whether that additional machinery yields better retrieval performance.

C Additional Results

C.1 Empirical evidence for Issue 1 (distractor-distribution mismatch)

To isolate what MDR-style in-batch training actually optimizes, we compare the encoder's hop-1 discrimination on the two negative distributions that matter—the *training pool* (other in-batch queries' gold chunks) and the *inference pool* (same-document distractors)—across three checkpoints: (i) a zero-shot Contriever initialization, (ii) an MDR-style reproduction trained on in-batch negatives, and (iii) the S^{2}LMR checkpoint trained on same-context negatives. All three share the same encoder, tokenizer, and RoPE positional encoding. Query states are teacher-forced with the true hop-0 answer so measurements isolate per-hop discrimination.

The pattern is symmetric at initialization ($\text{margin}_{\text{sc}}$=$\text{margin}_{\text{ib}}$=0.06) and diverges opposite to inference needs under MDR training ($\text{margin}_{\text{ib}}$>$\text{margin}_{\text{sc}}$). MDR's gradient signal itself is strong (median p(gold) on its in-batch pool is 0.04, implying gradient-magnitude proxy ≈ 0.96), so the failure is not gradient collapse but distributional misalignment—the encoder converges to a boundary that separates the wrong distribution. Consequently the MDR checkpoint ranks the hop-1 gold above the 99 same-document distractors on 0 of 128 samples, whereas S^{2}LMR does so on all 128 (Figure 7).

C.2 Teacher forcing or not: supervision noise vs. exposure bias

BR-style beam-expanded supervision was originally motivated by an exposure-bias concern: training only on gold prefixes leaves the model unfamiliar with the noisier trajectories it will produce at inference. S^{2}LMR accepts this trade-off because the alternative—training on noisy predicted prefixes—is empirically more harmful in the long-context regime. Fig. 2c and Fig. 3b show that as context length grows, the beam-expanded performance degrades noticeably faster than the teacher-forced curve, while at the training length the two are within 0.04 Fact EM (Table 2c, Issue 2). The exposure gap that teacher forcing incurs is small enough to be dominated by per-hop InfoNCE against the full same-context pool, which keeps the gold-vs-distractor margin large enough that small prefix perturbations at inference rarely flip the argmax.

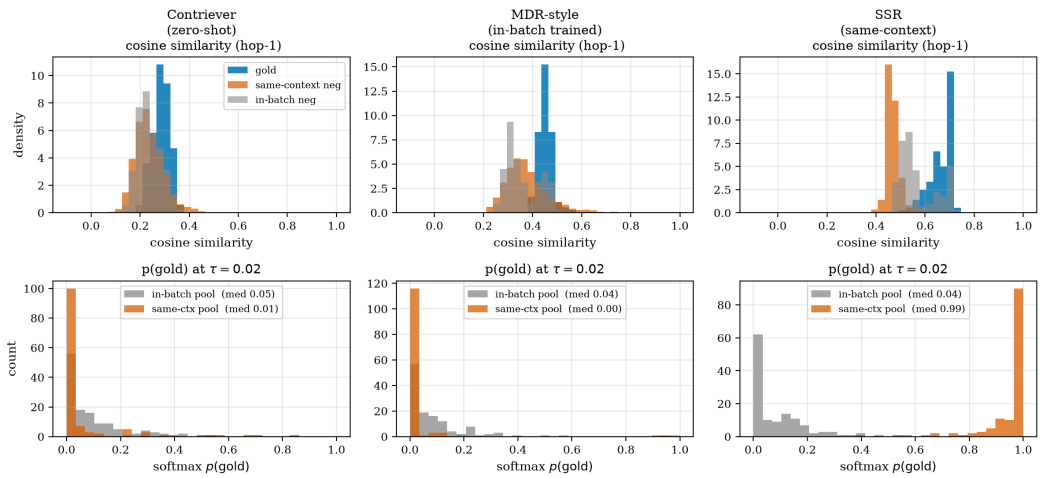


Figure 7: Distractor-distribution mismatch on 128 BabiLong QA3 samples. **Top row:** distributions of cosine similarity between the query state and (i) the gold chunk, (ii) 31 same-document distractors, (iii) 31 in-batch negatives. **Bottom row:** distributions of softmax $p(\text{gold})$ at τ=0.02 under the in-batch pool (MDR training pool, 32 candidates) and the same-context pool (inference pool, 32 candidates). Columns: Contriever (zero-shot), MDR-style checkpoint trained with in-batch negatives, and S^{2}LMR trained with same-context negatives.

C.3 Positional bottleneck in chunk-position-agnostic retrievers

This appendix expands the analysis in Sec. 3.1 of the positional-encoding bottleneck in existing supervised multi-hop retrievers. We (i) enumerate the architectural facts that make MDR and BR chunk-position-agnostic, (ii) quantify the positional structure of BabiLong QA3 that these architectures cannot exploit, (iii) measure the resulting per-hop retrieval margin, and (iv) show the failure mode is stable across all three hops.

C.3.1 Architecture: token-axis only positional encoding

MDR (Xiong et al., 2020). MDR uses a shared RoBERTa-base bi-encoder to embed candidates and queries independently. Each candidate chunk c_j is embedded as $E_a(c_j) = \text{RoBERTa}(c_j)$. Only tokens within c_j receive positional information via RoBERTa's absolute learned positional embeddings (maximum position 514). The resulting embedding depends solely on the text of c_j and is invariant to c_j's document-level position. At the query side, MDR concatenates the question with previously retrieved passages and re-encodes. The maximum sequence length is again 514 tokens, and no chunk-level position is exposed.

BR (Zhang et al., 2024). BR uses a DeBERTa cross-encoder that scores candidates in flat concatenation $[\text{CLS}; Q; \hat{p}_1; \ldots; \hat{p}_{t-1}; c_j; \text{SEP}]$ (maximum 512 tokens). DeBERTa's disentangled relative positional encoding operates on token-token offsets within this flat sequence. BR's Appendix E *explicitly shuffles the order of concatenated passages during training* to enforce robustness to permutations. This is architecturally equivalent to training the encoder to be order-invariant across chunks: any positional distinction between the concatenated passages that could have arisen from token positions in the flat sequence is deliberately regularized away.

Common consequence. In both architectures, the retrieval decision at hop t depends only on the *textual* content of c_j and the (question, previous picks) tuple, never on where c_j sits in the source document. We call this property *chunk-position-agnostic*.

C.3.2 Task-side positional structure in BabiLong QA3

BabiLong (Kuratov et al., 2024) embeds bAbI supporting facts as a temporally-ordered subsequence within a long stream of PG-19 noise sentences. QA3 ("three-supporting-facts") requires retrieving three temporally consecutive facts about a person's location history. Because the story is ordered in

Table 7: **Concentration of next-gold document-position offsets in BabiLong QA3**, computed on 500 samples per N. The last column reports the fraction of samples with $|\delta_t|/N \leq 0.10$ (range across the two hop transitions). The distribution is scale-invariant in N and far tighter than the uniform baseline, giving any position-aware retriever a substantial candidate-pool reduction.

	$\lvert\delta_2\rvert/N$			$\lvert\delta_3\rvert/N$			$P(\lvert\delta_t\rvert/N$
N	median	IQR	90th pct	median	IQR	90th pct	$\leq 0.10)$
Uniform	0.333	0.500	0.900	0.333	0.500	0.900	0.190
100	0.060	0.090	0.200	0.050	0.070	0.210	0.70–0.79
800	0.056	0.092	0.218	0.048	0.089	0.182	0.69–0.73
6400	0.055	0.088	0.214	0.049	0.087	0.193	0.71–0.73

time and the question refers to a temporal anchor ("*Where was X before Y?*"), each gold fact appears in a specific positional relation to the others.

Case study. Consider a QA3 sample from the test set with question *"Where was the apple before the bathroom?"* and answer *"office"*. The three supporting facts, ordered as the model must retrieve them under fixed-order supervision, are:

1. (chunk 48): *"John discarded the apple there."*
2. (chunk 31): *"John journeyed to the bathroom."*
3. (chunk 27): *"John went to the office."*

The doc positions are strictly decreasing: $48 \to 31 \to 27$. This is determined by the semantics of the question — the temporal marker "*before*" places each subsequent gold *earlier* in the story than the previously picked one. A question with "*after*" would invert the direction. The direction is therefore *semantically conditional*: which side of the picked chunk to search depends on the question, and cannot be reduced to a fixed rule.

Statistical structure. Table 7 reports the distribution of the normalized offset $|\delta_t|/N$, where $\delta_t = \mathrm{sf}_t - \mathrm{sf}_{t-1}$, across 500 QA3 samples at each $N \in \{100, 800, 6400\}$. Two properties stand out:

1. **Concentration.** The median $|\delta_t|/N$ is 0.05–0.06 (vs. 0.33 under uniform sampling), and 70% of samples have $|\delta_t|/N \leq 0.10$. The IQR is 0.07–0.09 (vs. 0.50 uniform). A retriever that could restrict candidates to $|\text{position} - \text{last pick}| \leq 0.10N$ would retain 70% of golds while shrinking the candidate pool $10\times$.
2. **Scale invariance.** The distribution normalized by N is essentially identical at $N = 100$, $N = 800$, and $N = 6400$: the positional structure is a property of the task, not of the context size.

The case study and statistics together establish a claim we will use repeatedly: *document position is a strong, semantically-conditional retrieval signal in BabiLong QA3*. A chunk-position-agnostic encoder cannot access this signal.

C.3.3 Semantic-only ranking fails at every hop

We now measure the retrieval margin achievable by a fine-tuned bi-encoder that mirrors MDR's chunk-position-agnostic architecture.

Proxy encoder. We use the Contriever (Izacard et al., 2021) bi-encoder architecture, fine-tuned on BabiLong QA3 ($N = 100$) for 2000 optimizer steps with per-hop InfoNCE loss, label smoothing 0.05, and CCR regularization $\lambda = 0.02$, with chunk-axis positional encoding disabled. Every element of the training setup matches our other BabiLong retrievers except the absence of chunk-axis positional information. The resulting encoder is architecturally chunk-position-agnostic while task-specifically fine-tuned.

Table 8: **Hop-1 semantic margin of a fine-tuned chunk-position-agnostic bi-encoder is scale-invariant in N.** Aggregated over 200 samples per N. The margin is ≈ 0 regardless of context size.

N	10	20	50	100	800	6400
gold_max_cos (median)	0.962	0.962	0.962	0.962	0.962	0.962
best_noise_cos (median)	0.963	0.963	0.963	0.963	0.963	0.963
margin (median)	−0.001	−0.001	−0.001	−0.001	−0.001	−0.001
$P(\text{margin} \leq 0)$	0.670	0.670	0.670	0.670	0.670	0.655

Table 9: **Per-hop $P(\text{margin} \leq 0)$ under oracle previous picks.** The failure rate is stable at every hop across N. Hop 3 is the hardest, hop 2 marginally easier due to the injected gold-prev-context.

N	Hop 1	Hop 2	Hop 3
100	0.670	0.530	0.795
800	0.670	0.520	0.795
6400	0.660	0.485	0.795

Metric. For each sample and each hop t, we encode the query context (question concatenated with oracle previous gold picks $\text{sf}_1, \ldots, \text{sf}_{t-1}$), score all remaining chunks by inner product with the query embedding, and record

$$\text{gold_max_cos}_t = \max_{j \in \text{gold}} \cos(q_t, c_j),$$
$$\text{best_noise_cos}_t = \max_{j \notin \text{gold}} \cos(q_t, c_j),$$
$$\text{margin}_t = \text{gold_max_cos}_t - \text{best_noise_cos}_t.$$

$P(\text{margin}_t \leq 0)$ is the fraction of samples on which the top-scoring chunk (which the retriever picks by argmax) is a distractor rather than a gold.

Scale invariance. Table 8 sweeps N from 10 to 6400 at hop 1. Both gold_max_cos and best_noise_cos are stable (0.962 vs. 0.963, respectively), the median margin sits at -0.001, and $P(\text{margin} \leq 0)$ ranges only from 0.655 to 0.670. Fine-tuning has successfully pushed both gold and distractors to high cosine (≈ 0.96), but has not opened any margin between them. And this failure is independent of the candidate pool size, ruling out "noise-tail-grows-with-N" as the explanation. The bottleneck is the encoder's inability to semantically separate a gold bAbI fact from a same-domain distractor.

Why fine-tuning cannot separate gold from distractors. BabiLong noise chunks are sampled from PG-19 book text but interleaved with bAbI-style distractor sentences that share the same syntactic template as the gold facts (subject-verb-location: "*Mary moved to the office.*"). Under InfoNCE fine-tuning, the encoder learns to give high cosine to any (question, bAbI-template sentence about the queried entity) pair. Because the distractors are indistinguishable from golds under a purely semantic similarity criterion — both mention the queried entity, both involve a location and an action — the fine-tune procedure pushes them to the same region of the embedding space. Positional information is the only signal that separates them, and this signal is absent by construction.

C.3.4 Per-hop failure persists across all three hops

Table 9 extends the measurement to hops 2 and 3, with oracle previous gold picks fed as prev-context. Two observations sharpen the diagnosis:

- **The failure mode persists at every hop.** $P(\text{margin} \leq 0)$ is 0.66 at hop 1, 0.53 at hop 2, and 0.80 at hop 3 (all at $N = 100$). The chunk-position-agnostic architecture fails to distinguish gold from distractor at every step of the multi-hop chain, not merely at hop 1.
- **Hop 3 is the hardest.** Hop 2 is slightly easier because the single gold prev-context injects task-specific semantic content that the encoder can exploit ($P(\text{margin} \leq 0) = 0.53$ vs. 0.66 at hop 1). Hop 3 is the hardest ($P = 0.80$): with two prev-context chunks the query context is longer and more diffuse, and the encoder's discriminative capacity is exceeded.

Table 10: Comparison of methods on HotPotQA and Musique. Bold text and underline denote the best and second best scores respectively.

Methods	HotPotQA				Musique (in-dist.)				Avg	
	Fact F1	Fact EM	Ans F1	Ans EM	Fact F1	Fact EM	Ans F1	Ans EM	Ans F1	Ans EM
Finetuned										
✓ Q-RAG (reproduced)	0.92	0.87	0.76	0.59	<u>0.79</u>	<u>0.68</u>	**0.59**	**0.43**	**0.68**	<u>0.51</u>
✓ Beam-Retriever	**0.97**	**0.94**	<u>0.77</u>	<u>0.61</u>	**0.86**	**0.69**	**0.59**	**0.43**	**0.68**	**0.52**
✓ Search-R1	0.81	0.66	0.65	0.52	–	–	–	–	–	–
○ RAG-RL	0.82	–	0.69	–	–	–	–	–	–	–
✓ **S^{2}LMR (Ours)**	<u>0.95</u>	<u>0.90</u>	**0.78**	**0.62**	<u>0.79</u>†	0.67	<u>0.56</u>	<u>0.42</u>	<u>0.67</u>	**0.52**
Zero Shot / no fine-tune										
× Multi-step RAG w.o. FT	0.73	0.54	0.65	0.50	0.51	0.30	0.40	0.27	0.53	0.39
✓ GraphReader	–	–	0.46	0.24	–	–	0.40	0.20	0.43	0.22
✓ Single step RAG	–	–	0.53	0.39	–	–	0.28	0.17	0.41	0.28

Table 11: Training and inference cost on BabiLong QA3. Retrieval-based methods (S^{2}LMR, Q-RAG, Beam-Retriever) report retrieval time only; the shared Qwen3-4B answer step is listed separately. End-to-end LLM methods report total inference (context + generation). OOM: out of memory. limit: exceeds model context length.

Method	Train (h)	Inference (s / sample)				
		1K	32K	128K	1M	10M
Ours (retrieval only)						
S^{2}LMR	**2.25**	0.20	1.00	3.20	27.8	279
Fine-tuned retrieval baselines (retrieval only)						
Q-RAG	54.4	0.20	1.10	3.42	29.0	318
Beam-Retriever	∼450	1.2	39	155	∼1200	∼12000
Answer generation (shared, added to any row above)						
+ Qwen3-4B answer	–	0.44	0.44	0.44	0.44	0.44
End-to-end fine-tuned baselines (total)						
RMT-137M	∼20	0.21	0.90	3.15	24.5	253
ARMT-137M	∼20	–	–	–	–	–
Llama3-ChatQA-1.5-8B+RAG	>40	0.65	0.75	1.23	7.82	64.8
Mamba-130M	>400	4.21	81.8	OOM	OOM	OOM
End-to-end zero-shot (total)						
Mamba-2.8B	0	8.81	133	OOM	OOM	OOM
Meta-Llama-3.1-8B-Instruct	0	0.61	11.9	49	limit	limit
Meta-Llama-3.1-70B-Instruct	0	–	–	–	–	–
gpt-4-0125-preview	0	–	–	–	–	–

Combined K=3 retrieval ceiling. Under oracle previous picks, the probability that the retriever picks the correct chunk at every hop is bounded above by the product of per-hop success rates: $0.34 \times 0.47 \times 0.20 = 0.031$. This is a strict upper bound: under actual (non-oracle) retrieval, incorrect early-hop picks would degrade the query context at later hops further. A chunk-position-agnostic bi-encoder, however extensively fine-tuned, cannot exceed this ceiling on BabiLong QA3.

C.4 OPEN-DOMAIN MULTI-HOP RETRIEVAL

Table 10 reports the full HotPotQA + Musique numbers.

C.5 TRAINING AND INFERENCE TIME ON BABILONG QA3.

Table 11 extends Table 4 to all measured context lengths (1K, 32K, 128K, 1M, 10M); all timings are measured on a single A6000 GPU with batch size 1.

C.6 HYPERPARAMETER ABLATION

Beyond robustness to training hyperparameters (Section 4.3), we evaluate S^{2}LMR's sensitivity to its three method-specific hyperparameters—the InfoNCE temperature τ, the confidence-regularizer

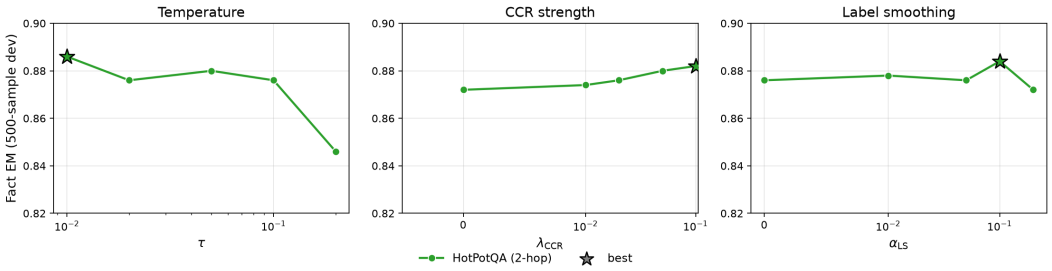


Figure 8: Fact EM vs (a) softmax temperature τ , (b) CCR strength λ_{CCR}, (c) label smoothing α_{LS}. Stars denote per-task best.

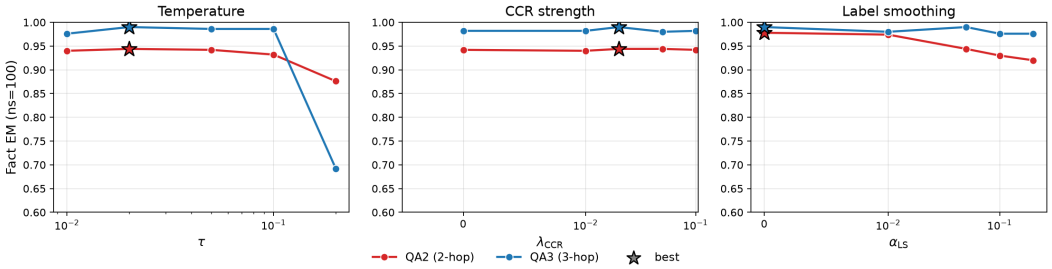


Figure 9: Fact EM vs (a) softmax temperature τ, (b) CCR strength λ_{CCR}, (c) label smoothing α_{LS}. Stars denote per-task best. S^{2}LMR maintains Fact EM > 0.94 (QA2) and > 0.97 (QA3) across a 10× range of each hyperparameter, degrading only at extreme values ($\tau = 0.2$ or $\alpha_{LS} \geq 0.05$).

weight λ_{CCR}, and the label-smoothing coefficient α_{LS}—across the long-context regime (BabiLong QA2 and QA3; Figure 9) and the open-domain regime (HotpotQA; Figure 8).

Across a $10\times$ range of each hyperparameter, S^{2}LMR's Fact EM stays within a narrow band—> 0.94 on QA2, > 0.97 on QA3, and within a ~ 0.04 band around 0.88 on HotpotQA—degrading only at extreme values ($\tau = 0.2$ or $\alpha_{\text{LS}} \geq 0.05$). The three benchmarks share overlapping optimal regions across all three hyperparameters, so a single default setting transfers across both long-context and open-domain retrieval without per-task tuning. Controlled component ablations further validate the contribution of S^{2}LMR's core design choices (Appendix C.7). Together with the training-hyperparameter robustness of Section 4.3, this shows that S^{2}LMR can be deployed on a new benchmark with minimal hyperparameter search.

C.7 Leave-One-Out Ablation of Core Mechanisms

Table 12 reports Fact EM on BabiLong QA3 when each of the three components corresponding to Issue 1–Issue 3 identified in Section 3.1 is individually removed from the full S^{2}LMR while all other design choices are held fixed.

Removing same-context negatives (Issue 1) drops EM to 0.28 (ΔEM $= -0.71$), confirming that distractor-matched negative sampling is essential rather than incidental. Replacing teacher-forced per-hop supervision with beam-expanded per-hop supervision in the spirit of Beam Retriever (Zhang et al., 2024) (Issue 2) costs 0.04 EM, showing that beam expansion of prev-hop contexts is neither necessary nor optimal. Removing chunk-axis RoPE (Issue 3) causes a near-total collapse of retrieval performance ($0.99 \rightarrow 0.11$, ΔEM $= -0.89$), the largest single source of gain in this ablation. Together, the three mechanisms are individually necessary and jointly account for essentially the entire gap between vanilla supervised bi-encoders and S^{2}LMR.

D Per-Benchmark Training Hyperparameters

Table 13 lists the main training hyperparameters for S^{2}LMR. Shared defaults across all runs: per-sample per-hop InfoNCE with τ=0.02, label smoothing α_{LS}=0.05, and confidence-regularizer weight λ_{CCR}=0.02; AdamW with weight decay 0.01 and gradient clipping 1.0; cosine LR schedule with 100 warmup steps; mean pooling with ℓ_2-normalized embeddings; and greedy argmax

Table 12: Leave-one-out ablation of S^{2}LMR's three core mechanisms on BabiLong QA3 (K=3, strict Fact EM).

Variant	Fact EM	ΔEM
S^{2}LMR (full)	**0.992**	—
− Issue 1 (same-context → in-batch negatives)	0.28	−0.71
− Issue 2 (teacher-forced → beam-expanded per-hop, B=2)	0.948	−0.04
− Issue 3 (chunk-axis RoPE → no chunk-axis PE)	0.106	−0.89

Table 13: Per-benchmark training hyperparameters for S^{2}LMR. $L_{\max}$: max tokens per chunk; Δ: chunk-axis RoPE step (Eq. 2).

Benchmark	$L_{\max}$	Batch	LR	Steps	Δ
BabiLong QA1/2/4/5	256	8	1.5×10^{-5}	2000	20
BabiLong QA3	64	32	1.06×10^{-5}	2000	200
HotpotQA	256	8	1.5×10^{-5}	2000	–
MuSiQue	512	8	1.5×10^{-5}	2500	–
RULER NIAH (S/MK/MV/MQ)	256	8	1.5×10^{-5}	2000	–

inference. At inference we set the retrieval horizon T equal to K, the number of gold supporting facts per query. BabiLong runs use `facebook/contriever`; all other benchmarks use `intfloat/multilingual-e5-large`. "Δ" is the chunk-axis RoPE step (Eq. 2); "–" means chunk-axis PE is disabled. The RULER SH/MH QA subtasks reuse the HotpotQA checkpoint at inference time (no additional training), following Sorokin et al. (2026). Q-RAG baselines are reproduced using the hyperparameters reported by Sorokin et al. (2026); the stability comparison in Section 4.3 additionally sweeps learning rate and effective batch size as described there.

Stability experiments. The training-stability comparison in Section 4.3 (Figures 5 and 6) uses the same recipe as "BabiLong QA3" above with an extended schedule matched to Q-RAG's reported setting: effective batch 64 (32×2 grad accumulation), lr=1.5×10^{-5}, 50K steps, linear LR schedule with 1000 warmup steps, and weight decay 0.1.

E FACT-LEVEL METRICS.

Let S_{gt} be the set of ground-truth supporting chunk indices and S_{pred} the set of chunks returned by the retriever. Following (Sorokin et al., 2026), our Fact-EM metric is defined as

$$\text{Fact-EM} = \begin{cases} 1, & \text{if } S_{\text{gt}} \subseteq S_{\text{pred}}, \\ 0, & \text{otherwise.} \end{cases}$$

Fact-EM gives full credit whenever the prediction covers all ground-truth chunks, even if it also contains additional, irrelevant ones. It does not require the two sets to be exactly equal.