SIMPLE LINKS, STRONG RETRIEVAL: LOCAL ENTITY EXPANSION FOR MULTI-HOP EVIDENCE RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieving complete supporting evidence from document collections remains a central challenge for knowledge-intensive language systems, particularly when answering a question requires connecting information across multiple sources. Similarity-based retrieval can overlook passages whose relevance depends on intermediate evidence, while graph-based and iterative approaches often introduce query decomposition, global propagation, or repeated reasoning to recover these connections. We present SCOPE, an entity-guided retrieval framework that addresses this challenge through a single-step local expansion and evidence-selection design. SCOPE turns precomputed passage–entity links into access paths beyond the initial dense ranking, then combines semantic, lexical, and entity-based signals with evidence-set selection to identify supporting passages. This modular design enables complementary evidence discovery without query decomposition or global graph propagation. On the established multi-hop QA benchmarks MuSiQue, 2Wiki, and HotpotQA, SCOPE outperforms leading graph-based baselines, including BrowseNet and HippoRAG 2, in both evidence retrieval and downstream question answering in our main comparison with a shared MiniLM backbone. On LoCoMo, however, graph-free retrieval with selection performs better, highlighting a limitation of entity-guided expansion in conversational memory. These results demonstrate that a simple local-expansion design can deliver competitive retrieval and answer quality alongside more elaborate retrieval pipelines.

1 INTRODUCTION

Recent advances in foundation models, exemplified by GPT-6, Gemini 3.8 Flash, and Claude Opus 5.5, have expanded the capabilities of language systems in complex reasoning, multimodal understanding, and tool use (OpenAI, 2026; Google DeepMind, 2026; Anthropic, 2026). These advances, however, do not eliminate a fundamental bottleneck: access to current, domain-specific, and verifiable knowledge beyond the information encoded during training. In knowledge-intensive applications, the evidence needed for a reliable answer is often distributed across documents, making its discovery and integration a central challenge. This challenge is especially pronounced in multi-hop reasoning, where a passage's relevance may depend on its relationship to other evidence rather than its direct similarity to the question. Effective retrieval must therefore recover complementary evidence across documents, supplying the connections that allow powerful reasoning models to produce grounded answers (Figure 1).

Existing methods address these evidence dependencies through iterative reasoning or graph-based retrieval. IRCoT alternates retrieval with intermediate reasoning (Trivedi et al., 2023); HippoRAG 2 combines a passage–phrase graph, LLM triple filtering, and Personalized PageRank (Jiménez Gutiérrez et al., 2025); BrowseNet constructs dependency graphs over decomposed queries to guide passage exploration (Pavan Kumar S et al., 2026). These approaches connect evidence beyond direct query matching, but introduce additional reasoning stages, graph operations, or query-specific structures into retrieval. This motivates a simpler design: can the connections needed for multi-hop evidence retrieval be exposed through local passage–entity links, without query decomposition, global graph propagation, or repeated online reasoning? Such a design would offer a more direct way to obtain complementary evidence while keeping the retrieval procedure simple.

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.

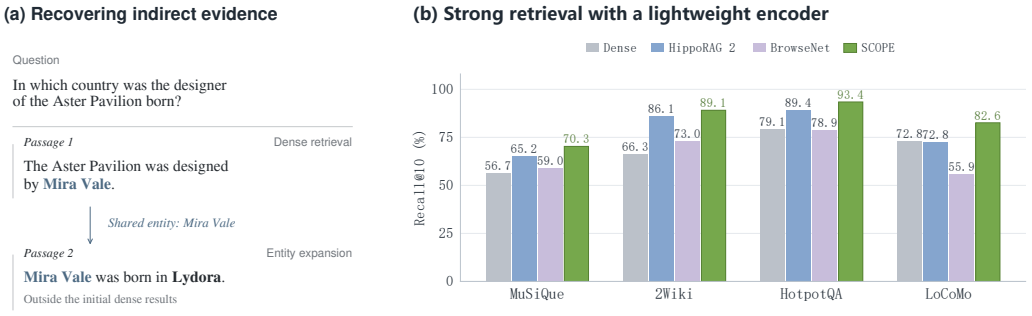

Figure 1: **Simple entity links support strong evidence retrieval.** (a) An illustrative example shows how a passage retrieved by dense similarity reveals a shared entity that connects to complementary evidence outside the initial retrieved set. (b) R@10 on the MuSiQue, 2Wiki, HotpotQA, and LoCoMo evaluation pools. All shown methods use MiniLM (22M) as the passage retrieval encoder under our local configurations; other indexing and retrieval components remain method-specific.

We introduce SCOPE, a retrieval framework built on a simple intuition: a retrieved passage can serve not only as evidence, but also as a clue to evidence elsewhere. A shared entity provides a natural point of connection, linking passages that describe different aspects of the same person, place, or event. Following these connections allows retrieval to move beyond what the question explicitly mentions and uncover complementary information across documents. SCOPE realizes this idea through a single local expansion over precomputed passage–entity links (Figure 2). The resulting candidates bring together direct query matches and passages discovered through their connections to the initial evidence. A subsequent selection stage combines semantic, lexical, and entity-based signals, then selects supporting evidence by considering the candidate passages jointly. This division gives each stage a clear role: expansion broadens access to potentially useful information, while selection determines what to return. The design requires neither query decomposition nor global graph propagation.

Experiments on MuSiQue, 2WikiMultihopQA, and HotpotQA demonstrate that SCOPE's simple design delivers strong multi-hop retrieval and question answering. In our main comparison, SCOPE outperforms leading graph-based systems, including BrowseNet and HippoRAG 2, in both evidence retrieval and downstream QA across all three benchmarks. The gains therefore extend beyond finding more relevant passages to providing better evidence for answering questions. These results show that local entity connections, coupled with effective selection, can match more elaborate retrieval pipelines on cross-document multi-hop tasks. Evaluation on the long-conversation benchmark LoCoMo reveals a different outcome: graph-free retrieval with selection provides stronger retrieval and QA performance. Together, these findings establish both the strength and the scope of SCOPE—a simple, effective design for the evaluated multi-hop tasks, with a clear boundary in conversational memory.

Our contributions are:

A simple framework for connected evidence retrieval. We introduce SCOPE, which uses local entity links to discover complementary evidence and a separate selection stage to identify supporting passages, without query decomposition or global graph propagation.

Strong multi-hop performance with an identifiable expansion benefit. SCOPE surpasses the evaluated leading graph-based baselines in both retrieval and QA on three multi-hop benchmarks in our main comparison with a shared MiniLM backbone. Matched controls further establish that entity-guided candidate expansion contributes beyond selection alone.

A clear boundary across retrieval settings. We show that the benefits of entity expansion depend on the evidence structure: it improves retrieval on the evaluated cross-document multi-hop tasks, while matched dense candidates and graph-free hybrid retrieval are stronger on LoCoMo.

2 Related Work

Dense and multi-hop retrieval. Dense passage retrieval learns query–passage matching (Karpukhin et al., 2020), while Contriever studies contrastive pretraining and GTR examines scaling dual encoders (Izacard et al., 2022; Ni et al., 2022). MiniLM provides a compact Transformer backbone, while Sentence-BERT supports efficient sentence-level semantic representations (Wang et al., 2020; Reimers & Gurevych, 2019); our encoder analysis also uses BGE, GTE-Qwen2, and NV-Embed-v2 (Xiao et al., 2023; Li et al., 2023; Lee et al., 2025). Multi-hop retrieval additionally conditions access on previously retrieved evidence: MDR trains a dense retriever for successive hops (Xiong et al., 2021), and Baleen condenses retrieved context across hops (Khattab et al., 2021). GAR generates auxiliary query contexts, whereas IRCoT alternates reasoning and retrieval (Mao et al., 2021; Trivedi et al., 2023). SCOPE uses one expansion from fixed dense seeds followed by selection over the resulting pool.

Graph-based access to evidence. PathRetriever learns reasoning paths over the Wikipedia graph (Asai et al., 2020), making expansion from previously accessed passages a close antecedent. SCOPE instead constructs passage–entity incidences and applies a single local expansion without learning a path policy. HippoRAG 2 uses query matching, LLM triple filtering, and Personalized PageRank; BrowseNet uses decomposed subqueries to guide query-specific exploration over an entity-linked graph of chunks (Jiménez Gutiérrez et al., 2025; Pavan Kumar S et al., 2026). G-Retriever combines prize-collecting Steiner-tree selection (Goemans & Williamson, 1995) with a trained graph–language interface (He et al., 2024). Our PCST adaptation and other baseline settings appear in Appendix A.2. GraphRAG builds community summaries for corpus-level questions, while LightRAG combines graph indexing with dual-level retrieval (Edge et al., 2024; Guo et al., 2024). SCOPE returns ranked original passages with no connected-subgraph requirement.

Fusion and Evidence Selection. BM25 provides lexical matching (Robertson & Zaragoza, 2009); RRF combines rankings without score calibration (Cormack et al., 2009). Instructed LLM rerankers improve passage ordering (Sun et al., 2023).

Conversational memory. LoCoMo evaluates very long-term conversational memory (Maharana et al., 2024). MemGPT manages memory tiers, Mem0 extracts and consolidates persistent memories, A-Mem dynamically links and updates notes, and Zep maintains temporal knowledge graphs (Packer et al., 2023; Chhikara et al., 2025; Xu et al., 2025; Rasmussen et al., 2025).

3 Method

SCOPE retrieves evidence by using initial query matches to expose related passages that may be difficult to find directly. Figure 2 illustrates the workflow: an offline passage–entity index supports local candidate expansion, followed by three-view rank fusion and evidence-set selection. We refer to fusion with preselection and the final selector together as the *evidence-selection pipeline*. This separation allows expansion to supply complementary candidates while selection determines which passages support the query.

3.1 Offline Passage–Entity Index

The offline index connects passages through the entities they mention. For a passage corpus $\mathcal{D}$, we construct a bipartite incidence graph $G = (\mathcal{D} \cup \mathcal{E}, \mathcal{I})$, where $\mathcal{E}$ contains canonical entities and $(d, e) \in \mathcal{I}$ links passage d to an extracted entity e. These links allow retrieval to move from an initial passage to others mentioning the same entity, without constructing a query-specific graph.

An LLM extracts entities and generates a summary σ_d for each passage. Entities include people, organizations, places, concepts, events, and other mentioned items. Extracted names are merged by case-insensitive matching or, failing that, by nearest entity-name embedding cosine similarity of at least 0.90 (Appendix A.2). After passage deduplication and entity normalization, each passage has an entity set $\mathcal{E}(d)$, and each entity has a passage list $\mathcal{P}(e) = \{d : e \in \mathcal{E}(d)\}$ with document frequency $\mathrm{df}(e) = |\mathcal{P}(e)|$.

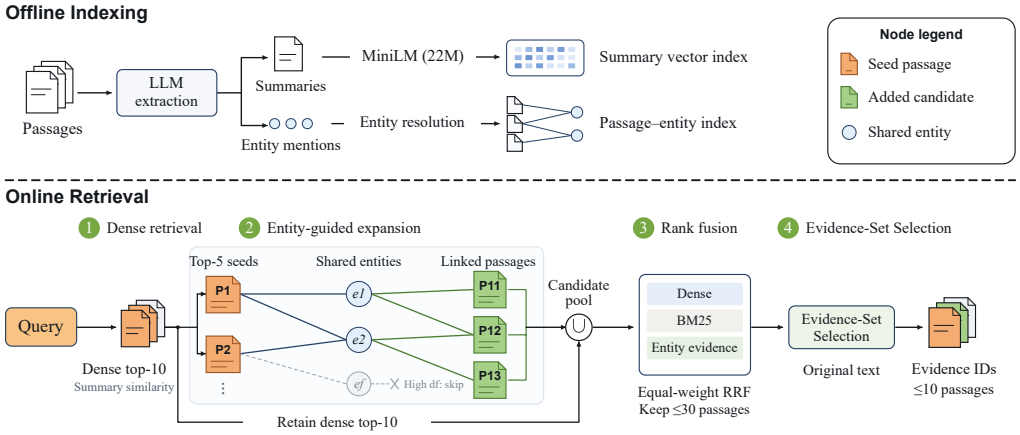

Figure 2: **Overview of SCOPE.** Offline indexing builds summary vectors and passage–entity links while retaining original passage text. Online retrieval uses the dense top five as seeds to recover additional candidates through shared entities, and unions these candidates with the dense top ten. Three-view RRF selects at most 30 passages for an evidence-set selection stage, which returns up to ten evidence IDs. The schematic highlights entity-linked passages outside the initial dense results; original text is used for BM25 and selection.

We also store the original text x_d and a normalized summary embedding $\mathbf{v}_d = f(\sigma_d)/\|f(\sigma_d)\|_2$. The main MiniLM configuration inherits summary embeddings from this joint indexing procedure. Dense retrieval uses these vectors.

3.2 Entity-Guided Candidate Expansion

For a query q, dense retrieval provides an initial entry point into the index. We encode $\mathbf{v}_q = f(q)/\|f(q)\|_2$ and score passages by $s_{\text{dense}}(q, d) = \mathbf{v}_q^\top \mathbf{v}_d$. Let $D_k(q)$ denote the top-k passages. The pool starts with $D_{k_d}(q)$, and the highest-ranked passages $S(q) = D_{k_s}(q)$ serve as seeds for expansion. Their entities provide the connections to additional evidence, so no query-entity extraction is needed.

Frequent entities can connect many passages with little relevance to the question. We limit these connections using a corpus-dependent frequency cap:

$$\tau = \max(\tau_{\min}, \lfloor \text{Percentile}_p(\{\text{df}(e) : e \in \mathcal{E}\}) \rfloor). \tag{1}$$

For each eligible entity in a seed, we add its linked passages to form

$$C(q) = D_{k_d}(q) \cup \bigcup_{s \in S(q)} \bigcup_{\substack{e \in \mathcal{E}(s) \\ \text{df}(e) \le \tau}} \mathcal{P}(e). \tag{2}$$

We call $C(q)$ the *graph candidate pool*; newly added passages are $C(q) \setminus D_{k_d}(q)$. Expansion follows one passage–entity–passage bridge and stops, retaining the original dense candidates alongside the additions. The pool size follows the eligible links rather than a fixed budget: on 2Wiki its median is 43 (IQR 32.75–59; further statistics and a bound appear in Appendix A.2).

We use $k_s = 5$, $k_d = 10$, $p = 99.5$, $\tau_{\min} = 10$, RRF constant $\kappa = 60$, and selector budget $b = 30$, fixed across datasets under the development protocol in Section 4.1.

3.3 Three-View Rank Fusion

The expanded pool contains passages connected to the seeds, but their usefulness still depends on the query. We combine three views to form a manageable candidate pool for the selector. Dense

query–summary similarity captures semantic relevance, BM25 over original text captures lexical matches, and entity evidence measures connections to the seeds:

$$s_{\text{ent}}(d; S(q)) = \sum_{s \in S(q)} \sum_{\substack{e \in \mathcal{E}(d) \cap \mathcal{E}(s) \\ \text{df}(e) \le \tau}} \log \frac{|\mathcal{D}|}{\text{df}(e)}. \tag{3}$$

The inverse-frequency weight emphasizes more specific connections. Each eligible shared entity contributes once per seed containing it, including the self-match $s = d$ when the candidate is itself a seed. Expansion and entity scoring use the same frequency cap.

Reciprocal rank fusion (RRF; Cormack et al., 2009) combines these complementary views without requiring their score scales to be calibrated:

$$s_{\text{RRF}}(q, d) = \sum_{v \in \{\text{dense,BM25,ent}\}} \frac{1}{\kappa + r_v(q, d)}, \qquad d \in C(q), \tag{4}$$

where r_v is a one-based rank within $C(q)$. Dense and BM25 scores are computed against the indexed corpus and restricted to this pool before ranking. Ties receive distinct consecutive ranks in NumPy sorting order, without a dense-similarity tie-break (Appendix A.2). Preselection keeps $B(q) = \text{Top}_{\min(b, |C(q)|)}(C(q); s_{\text{RRF}})$, bounding the number of passages passed to the selector rather than their total token length.

3.4 Evidence-Set Selection

We propose a selector that uses local entity connections to compose a complementary evidence set. Given the question, candidate passages, and their entity connections, it evaluates each candidate relative to the evidence already selected. Passage text is used to assess whether a shared entity links complementary facts, whether the candidate supplies a missing reasoning step, and whether it duplicates existing support. Selection therefore targets evidence completeness within the reader's passage budget.

Appendix A.2 specifies how invalid IDs and incomplete responses are handled. When evidence-set selection is omitted, *SCOPE w/o selection* returns the RRF top ten from the same offline index, completing retrieval without query-time LLM calls.

4 Experiments

4.1 Experimental Setup

Datasets and baselines. We evaluate on subsets of MuSiQue (Trivedi et al., 2022), 2WikiMultihopQA (Ho et al., 2020; 2Wiki), HotpotQA (Yang et al., 2018), and LoCoMo (Maharana et al., 2024). The first three test evidence retrieval for multi-hop QA; LoCoMo tests retrieval from conversation histories. We reserve MuSiQue development questions for configuration choices and transfer the core retrieval hyperparameters without per-dataset retuning. Appendix A provides pool sizes, development/test separation, deduplication, and eligibility rules.

We compare SCOPE with lexical and dense retrievers, graph-free hybrid retrieval, and the structure-augmented methods G-Retriever (He et al., 2024), BrowseNet (Pavan Kumar S et al., 2026), and HippoRAG 2 (Jiménez Gutiérrez et al., 2025). The main structure-augmented comparisons share a MiniLM passage backbone; stronger encoders are examined separately. G-Retriever is a PCST adaptation on SCOPE's graph. BrowseNet and HippoRAG 2 retain their own indexing components, with local adaptations documented in Appendix A.2. MiniLM dense and hybrid controls reuse SCOPE's summary vectors, including their shared indexing cost.

Shared evaluation protocol. SCOPE's extraction and selection use `gpt-4o-mini`, as do the shared QA reader and LoCoMo judge. Every QA condition supplies five original passages to the same reader and prompt, so the comparison evaluates the usefulness of the retrieved evidence. We report passage recall (R@k), span-answer EM/F1 for the multi-hop datasets, and judged accuracy (JA) for LoCoMo. Retrieval averages weight the four datasets equally; QA averages cover the

three EM/F1 datasets. Scores are percentages and differences are percentage points (pp). Paired confidence intervals and sign tests accompany the comparisons; LoCoMo also uses conversation-level inference (Efron, 1979; Dror et al., 2018; Appendix B.2). Reader settings and prompts appear in Appendices A.2 and E.

We distinguish comparisons of complete systems from comparisons that isolate candidate source. The latter hold candidate count, three-view fusion, and any selector fixed. The graph-free hybrid control instead removes both entity links and entity-evidence scoring from candidate generation and fusion. This distinction lets us assess both the practical pipeline and the contribution of expansion within it.

4.2 Retrieval and Downstream Question Answering

Table 1: Retrieval performance across four benchmarks. The best score in each column is **bold** and the second best is underlined. Scores are percentages; Average is the unweighted mean over all four datasets. Graph-augmented methods and SCOPE use the MiniLM passage backbone; Hybrid uses MiniLM; dense retrievers use the encoders shown. Stronger-encoder pairs appear in Table 6; –: unavailable.

Retriever	MuSiQue		2Wiki		HotpotQA		LoCoMo		Average	
	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10
Lexical retrieval										
BM25	43.71	51.26	65.75	70.78	71.88	85.70	78.52	84.61	64.96	73.09
Dense retrievers										
Contriever	46.60	54.27	56.53	63.20	74.26	82.15	61.56	74.39	59.74	68.50
GTR (T5-base)	48.38	56.74	68.18	71.18	72.67	79.40	67.68	76.30	64.22	70.90
MiniLM (summary)[a]	48.74	56.66	61.98	66.30	70.60	79.13	62.35	72.84	60.92	68.73
GTE-Qwen2 (1.5B)[b]	58.14	66.73	69.93	74.40	83.90	91.37	75.85	84.34	71.95	79.21
Structure-augmented RAG										
G-Retriever (PCST)†	48.42	57.48	62.93	70.23	72.46	80.46	64.49	75.15	62.07	70.83
BrowseNet	49.32	59.02	66.35	73.00	69.33	78.87	45.13	55.87	57.53	66.69
HippoRAG 2	57.25	65.24	81.58	86.08	81.46	89.35	62.97	72.77	70.82	78.36
Retrieval with selection										
Hybrid + selection	–	65.86	–	74.60	–	**94.12**	–	**91.64**	–	81.56
SCOPE (ours)	**60.98**	**70.29**	**85.55**	**89.10**	**89.62**	93.43	**78.96**	82.60	**78.78**	**83.86**

a: MiniLM embeds LLM summaries; this shared indexing cost is nonzero and not separately metered. *b*: GTE input/query-template provenance differs across datasets (Appendix C.2). †: PCST adaptation on the SCOPE graph. NV-Embed-v2 appears in Table 6.

Table 2: Answer generation with a shared span reader and top-five passages on MuSiQue, 2Wiki, and HotpotQA. Scores are percentages; Average covers these three EM/F1 datasets. LoCoMo uses JA and is excluded from Average. Best scores are **bold**; second best are underlined.

Retriever	MuSiQue		2Wiki		HotpotQA		LoCoMo	Average	
	EM	F1	EM	F1	EM	F1	JA	EM	F1
MiniLM-L6-v2 (22M)	17.13	30.87	30.50	36.12	39.19	51.82	80.67	28.94	39.60
G-Retriever (PCST)†	16.38	30.06	30.20	36.67	41.53	53.70	81.97	29.37	40.14
BrowseNet	20.63	33.62	32.50	39.23	41.84	53.75	70.54	31.66	42.20
HippoRAG 2	24.75	38.76	40.10	48.41	48.83	61.79	81.58	37.89	49.66
SCOPE (ours)	**32.50**	**46.32**	**43.20**	**51.89**	**55.51**	**69.59**	**86.81**	**43.74**	**55.93**

† PCST retrieval adapted to the SCOPE graph. CoT results and paired QA comparisons are in Appendix B.

SCOPE combines a simple retrieval design with consistently strong multi-hop performance. Across MuSiQue, 2Wiki, and HotpotQA, it outperforms BrowseNet, HippoRAG 2, and G-Retriever on both retrieval metrics and downstream QA in our main comparison (Tables 1 and 2). This advantage holds across all three benchmarks with the same core retrieval hyperparameters, supporting the

transferability of the design beyond the dataset used for configuration. A single local expansion followed by evidence selection is thus sufficient to surpass the evaluated, more elaborate graph-based pipelines in these settings. With a shared terminal selector, paired bootstrap comparisons find no significant R@10 or QA F1 differences between SCOPE and HippoRAG 2 on MuSiQue and 2Wiki with either MiniLM or NV-Embed-v2 (Appendix C.4, Table 23).

The retrieval results highlight the strength of this design. SCOPE achieves the highest average R@5 and R@10 among the methods in Table 1, improving average R@10 over MiniLM dense retrieval by 15.1 pp. Matched encoder comparisons in Table 6 show that SCOPE improves over dense retrieval across MiniLM, BGE-large, GTE-Qwen2, and NV-Embed-v2. Its advantage extends to the strong graph-free hybrid system with selection: SCOPE achieves statistically significant R@10 gains on MuSiQue and 2Wiki, while the small difference favoring hybrid on HotpotQA is not statistically significant (Table 20 in the appendix). These results demonstrate that the pipeline remains competitive against both stronger dense encoders and a hybrid alternative with selection.

Better retrieval also translates into better answers. SCOPE achieves the highest EM and F1 on every multi-hop benchmark in Table 2, rather than gaining only on an aggregate score. All methods use the same reader, prompts, and number of input passages, so this consistent advantage reflects the usefulness of the evidence supplied to answer generation. The joint improvements in recall and QA show that SCOPE both discovers supporting passages and assembles evidence that the reader can use effectively (paired QA comparisons appear in Appendix B.1).

The strongest benefits arise in cross-document multi-hop QA. On LoCoMo, graph-free retrieval with selection performs better, identifying conversational memory as a boundary of the current design (Section 4.5). This contrast clarifies where local entity connections are most useful and motivates adapting the candidate source to the retrieval setting.

4.3 What Does Entity Expansion Contribute Beyond Selection?

Table 3: **Component ablations.** MiniLM R@10 (%).

Configuration	MuS	2Wiki	Hot	LoCo
SCOPE (full)	70.29	89.10	93.43	82.60
Matched dense	65.58	71.88	88.35	86.50
w/o fusion	68.93	85.70	92.21	80.65
w/o freq. cap	70.21	84.45	93.27	90.40
w/o selection	62.54	80.30	88.93	77.77

Matched dense replaces expanded candidates under the same fusion. No-cap is diagnostic (Appendix C).

Table 4: **Retrieval-gain decomposition.** MiniLM R@10 changes (pp).

Configuration	MuS	2Wiki	Hot	LoCo
Fusion: dense	+3.64	+3.65	+5.19	+6.73
Graph pool − dense	+2.25	+10.35	+4.61	-1.79
Selection: graph pool	+7.75	+8.80	+4.50	+4.82
Selection: dense	+5.29	+1.93	+4.03	+6.93

Fusion: dense adds fusion to matched dense retrieval; Graph pool − dense changes candidate source under fusion. Selection rows add selection to the named fusion-only path.

MuS: MuSiQue; Hot: HotpotQA; LoCo: LoCoMo. Effects are conditional, not additive. The 2Wiki candidate-source gain has 95% CI $[9.03, 11.73]$ pp (Appendix A.3).

We examine the roles of candidate expansion, three-view fusion, entity-frequency filtering, and evidence-set selection through component ablations and matched controls (Tables 3 and 4). The full pipeline performs best among these variants on all three multi-hop datasets. The results support a design in which discovering complementary evidence and selecting it effectively work together.

Entity expansion provides a better starting point for evidence selection. Replacing expanded candidates with the same number of dense candidates reduces retrieval performance across all three multi-hop datasets, even when fusion and selection are unchanged. The advantage also appears before evidence-set selection, showing that local entity links expose useful evidence that deeper dense retrieval alone can miss. Expansion therefore contributes through the passages it discovers, rather than merely increasing the number of candidates considered.

Fusion and selection make this broader evidence pool useful. Removing three-view fusion lowers performance on every dataset, supporting the combination of semantic relevance, lexical matching, and entity connections. Removing selection produces the largest decline among the selection ablations, highlighting the value of considering candidate passages jointly. Together, these findings explain the

division of labor in SCOPE: expansion creates opportunities to recover complementary evidence, and selection turns those opportunities into stronger retrieval results.

Frequency filtering helps keep expansion focused. Its removal lowers performance on the three multi-hop datasets, most noticeably on 2Wiki, consistent with frequent entities introducing less selective connections. LoCoMo instead benefits from removing the cap, suggesting that recurring entities can play a different role in conversational evidence. This contrast identifies the frequency cap as a useful control over expansion, with room to adapt its strength to the evidence structure.

Table 5: **SCOPE versus graph-free hybrid retrieval.** Same selector and matched candidate counts. R@10 is a percentage; differences and paired 95% CIs are pp.

Dataset	Hybrid	SCOPE	Δ	95% CI
MuSiQue	65.86	70.29	+4.43	$[+2.50, +6.40]$
2Wiki	74.60	89.10	+14.50	$[+12.70, +16.40]$
2Wiki / NV	82.95	95.80	+12.85	$[+11.50, +14.30]$
HotpotQA	94.12	93.43	-0.69	$[-2.00, +0.60]$
LoCoMo	91.64	82.60	-9.05	$[-10.50, -7.70]$

MiniLM except the NV-Embed-v2 row. Hybrid uses dense+BM25 RRF. No-selection controls and full statistics appear in Appendix C.1. LoCoMo QA: 89.55% hybrid versus 86.81% SCOPE (categories 1–4).

The graph-free hybrid control further tests whether the gains persist with a stronger candidate source and the same selector. SCOPE retains an advantage on MuSiQue and 2Wiki, including the 2Wiki comparison with NV-Embed-v2 (Table 5). On 2Wiki, evidence trajectories show that expansion recovers gold passages absent from the initial dense results and that many survive selection into the reader's context; SCOPE also supplies complete gold evidence more often than hybrid retrieval with selection (Appendix C.5). These findings connect the component-level gains to the purpose of the design: local entity links help bring otherwise missed supporting evidence within the reader's reach.

4.4 Full-Pipeline Gains Across Encoders

Table 6: Retrieval performance across encoders (R@10, %). Gain is the unweighted mean improvement of SCOPE over Dense with the same encoder across the four datasets (pp).

Method	Encoder	MuSiQue	2Wiki	HotpotQA	LoCoMo	Gain
Dense	MiniLM (22M)	56.66	66.30	79.13	72.84	–
SCOPE	MiniLM (22M)	70.29	89.10	93.43	82.60	**+15.12**
Dense	BGE-large (335M)	65.86	75.68	93.64	79.80	–
SCOPE	BGE-large (335M)	74.74	94.73	97.88	86.57	**+9.74**
Dense	GTE-Qwen2 (1.5B)	68.20	75.30	91.79	85.44	–
SCOPE	GTE-Qwen2 (1.5B)	77.18	93.98	97.83	89.54	**+9.45**
Dense	NV-Embed-v2 (7B)	78.23	80.65	98.09	87.54	–
SCOPE	NV-Embed-v2 (7B)	83.17	95.80	98.57	90.52	**+5.89**

SCOPE's retrieval gains extend beyond MiniLM. Across the tested encoder families, the full pipeline improves over the corresponding dense retriever on every dataset (Table 6), with only a small, statistically inconclusive gain for NV-Embed-v2 on HotpotQA. This consistency shows that the design remains useful as passage representations become stronger. The matched-source analysis on 2Wiki further supports an expansion benefit across these encoders before evidence-set selection (Appendix C.2). Local entity connections can therefore complement stronger semantic representations, extending the value of SCOPE beyond a lightweight backbone.

With a shared NV-Embed-v2 backbone, SCOPE remains competitive with BrowseNet and HippoRAG 2 (Appendix C.3).

4.5 When Entity Expansion Helps

The boundary results suggest that the value of an entity link depends on whether it points to complementary evidence. In the 2Wiki diagnostic, gold passage pairs are much more likely to share an eligible entity than random pairs. This makes a shared entity a useful clue for reaching missing

evidence. The question-type results reinforce this interpretation: expansion helps bridge-comparison and compositional questions, but is less effective for comparison questions whose evidence can be retrieved independently (Appendix C.5). The benefit is therefore clearest when answering the question requires following a connection beyond the initial matches.

LoCoMo helps explain the other side of this pattern. Its entity links are less selective in the multi-evidence diagnostic: recurring entities connect many conversational passages without necessarily identifying the evidence needed for a particular question. Hybrid and matched dense retrieval with selection perform better in this setting, while the filtering variants reveal a different response to the frequency cap (Appendix B.3). These findings point to a useful design principle: expansion benefits from connections that distinguish supporting evidence, rather than simply from having more connections. SCOPE's strength on the evaluated multi-hop tasks follows this principle, while conversational memory motivates more adaptive candidate discovery.

4.6 Cost–Quality Tradeoffs

Table 7: **Cost–quality tradeoff on 2Wiki.** Indexing covers 6,119 passages; online costs are per query. R@10 is a percentage.

Method	Offline indexing		Online retrieval/query		R@10
	Tokens (M)	Time (h)	Tokens	Time (s)	
Dense (MiniLM)	Shared[a]	–	0	–	66.30
HippoRAG 2	7.02	0.79	2,893	1.16	86.08
Hybrid RRF	Shared[a]	–	0	–	72.02
Hybrid + selection	Shared[a]	–	3,350	–	74.60
SCOPE w/o selection	9.54	1.67	0	0.011	80.30
SCOPE (full index)	9.54	1.67	4,247	0.99	89.10

a: shared LLM summary generation, not separately metered; not zero offline cost. –: unavailable. QA is excluded; zero LLM tokens do not mean zero local computation. Full-SCOPE/HippoRAG 2 timings use the same machine and day. Complete accounting, additional methods, and slim projections appear in Appendix D, Table 26.

SCOPE supports multiple cost–quality operating points with the same offline index (Table 7). Without selection, it requires no query-time LLM calls while retaining an R@10 of 80.30 on 2Wiki, substantially above MiniLM dense retrieval. Adding selection raises R@10 to 89.10. This separation allows online computation to be adjusted without rebuilding the evidence connections; complete cost accounting appears in Appendix D.

5 Conclusion

We presented SCOPE, a simple entity-guided retrieval pipeline that connects initial dense matches to complementary evidence through precomputed passage–entity links. A single local expansion, followed by rank fusion and evidence-set selection, retrieves supporting passages without query decomposition or global graph propagation. Our experiments show improved retrieval over the corresponding dense baselines across the tested encoders and stronger downstream QA than the evaluated structure-augmented baselines with a shared MiniLM backbone. Matched comparisons further establish that candidate expansion contributes beyond selection alone. These findings support local entity links as a practical mechanism for cross-document multi-hop retrieval, while the LoCoMo results highlight the need for different candidate strategies in conversational memory. Separating expansion from evidence selection also allows retrieval without online LLM calls when selection is omitted. Future work will explore query-adaptive expansion and richer semantic connections to extend this simple design to a broader range of evidence dependencies.

Reproducibility Statement

Section 3 specifies the retrieval algorithm; Appendices A and E describe the evaluation pools, preprocessing, hyperparameters, fallbacks, and extraction, selector, reader, and judge prompts. Appendix A.1 separates the runs used for headline results, diagnostics, and cost measurements. Upon acceptance, we will release the research code, configurations, and shareable per-question results, with scripts for reconstructing evaluation subsets and generating tables. Dataset redistribution will follow the original licenses. The API model identifier is `gpt-4o-mini`; a dated model snapshot was not recorded, limiting exact replication of model outputs.

Ethics Statement

This study evaluates retrieval on existing QA and conversational-memory benchmarks. Entity extraction and linking can propagate errors or biases in source passages, and conversational retrieval can expose sensitive information when applied to personal histories. Deployment therefore requires appropriate consent, access controls, and data minimization. LLM-based judging may introduce systematic errors, particularly because the reader and judge use the same model family; we report it as judged accuracy rather than human-verified correctness. Released benchmark-derived material will respect the source datasets' licenses and redistribution restrictions.

AI Use Statement

We use `gpt-4o-mini` for passage summarization and entity extraction, evidence-set selection, answer generation, and LoCoMo answer judging. We also used Codex to assist manuscript drafting and editing, literature lookup, reference and table formatting, figure editing, manuscript and table-production code, and consistency checks of reported results against the underlying output files. Beyond these uses, generative AI was not used to design or provide feedback on the method or experiments, implement the research method or data-processing code, interpret results, or clean or reformat datasets. The authors have reviewed all AI-assisted work and take responsibility for the final text, claims, code, and results.

References

Anthropic. Introducing Claude Opus 5.5, 2026. URL https://www.anthropic.com/claude-opus-5-5.

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://arxiv.org/abs/1911.10470.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. arXiv:2504.19413, 2025. URL https://arxiv.org/abs/2504.19413.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759, 2009. doi: 10.1145/1571941.1572114.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of ACL*, 2018. URL https://aclanthology.org/P18-1128/.

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

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979. doi: 10.1214/aos/1176344552.

Michel X. Goemans and David P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995. doi: 10.1137/S0097539793242618.

Google DeepMind. Introducing Gemini 3.8 Flash and 3.8 Flash Cyber, 2026. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/3-8-flash-and-3-8-flash-cyber/.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. LightRAG: Simple and fast retrieval-augmented generation. arXiv:2410.05779, 2024. URL https://arxiv.org/abs/2410.05779.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-Retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *Advances in Neural Information Processing Systems*, 2024.

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

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. URL https://openreview.net/forum?id=jKN1pXi7b0.

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21497–21515, 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Vladimir Karpukhin, Barlas Oğuz, 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*, pp. 6769–6781, 2020. 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 *Advances in Neural Information Processing Systems*, 2021. URL https://arxiv.org/abs/2101.00436.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-Embed: Improved techniques for training LLMs as generalist embedding models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c4bf73386022473a652a18941e9ea6f8-Abstract-Conference.html.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. arXiv:2308.03281, 2023. URL https://arxiv.org/abs/2308.03281.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024. URL https://aclanthology.org/2024.acl-long.747/.

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. doi: 10.18653/v1/2021.acl-long.316. URL https://aclanthology.org/2021.acl-long.316/.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y. Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9844–9855, 2022. doi: 10.18653/v1/2022.emnlp-main.669. URL https://aclanthology.org/2022.emnlp-main.669/.

OpenAI. Safety overview: GPT-6 Astra, 2026. URL https://openai.com/index/safety-overview-gpt-6-astra/.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. arXiv:2310.08560, 2023. URL https://arxiv.org/abs/2310.08560.

Pavan Kumar S, Kiran Kumar Nakka, C Vamshi Krishna Reddy, Divyateja Pasupuleti, Prakhar Agarwal, Harpinder Jot Singh, Anshu Avinash, and Nirav Pravinbhai Bhatt. BrowseNet: Graph-based associative memory for contextual information retrieval. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=2q5CugVPoK.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory. arXiv:2501.13956, 2025. URL https://arxiv.org/abs/2501.13956.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. 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. 3982–3992, 2019. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14918–14937, 2023. doi: 10.18653/v1/2023.emnlp-main.923. URL https://aclanthology.org/2023.emnlp-main.923/.

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. 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 *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037, 2023. doi: 10.18653/v1/2023.acl-long.557. URL https://aclanthology.org/2023.acl-long.557/.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2002.10957.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-Pack: Packaged resources to advance general chinese embedding. arXiv:2309.07597, 2023. URL https://arxiv.org/abs/2309.07597.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oğuz. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2009.12756.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/19909c36f51abc4856b4560aff3d36d6-Abstract-Conference.html.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of EMNLP*, 2018. URL https://aclanthology.org/D18-1259/.

A Evaluation Protocol and Implementation

Data and eligibility. The MuSiQue passage pool is deduplicated across the 1,000 questions; 800 questions are scored after reserving 200 for development, without overlapping decomposition identifiers between the two sets. The passage corpus contains 11,510 items. 2Wiki uses its 1,000-question, 6,119-passage pool. LoCoMo retrieval is conversation-local: supporting evidence is mapped to passage identifiers within the relevant conversation, and questions without mapped gold evidence are not scored. The resulting retrieval denominator is 1,977; QA further excludes category 5, giving 1,531 questions. These filters and denominators apply before averaging scores. Questions within one conversation share a retrieval corpus, so per-question uncertainty does not capture all between-conversation variability.

HotpotQA evaluation protocol. HotpotQA uses 944 unique question identifiers over 9,221 passages. The source 1,000-row sampling file contains 53 repeated identifiers (56 excess rows); only the first occurrence contributes to each main retrieval, QA, ablation, and encoder-matrix score. Deduplication is shared by the table validators. Extraction uses `gpt-4o-mini`, with fallback extraction for 22 passages. Four SCOPE selection refusals fall back to fusion; BrowseNet records ten graph-retrieval fallbacks. Three shared-reader refusals are scored as empty answers. These fallback cases are included in the reported scores.

Reader and judge configuration. Both the QA reader and LoCoMo judge use `gpt-4o-mini` with temperature 0. This is the configured model identifier; no dated model snapshot is recorded. In the main-table span-reader conditions for MuSiQue, 2Wiki, and HotpotQA, the reader receives the top five original passages and the single-span prompt reproduced in Appendix E. EM and token F1 use the official MuSiQue normalization. LoCoMo uses the detailed-answer generation and binary judging templates in the same appendix; the judge returns a JSON label, `CORRECT` or `WRONG`, with format differences and partial but accurate answers accepted as specified in the prompt. Judged accuracy is the fraction labeled `CORRECT`. The LoCoMo column of the main QA table includes only categories 1–4 (1,531 questions), using the same prompts as the LoCoMo evaluation procedure.

Scoring and inference. Recall uses the overlap of unique retrieved and gold passage IDs divided by the number of gold passages for each question. QA normalizes answer strings for EM and token F1; the reader receives original text from a common top-five retrieval interface. The LoCoMo judge compares predicted answers against the reference under the evaluation prompt. It is a separate role from the extractor, selector, and answer reader, even when those roles use the same model family. Scores for MuSiQue, 2Wiki, and LoCoMo are computed from per-question results. HotpotQA retrieval, QA, ablation, and encoder means are recomputed from per-question results after deduplication; reported paired statistics come from the experiment summaries.

A.1 Run Provenance and Repeatability

Table 8 distinguishes experiment groups that share a method name but use separate model executions or input settings. Main retrieval and QA means are computed from per-question outputs; the supplementary hybrid, cluster, encoder-interval, and evidence-trajectory analyses are available as aggregate statistics. The release will include a table-to-run ledger with result identifiers and content hashes. Within-run bootstrap intervals quantify question-sampling uncertainty; they are not estimates of run-to-run variance.

Two LoCoMo dense-reader/judge executions differ by 0.50 pp in judged accuracy, with 95.8% agreement between question-level labels. This pair characterizes one evaluation condition. The API configuration uses temperature zero, but an undated model identifier and service-side execution can still affect output reproducibility.

Independent index rebuilds on 2Wiki. We rebuild the index from scratch three times on the same 6,119 passages, regenerating summaries and entities, and evaluate retrieval and QA on the same 1,000 questions with MiniLM and fixed hyperparameters. Extraction, selection, and the reader use `gpt-4o-mini` at temperature zero. Each run compares SCOPE with Dense and count-matched dense candidates using that run's vectors; Dense also varies because the summary text is regenerated.

Table 8: Experiment groups and calculation inputs. Q: per-question outputs; A: aggregate statistics; M: mixed inputs; L: metering logs or estimates. Scores are percentages. Data codes describe the inputs to the reported calculations, not publication availability.

Group	Data	Table / configuration
Main retrieval	Q	Table 1; fixed cohorts, 2Wiki SCOPE R@10 89.10.
Component ablations	M	Table 3; aggregate values except HotpotQA means recomputed after deduplication. 2Wiki without selection: 80.30.
Main QA and sign tests	Q	Table 2, Tables 12–15; 2Wiki F1 51.89. Bootstrap intervals also use experiment summaries.
QA selection ablation	Q	Appendix C; separate 2Wiki reader execution, full F1 51.76.
Independent rebuilds	Q	Table 9; three new 2Wiki indices, each with retrieval and QA on the same 1,000 questions.
Hybrid control	A	Tables 5, 19, 20; separate execution, 2Wiki graph without selection 80.25.
LoCoMo diagnostic	A	Table 18; matched dense R@10 86.52 versus 86.50 in Table 3; no cap 90.19 versus 90.40.
Encoder matrix	M	Table 6; means use per-question outputs and experiment summaries. Input configurations are specified in Appendix C.2.
Encoder intervals	A	Table 17; paired question-level bootstrap summaries.
Candidate source	M	Table 21; GTE official-template rerun; dedicated BGE instruction run, dense R@10 75.82 versus 75.68 without the instruction.
Cluster/evidence analyses	A	Tables 16, 25, 24; conversation statistics and evidence-stage counts.
Cost measurements	L	Table 7; 2Wiki token probe R@10 89.18; retrieval reference 89.10. LoCoMo indexing uses a separate probe.

Table 9: Three independent 2Wiki rebuilds with MiniLM on the same 1,000 questions. Scores are percentages; differences and sample standard deviations are in percentage points.

Metric	Run 1	Run 2	Run 3	Mean ± SD
Retrieval R@10				
Dense	66.90	66.48	66.40	66.59 ± 0.27
Matched dense + selection	71.75	72.58	72.20	72.18 ± 0.41
SCOPE w/o selection	80.33	80.28	80.30	80.30 ± 0.03
SCOPE	89.38	89.05	88.85	89.09 ± 0.26
Δ: SCOPE − Dense	22.48	22.58	22.45	22.50 ± 0.07
Δ: SCOPE − matched dense	17.63	16.48	16.65	16.92 ± 0.62
Δ: graph − matched dense (no selection)	10.75	10.55	10.60	10.63 ± 0.10
Answer F1				
Dense	36.11	37.17	36.66	36.65 ± 0.53
SCOPE	52.85	51.83	52.04	52.24 ± 0.54
Matched dense + selection	44.42	43.33	43.53	43.76 ± 0.58
Δ: SCOPE − Dense	16.75	14.66	15.38	15.59 ± 1.06
Δ: SCOPE − matched dense	8.43	8.50	8.51	8.48 ± 0.04

Table 9 reports each run and the mean and sample standard deviation across the three runs, not a bootstrap confidence interval.

SCOPE R@10 is $89.09 \pm 0.26\%$, close to the main run's 89.10%. The within-run F1 gain over count-matched dense candidates averages 8.48 ± 0.04 pp; this repeated-run estimate is used for the candidate-source QA claim in the main text. SCOPE's repeated-run F1 is $52.24 \pm 0.54\%$; the main QA table continues to report its separate single-run F1 of 51.89%. Similar aggregate recall does not establish identical retrieved lists or general repeatability across datasets. Each rebuild evaluation includes six selector content-filter fallbacks to RRF order and 16–17 reader refusals across the evaluated configurations, with refused answers scored zero.

A.2 Implementation Details

Passage deduplication and entity normalization. Indexing deduplicates passages by text hash, then extracts summaries and entity mentions. Each mention is matched to an existing entity by case-insensitive name lookup; otherwise, the nearest entity-name embedding is merged when its cosine similarity is at least 0.90. Unmatched mentions create new entities. The indices also contain facets and other relation types, but the retrieval procedure follows only passage–entity incidences.

Ranking ties and candidate counts. Graph candidates are enumerated in lexicographic passage-ID order before fusion. Each view is sorted by `numpy.argsort(-scores)` with its default quicksort, assigning distinct one-based positions even to tied scores; no average-rank or dense-similarity tie-break is applied. Default quicksort is not stable, so passage-ID order is not a guaranteed tie-break. RRF preselection likewise uses descending `argsort`. The 2Wiki candidate-pool statistics in Section 3.2 use the main run's per-question counts before top-30 selection, with linear interpolation for quartiles. Across 1,000 questions, the mean is 46.35 passages, the median is 43 (IQR 32.75–59), and the range is 10–125. The pool is a subset of the indexed corpus; its size is at most $\min(|\mathcal{D}|, k_d + \sum_{s \in S(q)} \sum_{e \in \mathcal{E}(s):\mathrm{df}(e) \le \tau} (\mathrm{df}(e) - 1))$, because each seed is already in $D_{k_d}(q)$.

Baseline adaptations. The PCST baseline uses SCOPE's passage–entity graph rather than G-Retriever's trained GNN–language-model interface. BrowseNet retains its entity-linking modules and uses ten candidate subgraphs; its LoCoMo query decomposition transfers the MuSiQue demonstrations. These settings share a MiniLM passage backbone with SCOPE but do not make the complete index builders identical. The main comparisons report locally evaluated retrieval outputs, not scores copied from the original papers.

SCOPE configuration. The retrieval algorithm accesses the entity-to-passage incidence relation (`involves_entity`), normalized passage vectors, entity memberships, and original text. MiniLM represents each passage by a 384-dimensional vector. The seed count is five, the dense fallback depth is ten, the fusion constant is 60, the maximum selector pool is 30, the retrieved list length is ten, and the reader depth is five. The frequency rule uses percentile 99.5 with a minimum cap of ten; in the 2Wiki pool it yields a cap of 15. The percentile is fixed at 99.5 in the main configuration; the no-cap diagnostic removes this constraint.

Fusion selects the selector pool; all four datasets present the selected candidates in descending dense query–summary similarity. Invalid LLM identifiers are removed; selection stops when the selector reports nothing missing; remaining slots follow RRF order (Appendix E.3). This fallback can make presentation order relevant even when the same candidate set is presented.

The original extractor supplies summaries and entities as part of a larger prompt that also requests facets. The main scores are obtained from indices made with that prompt. A storage simplification removes unused facets and temporal, evolutionary, and causal relations while retaining summaries and entity incidences. In the MuSiQue pruning check, R@2/5/10 in the configuration without query-time LLMs are unchanged; 799 of 800 questions have identical per-question recall records, with one R@1 tie difference.

A.3 Candidate-Count Controls

The fusion-only control uses a dense candidate pool with the same pre-fusion passage count as the entity-expanded pool, applies the same three scoring views, and returns the top ten. Thus its

comparison to graph-plus-fusion isolates candidate source without an online retrieval LLM. On MuSiQue, 2Wiki, HotpotQA, and LoCoMo, the control with selection likewise applies fusion to the count-matched dense pool before selection.

For MuSiQue, the matched dense control uses the same candidate count before preselection. It takes that many dense-ranked candidates, applies the same three-view RRF preselection to select the selector pool, and then uses the same evidence-set selector. Both configurations retain entity-evidence scoring; only the source of additional candidates changes. The matched control achieves R@10 of 65.58% versus SCOPE's 70.29%, a paired SCOPE-minus-control gain of 4.71 pp (95% bootstrap CI $[3.00, 6.50]$ pp, sign-test $p = 4.6 \times 10^{-7}$). The dense-candidate replacement in Table 3 and its figure use this matched-candidate result with the same three-view RRF preselection. The corresponding selection increment in Table 4 is 5.29 pp relative to the dense-plus-fusion mean. Prompt lengths and token costs can differ at the same candidate count.

The 2Wiki no-selection candidate-source gain is 10.35 pp (95% paired bootstrap CI $[9.03, 11.73]$ pp), computed from the 1,000 aligned question-level differences with 10,000 resamples and seed 20260924. This interval measures question-level uncertainty, separately from the index-rebuild SD.

B Additional Retrieval and QA Comparisons

Table 10: Additional retrieval results: R@2 (%). Main retrieval results appear in Table 1; NV-Embed-v2 R@10 appears in Table 6.

Method	MuSiQue	2Wiki	HotpotQA	LoCoMo
BM25	34.47	55.28	54.77	69.46
Contriever	34.69	44.83	56.30	46.06
GTR (T5-base)	37.79	61.10	58.85	55.07
Dense (MiniLM)	37.22	51.70	56.36	48.66
Dense (NV-Embed-v2)	52.49	69.03	82.26	66.28
GTE-Qwen2-1.5B	45.83	59.83	70.71	60.36
G-Retriever (PCST adaptation)	37.41	51.83	56.36	48.51
BrowseNet	37.00	51.85	51.80	32.77
HippoRAG 2	41.93	65.68	63.56	47.89
SCOPE	45.21	72.60	79.71	69.31

GTE and NV input protocols follow Appendix C.2. G-Retriever is the PCST adaptation.

Table 11: 2Wiki QA with the additional chain-of-thought reader (%). The span-reader results appear in Table 2.

Method	EM	F1
MiniLM-L6-v2 (22M)	39.10	44.00
G-Retriever (PCST)†	39.80	45.71
BrowseNet	44.40	50.63
HippoRAG 2	<u>51.90</u>	<u>60.13</u>
SCOPE (ours)	**55.30**	**63.96**

The supplementary retrieval table includes Contriever, GTR, GTE-Qwen2-1.5B, and NV-Embed-v2. NV-Embed-v2's dense R@10 exceeds SCOPE/MiniLM by 7.9 pp on MuSiQue and 4.9 pp on LoCoMo. The distinction between top-ten recall and top-five answer context also prevents interpreting either metric as a substitute for the other.

B.1 Paired QA Comparisons

Tables 12–15 supplement Table 2. We report 95% intervals from 10,000 paired question-level bootstrap resamples and the separate sign-test p-values; the latter are not bootstrap p-values. The comparisons use span F1 on MuSiQue, 2Wiki, and HotpotQA, and judged accuracy on LoCoMo

categories 1–4. The two-sided exact sign tests exclude tied question-level scores and test equal win and loss probabilities among non-tied pairs. Tests are unadjusted for multiple comparisons. LoCoMo resampling is by question, not by conversation; its intervals do not capture uncertainty across conversation clusters.

Table 12: MuSiQue paired QA comparisons supporting Table 2 (span F1; $n = 800$). Differences and confidence intervals are in percentage points.

Comparison	Δ	95% CI	p (sign test)
SCOPE − G-Retriever	+16.26	$[+13.50, +19.20]$	1.6×10^{-24}
SCOPE − all-MiniLM-L6-v2	+15.44	$[+12.80, +18.30]$	1.8×10^{-23}
SCOPE − BrowseNet	+12.70	$[+9.80, +15.70]$	2.2×10^{-13}
SCOPE − HippoRAG 2	+7.56	$[+4.80, +10.40]$	1.4×10^{-6}

Table 13: 2Wiki paired QA comparisons supporting Table 2 (span F1; $n = 1000$). Differences and confidence intervals are in percentage points.

Comparison	Δ	95% CI	p (sign test)
SCOPE − MiniLM	+15.77	$[+12.89, +18.69]$	3.3×10^{-29}
SCOPE − G-Retriever	+15.22	$[+12.30, +18.10]$	2.9×10^{-25}
SCOPE − BrowseNet	+12.66	$[+9.70, +15.60]$	1.2×10^{-16}
SCOPE − HippoRAG 2	+3.48	$[+0.70, +6.20]$	1.3×10^{-2}

Table 14: HotpotQA paired QA comparisons supporting Table 2 (span F1; $n = 944$). Differences and confidence intervals are in percentage points.

Comparison	Δ	95% CI	p (sign test)
SCOPE − all-MiniLM-L6-v2	+17.78	$[+15.10, +20.50]$	1.4×10^{-29}
SCOPE − BrowseNet	+15.85	$[+13.10, +18.60]$	3.1×10^{-24}
SCOPE − G-Retriever	+15.90	$[+13.20, +18.70]$	4.6×10^{-24}
SCOPE − HippoRAG 2	+7.80	$[+5.30, +10.30]$	1.7×10^{-8}

Table 15: LoCoMo paired QA comparisons supporting Table 2 (JA; $n = 1531$). Differences and confidence intervals are in percentage points.

Comparison	Δ	95% CI	p (sign test)
SCOPE − BrowseNet	+16.26	$[+13.80, +18.70]$	7.6×10^{-39}
SCOPE − all-MiniLM-L6-v2	+6.14	$[+4.30, +8.00]$	1.3×10^{-10}
SCOPE − HippoRAG 2	+5.23	$[+3.20, +7.30]$	9.7×10^{-7}
SCOPE − G-Retriever	+4.83	$[+2.90, +6.70]$	8.6×10^{-7}

B.2 Conversation-Level and Encoder-Matrix Inference

The supplementary LoCoMo retrieval analysis treats ten conversations as clusters. For the 95% CI, cluster bootstrap resamples ten conversations with replacement, pools all question-level paired differences in each draw, and takes the 2.5th and 97.5th percentiles of the pooled means. This statistic weights conversations by their question counts. In contrast, the one-sample t-test against zero uses ten conversation-specific mean differences with equal conversation weights. The interval and test therefore use different weighting schemes. SCOPE exceeds Dense in all ten, with a reported conversation-level 95% interval of $[7.10, 13.50]$ pp for the recall difference. Against the matched dense control, only two conversations have positive differences and the interval is $[-5.40, -2.20]$ pp.

Table 16: LoCoMo retrieval comparisons at the conversation level. Confidence intervals are in R@10 percentage points; positive counts are over ten conversations.

Comparison	Positive conversations	95% CI	p
SCOPE − Dense	10/10	$[+7.10, +13.50]$	0.002
SCOPE − Matched dense	2/10	$[-5.40, -2.20]$	0.004
SCOPE − BM25	4/10	$[-4.70, +0.80]$	0.299
No LLM: graph - matched dense	2/10	$[-3.10, -0.50]$	0.026
No LLM: graph - dense	10/10	$[+3.90, +6.20]$	< 0.001

CIs resample conversations and weight by question count; the t-test uses ten equally weighted conversation means. See Appendix B.2.

Against BM25, four are positive and the conversation-level t-test gives $p = 0.30$: no difference is detected.

Table 17: Paired SCOPE-minus-dense R@10 differences and 95% confidence intervals for the encoder matrix (pp). HotpotQA uses 944 unique questions.

Encoder	MuSiQue	2Wiki	HotpotQA	LoCoMo
MiniLM	$+13.64$ $[+12.00, +15.30]$ $+8.87$	$+22.80$ $[+21.30, +24.30]$ $+19.05$	$+14.30$ $[+12.80, +15.80]$ $+4.24$	$+9.76$ $[+8.40, +11.10]$ $+6.77$
bge-large	$[+7.30, +10.50]$ $+8.98$	$[+17.60, +20.50]$ $+18.68$	$[+3.30, +5.20]$ $+6.04$	$[+5.60, +7.90]$ $+4.10$
GTE-Qwen2	$[+7.40, +10.40]$ $+4.94$	$[+17.20, +20.10]$ $+15.15$	$[+5.00, +7.20]$ $+0.48$	$[+3.10, +5.10]$ $+2.97$
NV-Embed-v2	$[+3.50, +6.50]$	$[+13.80, +16.60]$	$[-0.05, +1.01]$	$[+2.00, +4.00]$

Paired question-level intervals; LoCoMo cluster inference appears in Table 16. GTE provenance follows Table 6.

All sixteen encoder-matrix point differences are positive and fifteen paired 95% intervals exclude zero (Table 17). The exception is NV-Embed-v2 on HotpotQA: $\Delta = 0.48$ pp, CI $[-0.05, 1.01]$ pp, $p = 0.12$. HotpotQA inference uses 944 unique questions; LoCoMo matrix intervals remain question-level and are distinguished from the conversation-cluster results above.

B.3 LoCoMo QA Diagnostics

Table 18: LoCoMo diagnostic variants. Scores are percentages; differences are pp. Paired differences compare each variant with default SCOPE on JA categories 1–4.

Variant	R@10	JA (all)	JA (cat. 1–4)	Δ	p
SCOPE default	82.60	87.00	86.81	–	–
Matched dense + selection	86.52	88.72	89.16	+2.35	3.2×10^{-3}
No frequency cap	90.19	89.12	89.55	+2.74	5.3×10^{-4}
Hybrid + selection	91.64	88.77	89.55	+2.74	1.0×10^{-3}

Diagnostic runs; the primary QA metric is judged accuracy on categories 1–4. Run groups are distinguished in Appendix A.1.

The main results use the default configuration. Removing expansion, relaxing the frequency cap, and using graph-free hybrid retrieval improve judged accuracy on categories 1–4 in this diagnostic run. The no-cap and hybrid variants have the same aggregate score (89.55%). Two runs of the same Dense configuration differ by 0.50 pp in judged accuracy, with 95.8% per-question agreement. Diagnostic retrieval scores (e.g., 86.52% without expansion and 90.19% without the cap) retain their own run rather than replacing the older ablation-run scores.

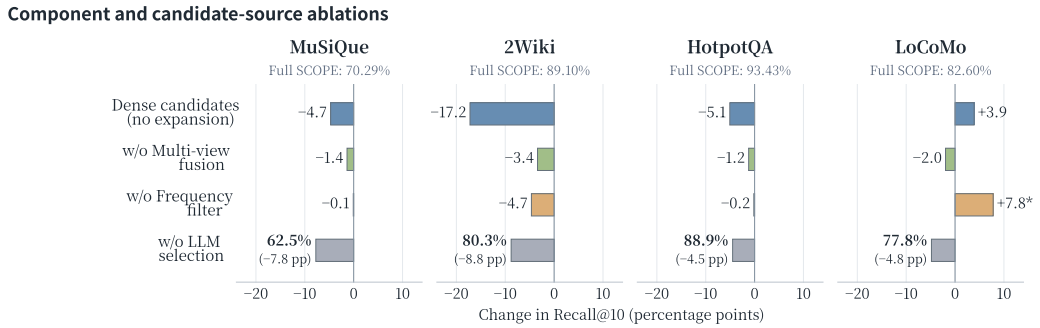

Figure 3: **Conditional effects of components and candidate source.** Changes in R@10 relative to full SCOPE with MiniLM on MuSiQue, 2Wiki, HotpotQA, and LoCoMo (Table 3). The first row replaces entity-expanded candidates with count-matched dense candidates; the remaining rows remove individual components. The last row also reports absolute recall without evidence-set selection: 62.5%, 80.3%, 88.9%, and 77.8%, respectively, with zero online retrieval LLM calls. Offline LLM indexing remains necessary. The LoCoMo no-filter improvement is a test-set diagnostic.

C Additional Ablations and Encoder Controls

Figure 3 visualizes the component and candidate-source effects reported numerically in Table 3.

Operational definitions of component ablations. Without multi-view fusion, we keep the candidate set unchanged and select at most 30 passages by descending dense similarity before applying the same evidence-set selector. The candidate budget is unchanged; only the preselection rule is replaced. Removing the frequency filter sets the document-frequency cap to the corpus passage count, lifting the restriction in both places where it acts: the edge-following rule of candidate expansion and the shared-entity contributions to the entity-evidence scoring view. On LoCoMo, this expands the candidate pool to approximately 96% of conversation passages, providing context for the observed 7.80 pp recall gain. The diagnostic changes both expansion and entity scoring.

Removing a single fusion view has a smaller effect than removing the whole fusion stage in most conditions. BM25 is particularly useful on LoCoMo; its removal reduces R@10 by 2.4 pp. In contrast, removing BM25 or the dense view slightly improves 2Wiki retrieval, and the corresponding QA comparisons do not show a statistically reliable benefit. The result supports a multi-view design in the tested pipeline, not a claim that each view improves every dataset.

In a separate 2Wiki reader run, removing evidence-set selection reduces F1 from 51.76 to 38.04, a within-run loss of 13.72 pp. For the candidate-source comparison, we report the three independent rebuilds in Appendix A.1: SCOPE exceeds count-matched dense candidates under shared selection by 8.48 ± 0.04 pp in F1. These repeat-run statistics do not replace the single-run means in the main QA table.

C.1 Graph-Free Hybrid Control

Dense and BM25 each retrieve candidates independently; their rankings are interleaved until the graph configuration's per-question candidate budget is reached. Equal-weight two-view RRF selects up to 30 candidates for the same selector used by SCOPE. The hybrid configuration without selection returns the RRF top ten without a retrieval LLM. Neither hybrid candidate stage reads graph features or entity incidences; the selector's link field uses the corpus entity index for all systems (Appendix E.3). Candidate counts are matched, but token lengths and measured costs are not established as equal. SCOPE retains its three-view RRF preselection, so this is a graph-free system comparison, not a one-factor ablation of graph expansion.

Table 19: **Graph-free hybrid controls under matched candidate counts.** All scores are R@10 as percentages. Bold compares the two pipelines with selection within each row.

Dataset / encoder	Dense	No selection		Same selector		SCOPE − Hybrid + selection	Graph − Hybrid, no selection
		Hybrid	Graph + RRF	Hybrid	SCOPE		
MuSiQue / MiniLM	56.66	60.48	62.54	65.86	**70.29**	+4.43	+2.06
2Wiki / MiniLM	66.30	72.02	80.25	74.60	**89.10**	+14.50	+8.23
2Wiki / NV	80.65	77.98	89.68	82.95	**95.80**	+12.85	+11.70
HotpotQA / MiniLM	79.13	86.92	88.93	**94.12**	93.43	-0.69	+2.01
LoCoMo / MiniLM	72.84	87.02	77.77	**91.64**	82.60	-9.05	-9.24

Hybrid combines dense and BM25 with two-view RRF; SCOPE uses three views. Candidate counts are matched. NV denotes NV-Embed-v2. Paired statistics are in Table 20.

Table 20: Paired differences for the graph-free hybrid controls in R@10 percentage points. Brackets report paired 95% confidence intervals from 10,000 paired question-level bootstrap resamples; p-values use a separate sign test.

Dataset / encoder	SCOPE − Hybrid + selection			Graph − Hybrid, no selection	
	Δ	95% CI	p	Δ	p
MuSiQue / MiniLM	+4.43	$[+2.50, +6.40]$	1.8×10^{-5}	+2.06	1.3×10^{-2}
2Wiki / MiniLM	+14.50	$[+12.70, +16.40]$	1.1×10^{-51}	+8.23	9.3×10^{-28}
2Wiki / NV	+12.85	$[+11.50, +14.30]$	2.7×10^{-76}	+11.70	4.1×10^{-81}
HotpotQA / MiniLM	-0.69	$[-2.00, +0.60]$	5.3×10^{-1}	+2.01	6.4×10^{-3}
LoCoMo / MiniLM	-9.05	$[-10.50, -7.70]$	8.4×10^{-33}	-9.24	5.0×10^{-31}

Differences use paired observations before rounding. The HotpotQA no-selection CI is $[+0.60, +3.50]$ pp. LoCoMo intervals are question-level; 0/10 conversations favor SCOPE over hybrid with selection.

For HotpotQA, full SCOPE uses the Table 1 score of 93.43% and hybrid achieves 94.12%, yielding a difference of -0.69 pp (paired 95% CI $[-2.00, +0.60]$ pp, sign-test $p = 0.53$). The difference is not significant. Without selection, graph candidates improve over hybrid RRF by 2.01 pp (paired 95% CI $[+0.60, +3.50]$ pp, $p = 6.4 \times 10^{-3}$). The 2Wiki/MiniLM no-LLM graph score is 80.25 in this control run, versus the 80.30 ablation score. All other values likewise retain this comparison's reported run. The supplementary analyses use aggregate experiment statistics, distinguished from the per-question main-table calculations in Appendix A.1. The reported confidence intervals and sign tests follow the protocol above, without a multiple-comparison correction. LoCoMo's 0/10 positive conversations provides a descriptive check across conversations, not a replacement for cluster-aware uncertainty estimates.

C.2 Encoder Inputs and Candidate-Source Gains

Table 21: 2Wiki candidate-source comparison with shared fusion and no retrieval-time LLM calls. Scores are R@10 (%); Δ is graph minus matched dense (pp).

Encoder	Matched dense + fusion	Graph + fusion	Δ
MiniLM-L6	69.95	80.30	+10.35
bge-large	78.00	85.88	+7.88
GTE-Qwen2-1.5B	76.98	85.68	+8.70
NV-Embed-v2	81.30	89.68	+8.38

BGE uses a query instruction here; its full-pipeline matrix run omits that instruction. Differences are computed before rounding.

The encoder matrix uses `all-MiniLM-L6-v2` (22M, 384d), `BAAI/bge-large-en-v1.5` (335M, 1024d), `Alibaba-NLP/gte-Qwen2-1.5B-instruct` (1.5B, 1536d), and `nvidia/NV-Embed-v2` (7B, 4096d). The graph topology is held fixed when swapping passage and query

vectors. Each experimental pair compares dense retrieval and SCOPE with identical passage vectors and query formatting within that run. The GTE checkpoint is the Qwen2-based instruction model documented in the official model card,[1] while Li et al. (2023) describes the original GTE framework.

Re-embedding supports both stored summaries and raw passage text, and the instruction runs use model-specific query formatting. BGE's 2Wiki full-pipeline run omits the query instruction (dense R@10 75.68), whereas the dedicated candidate-source experiment includes it (75.82). Table 21 preserves the latter run and computes each difference within its own configuration. These values should not be spliced into the full-pipeline table or presented as one universal instruction protocol. Appendix A.1 identifies the distinct experiment groups.

The MiniLM dense-retrieval baseline and graph-free Hybrid controls use the same summary vectors as SCOPE; the Hybrid BM25 view uses original passages. By contrast, the count-matched dense-candidate controls retain entity-evidence scoring, as specified in Appendix A.3. The Contriever, GTR, and GTE baseline runs with the embedding-model override in Table 1 embed original passages. On MuSiQue, a separate original-passage versus summary comparison differs in R@10 by 0.06 pp; the comparison covers only this corpus.

The GTE pairs in Table 6 use the official model query template. On MuSiQue, 2Wiki, and LoCoMo, the reruns preserve original-passage vectors and update query formatting; SCOPE improves R@10 over Dense by 8.98, 18.68, and 4.10 pp, respectively (question-level sign-test $p = 7.9 \times 10^{-27}$, 1.3×10^{-121}, and 1.9×10^{-14}). The 2Wiki GTE candidate-source control in Table 21 uses this same official-template run. The GTE baseline scores in Table 1 on these three datasets come from earlier runs whose exact query template was not recorded; they are not the Dense counterparts of the updated encoder pairs.

HotpotQA's encoder matrix uses summary embeddings and documented model-specific query templates, including the official GTE template. Its GTE Dense R@10 is 91.79, whereas the original-passage GTE baseline in Table 1 scores 91.37. Thus the GTE passage inputs differ across datasets even though each Dense–SCOPE pair is matched. Table 6 reports the four datasets separately without a pooled average. No original-passage NV-Embed-v2 retrieval aggregate is reported for HotpotQA, so the corresponding supplementary cells are marked –.

C.3 Shared Strong-Encoder Comparison Across Methods

Table 22 holds the NV-Embed-v2 encoder and instruction template fixed across five systems. Each system retains its own construction and retrieval mechanism; sharing an encoder does not equate the remaining components or their costs. The reported cohorts contain 800 MuSiQue, 1,000 2Wiki, and 1,977 LoCoMo questions, with per-question alignment across methods. The table reports aggregate scores and paired comparisons against BrowseNet and HippoRAG 2. Reported paired intervals include zero for both BrowseNet and HippoRAG 2 on MuSiQue and 2Wiki. The paired sign-test p-values against BrowseNet and HippoRAG 2, respectively, are 1.00 and 0.79 on MuSiQue, 0.63 and 0.74 on 2Wiki, and 3.5×10^{-30} and 8.0×10^{-3} on LoCoMo. Only the LoCoMo differences are significant; no multiple-comparison correction is applied. LoCoMo question-level tests do not account for conversation clustering. This strong-encoder system comparison was not evaluated on HotpotQA.

C.4 Shared-Selector Comparison and Interpretation

We pass HippoRAG 2's top-30 passages, in its original retrieval order, to the same terminal selector used by SCOPE, sharing the model, prompt, response parsing, and padding rules. Both systems use the same reader over the selected top-five passages. We evaluate MuSiQue and 2Wiki with MiniLM and NV-Embed-v2. Across all four settings, we find no statistically significant differences in mean R@10 or QA F1 under paired 95% bootstrap inference (Table 23).

These results support local evidence completion as a viable alternative to global graph propagation. SCOPE uses initially retrieved passages as anchors for recovering supporting evidence through local entity connections, followed by fusion and supporting-chain selection. The value of this design is demonstrated by the evidence trajectories and matched candidate-source controls: on 2Wiki/MiniLM,

[1] https://huggingface.co/Alibaba-NLP/gte-Qwen2-1.5B-instruct

Table 22: Retrieval with a shared NV-Embed-v2 encoder and instruction template. Each method retains its own graph construction and retrieval mechanism. R@10 is reported as a percentage; best scores are **bold** and second best are underlined within each dataset.

Dataset	Dense	G-Retriever	BrowseNet	HippoRAG 2	SCOPE	SCOPE − BrowseNet	SCOPE − HippoRAG 2
MuSiQue-800	78.23	78.31	82.87	82.81	**83.17**	+0.29	+0.35
2Wiki-1000	80.65	84.38	**96.25**	95.65	95.80	−0.45	+0.15
LoCoMo-1977	87.54	88.31	82.69	89.05	**90.52**	+7.83	+1.47

Aligned question cohorts. Last two columns: paired differences (pp). Statistical tests and inference details appear in Appendix C.3.

Table 23: R@10 and QA F1 with a shared terminal selector and reader. Scores are percentages. Confidence intervals are paired 95% bootstrap intervals for SCOPE minus HippoRAG 2 with selection, in percentage points.

Setting	System / comparison	R@10	QA F1
MuSiQue MiniLM	HippoRAG 2 + selection	71.77	46.20
	SCOPE	70.29	46.27
	Difference: 95% CI	$[-3.29, +0.33]$	$[-2.69, +2.77]$
2Wiki MiniLM	HippoRAG 2 + selection	88.62	53.17
	SCOPE	89.10	51.63
	Difference: 95% CI	$[-0.97, +1.95]$	$[-4.13, +1.09]$
MuSiQue NV-Embed-v2	HippoRAG 2 + selection	83.49	52.79
	SCOPE	83.17	53.73
	Difference: 95% CI	$[-1.98, +1.33]$	$[-1.37, +3.27]$
2Wiki NV-Embed-v2	HippoRAG 2 + selection	96.40	56.57
	SCOPE	95.80	56.06
	Difference: 95% CI	$[-1.50, +0.27]$	$[-3.03, +1.95]$

local expansion recovers 671 of the 906 gold-passage instances missed by dense top-10, and 586 reach the reader's top-five context; the corresponding counts for matched dense retrieval are 172 and 162 (Table 24). Thus, the contribution is a concrete mechanism for turning partial retrieval into more complete reader evidence, while achieving competitive final performance under a shared terminal selector. The shared-selector comparison does not isolate local versus global propagation because the systems retain different indices. It nevertheless identifies a practical trade-off: in this comparison on 2Wiki/MiniLM, SCOPE uses 4,236 online LLM tokens per query versus 7,629 for HippoRAG 2 with selection, although its offline index construction is more expensive. For a fixed corpus and a reused index, the offline construction cost per query decreases as query volume grows, while the cumulative online token savings relative to HippoRAG 2 with selection increase. The choice between the methods therefore depends on query volume, index-rebuild frequency, and application requirements.

C.5 Evidence Trajectories and Complete Coverage

Entity connectivity and question types. In the 2Wiki diagnostic, 41.2% of gold passage pairs share an entity after frequency filtering, versus 0.1% of random pairs. For LoCoMo, the same diagnostic includes 353 questions with at least two gold passages and samples random pairs within each conversation. With its frequency filter, gold and random pair sharing rates are 23.3% and 9.8%; without filtering, both are 100%. These rates concern the multi-evidence diagnostic subset, not all 1,977 retrieval questions. Graph-pool versus matched-dense gains after selection are 29.7 pp on bridge-comparison questions and 23.9 on compositional questions, versus -5.1 on comparison questions with independently retrievable evidence. These subgroup observations characterize where expansion helps; they are neither a learned routing policy nor a causal explanation on their own.

Table 24: Gold-passage instances missed by dense top-10, followed through both candidate sources. The same count budget and selection stages are used within each pair.

Stage	2Wiki / MiniLM		2Wiki / NV		HotpotQA / MiniLM	
	Graph	Matched	Graph	Matched	Graph	Matched
Missed by dense top-10	906	906	578	578	394	394
Entered candidates	671	172	510	197	278	191
After RRF preselection	644	166	497	170	275	184
Selected top-10	632	164	462	165	272	178
Reader top-5	586	162	401	151	259	174

Matched denotes count-matched dense retrieval. NV denotes NV-Embed-v2. All HotpotQA stages use the same index.

Table 25: Complete evidence coverage (All-gold@5), the fraction of questions whose reader-visible top five contain all gold passages. Scores are percentages; differences (pp) compare SCOPE with matched dense retrieval and selection.

Dataset	Dense	Matched dense + selection	Hybrid + selection	SCOPE	Δ	p
2Wiki / MiniLM	27.10	42.30	46.50	70.10	+27.80	7.5×10^{-58}
2Wiki / NV	51.00	64.90	60.30	78.70	+13.80	4.6×10^{-25}
HotpotQA / MiniLM	46.93	74.36	82.52	81.25	+6.89	1.2×10^{-6}

NV denotes NV-Embed-v2.

Table 24 traces gold evidence missed by dense top-10 through candidate inclusion, three-view RRF preselection, selection into the top ten, and the reader-visible top five. Counts are gold-passage instances, not counts of questions. The matched dense control follows the same selection stages. Expansion recovers 671 of 906 missed instances on 2Wiki/MiniLM (74%), 510 of 578 on 2Wiki/NV-Embed-v2 (88%), and 278 of 394 on HotpotQA/MiniLM (71%); 586, 401, and 259, respectively, reach the reader's top five. The corresponding matched dense candidate-recovery counts are 172, 197, and 191.

The failure subset contains questions with gold evidence missed by dense top-10 and no gold recovered by expansion: 128 questions on 2Wiki and 99 on HotpotQA. SCOPE All-gold@5 is necessarily zero on both: at least one gold passage is absent from its candidate pool. Count-matched dense retrieval instead achieves 19.50% and 31.30%, respectively. Because selection into these subsets depends on expansion failure, these comparisons do not estimate relative performance over all questions. Expansion recovery on comparison questions is only 3.4% and 8.8%, respectively, and SCOPE underperforms the matched dense control in those groups.

D Cost and Storage Accounting

Table 26 expands the main-text comparison in Table 7 with PCST, BrowseNet, and projected slim indexing; all shared entries retain the same measurements and retrieval scores. Offline extraction, online retrieval, passage embedding, and QA are distinct costs. Dense and Hybrid reuse checkpoint summary embeddings, so their offline LLM cost is shared summary generation rather than zero. Summary generation is bundled into the extraction prompt and has not been separately metered. On 2Wiki, a lean extractor producing summaries and entities uses 2.69M tokens; we report this as a reference cost for an alternative summary-and-entity extraction configuration, not as a cost bound or a measured allocation from the full extraction prompt. The alternative changes the prompt and summaries; its 2.69M tokens cannot be assigned to summary-only generation. Summary-only costs are not separately measured on any corpus. The Shared entries in the cost tables avoid charging the full graph construction to graph-free controls while acknowledging their summary-generation requirement. Tables 26, 27, 28, and 29 combine retrieval scores with separate cost measurements. SCOPE's 2Wiki online tokens come from an instrumented 1,000-query rerun (R@10 89.18, versus the main-run 89.10); LoCoMo indexing was measured with a separate cost probe. HippoRAG 2 tokens come from cached API usage, excluding its QA reader. The no-query-LLM variant retains

Table 26: **Cost–quality tradeoff on 2Wiki.** Offline cost is paid once for 6,119 passages; online cost is per query. R@10 is a percentage. SCOPE rows distinguish retrieval without selection, the full pipeline, and projected slim indexing. Full-SCOPE and HippoRAG 2 timings were measured on the same machine and day.

Method	Offline indexing		Online retrieval/query		R@10
	Tokens (M)	Time (h)	Tokens	Time (s)	
Dense (MiniLM)	Shared[a]	–	0	–	66.30
G-Retriever (PCST)[s]	9.54	1.67	0	0.10	70.23
BrowseNet	0.00	2.00[e]	466	0.47	73.00
HippoRAG 2	7.02	0.79	2,893	1.16	86.08
Hybrid RRF	Shared[a]	–	0	–	72.02
Hybrid + selection	Shared[a]	–	3,350	–	74.60
SCOPE w/o selection	9.54	1.67	0	0.011	80.30
SCOPE (full index)	9.54	1.67	4,247	0.99	89.10
SCOPE (slim projection)	8.74[e]	1.47[e]	4,247	0.99	89.10

a: shared summary generation, not separately metered. s: reused SCOPE index charged to PCST. e: estimated or extrapolated. –: unavailable. Accounting definitions are in Appendix D.

the full index cost. MuSiQue SCOPE online tokens estimate input prompt length only, whereas the 2Wiki and LoCoMo reports count API input and output tokens.

Table 26 uses same-day SCOPE/HippoRAG 2 query timings; API latency is environment-dependent.

PCST and BrowseNet timings are outside the same-machine, same-day SCOPE/HippoRAG 2 comparison. End-to-end answer-generation latency was not measured.

Structure versus encoder scale. On 2Wiki, SCOPE with a 22M MiniLM backbone reaches R@10 of 89.10, compared with 80.65 for dense retrieval using 7B NV-Embed-v2. Embedding the corpus on the same CPU node takes 0.7 minutes with MiniLM and 424.2 minutes with NV-Embed-v2, an approximately $606\times$ difference (Table 30). The full pipeline, including selection, therefore exceeds the quality obtained by scaling the dense encoder alone in this setting. The $606\times$ ratio concerns encoder-side embedding only; it excludes SCOPE indexing and online selection and is not a total-system cost reduction.

Slim indexing preserves the retrieval representation. Removing unused facets and temporal, evolutionary, and causal relations reduces one MuSiQue graph serialization from 198.8 to 40.7 MB (79.5%). In the pruning check without query-time selection, 799 of 800 questions have identical recall records; the remaining difference is an R@1 tie, and aggregate R@2/5/10 are unchanged (Appendix A.2). This supports removing unused stored structure while preserving the representations used for retrieval. Small-sample measurements project 8.4% fewer extraction tokens and 11.8% less indexing time: 8,744,051 tokens and 1.47 hours on 2Wiki. Table 26 therefore labels this row *slim projection*, retaining the full-index retrieval score as a reference. The alternative lean extractor changes the prompt and summaries; its lower costs do not describe the main results.

Time amortization does not imply monetary savings. Using the projected slim indexing time and the recorded same-day online latencies, cumulative indexing-plus-retrieval time for Q sequential queries is

$$T_{\text{slim}}(Q) = 1.47 \times 3600 + 0.99Q, \tag{5}$$

$$T_{\text{Hippo}}(Q) = 0.79 \times 3600 + 1.16Q. \tag{6}$$

The resulting break-even query count is

$$Q^* = \frac{(1.47 - 0.79) \times 3600}{1.16 - 0.99} \approx 14{,}400. \tag{7}$$

This is an illustrative projection from rounded timings, not a measured workload crossover. It assumes a fixed corpus, one index per method, stationary per-query latency, sequential processing, and no rebuilding; API latency and concurrency can change the result.

The token comparison has no such crossover: slim indexing requires 8.74M versus 7.02M tokens, and each query uses about 4,247 versus 2,893 tokens. Both differences remain positive for any nonnegative Q. At the experimental accounting rates of \$0.15 per million input tokens and \$0.60 per million output tokens, slim/HippoRAG 2 indexing costs are \$3.33/\$1.80 and retrieval costs are \$0.65/\$0.46 per 1,000 queries. Both monetary differences are also positive, so there is no monetary break-even under that tariff. These historical calculations exclude QA and local-compute charges and are not current price quotations.

Table 27: MuSiQue cost accounting. R@10 is a percentage; indexing tokens are in millions.

Method	Offline indexing		Online retrieval/query		R@10
	Tokens (M)	Time (h)	Tokens	Time (s)	
Dense (MiniLM)	Shared[a]	–	0	–	56.66
G-Retriever (PCST)[s]	18.08[e]	10.16	0	–	57.48
BrowseNet	0.00	–	448	–	59.02
HippoRAG 2	13.20[e]	–	2,900[e]	–	65.24
Hybrid RRF	Shared[a]	–	0	–	60.48
Hybrid + selection	Shared[a]	–	4,226	–	65.86
SCOPE w/o selection	18.08[e]	10.16	0	–	62.54
SCOPE (full index)	18.08[e]	10.16	∼4,744	0.96	70.29
SCOPE (slim projection)	16.56[e]	8.96[e]	∼4,744	0.96	70.29

∼: SCOPE online tokens estimate prompt characters divided by four, excluding completion tokens. Hardware/API settings are in Appendix D; other symbols follow Table 26.

Table 28: HotpotQA cost accounting on 9,221 passages. R@10 is a percentage; indexing tokens are in millions.

Method	Offline indexing		Online retrieval/query		R@10
	Tokens (M)	Time (h)	Tokens	Time (s)	
Dense (MiniLM)	Shared[a]	–	0	–	79.13
G-Retriever (PCST)[s]	16.18	4.91[c]	0	–	80.46
BrowseNet	0.00	2.28	286	2.08	78.87
HippoRAG 2	11.19	1.77	2,720	2.42	89.35
Hybrid RRF	Shared[a]	–	0	–	86.92
Hybrid + selection	Shared[a]	–	∼4,060	–	94.12
SCOPE w/o selection	16.18	4.91[c]	0	–	88.93
SCOPE (full index)	16.18	4.91[c]	∼4,060	∼1.81	93.43
SCOPE (slim projection)	14.82[e]	4.33[ec]	∼4,060	∼1.81	93.43

c: concurrent indexing with contention. SCOPE online costs are approximate allocations across two configurations. HippoRAG uses first-run latency. Other symbols follow Table 26.

Table 29: LoCoMo cost accounting. R@10 is a percentage; indexing tokens are in millions.

Method	Offline indexing		Online retrieval/query		R@10
	Tokens (M)	Time (h)	Tokens	Time (s)	
Dense (MiniLM)	Shared[a]	–	0	–	72.84
G-Retriever (PCST)[s]	1.61	0.27	0	–	75.15
BrowseNet	0.00	0.33	416	0.26	55.87
HippoRAG 2	1.16	0.12[c]	2,825	1.18[c]	72.77
Hybrid RRF	Shared[a]	–	0	–	87.02
Hybrid + selection	Shared[a]	–	∼5,789	–	91.64
SCOPE w/o selection	1.61	0.27	0	–	77.77
SCOPE (full index)	1.61	0.27	5,789	1.21[t]	82.60
SCOPE (slim projection)	1.48[e]	0.24[e]	5,789	1.21[t]	82.60

Indexing uses a separate probe. t: SCOPE latency covers 301 queries in two conversations on a different date. c: HippoRAG timings use a different compute cluster from SCOPE. Online tokens include input and output; other symbols follow Table 26.

Table 30: Encoder computation on 2Wiki (6,119 passages). All four measurements use CPU inference on the same compute node with 32 CPUs and 180 GB memory; downloading and loading models are excluded.

Encoder	Parameters	Dim.	Embed (min)	Relative	SCOPE R@10
all-MiniLM-L6-v2	22M	384	0.7	1×	89.10
bge-large-en-v1.5	335M	1024	13.5	19×	94.73
GTE-Qwen2-1.5B	1.5B	1536	81.9	117×	93.98
NV-Embed-v2	7B	4096	424.2	606×	95.80

Rounded ratios; encoder timings use a separate compute environment from Table 26. Quality values refer to Table 6; the GTE query-template rerun does not remeasure passage-embedding time.

E LLM PROMPTS

The following templates are reproduced from the indexing and evaluation code; doubled braces in the templates escape literal JSON braces during Python string formatting. Placeholders are filled with the input text chunk, question, original passage text, reference answer, or generated answer as indicated.

E.1 EXTRACTION

The following are the complete extraction user-message templates. Each uses the system message `Return strict JSON only.` The main extraction template also requests fields not used by retrieval. The fallback extracts only summaries and entities.

Main extraction `You are an information extraction engine.`

```
Task:
Extract structured memory data from the text chunk below.
Text chunk:
{chunk}
Return a single JSON object with exactly these top-level keys:
- episode_summary: string
- entities: array of objects
- facet_points: array of objects
- facets: array of objects
- temporal_info: array of objects
Schema details:
1) episode_summary
- A concise but comprehensive summary of ALL events and facts mentioned in
the chunk. Include specific details like names, dates, places, objects,
and quantities.
2) entities
- Each item must be:
{{
"name": string,
"entity_type": string
}}
- entity_type should be one of: "person", "organization", "place",
"concept", "event", "other".
- Keep names as they appear in the text whenever possible.
- Include specific items mentioned (books, foods, activities, pets, places
visited, etc.) as entities with type "concept" or "other".
3) facet_points
- Each item must be:
{{
"content": string,
"related_entity_name": string or null,
"timestamp_text": string or null
```

```
}}
- content should be atomic and factual.
- IMPORTANT: Be specific.  Include concrete details like exact names,
quantities, colors, and descriptions.
Good:  "Melanie made a cup in her pottery class"
Bad:  "Melanie does pottery"
Good:  "Caroline recommended the book Becoming Nicole to Melanie"
Bad:  "Caroline recommended a book"
Good:  "Melanie and her family saw the Perseid meteor shower during their
camping trip"
Bad:  "Melanie went camping"
- Extract every distinct fact as a separate facet_point.  Do not merge
multiple facts into one.
4) facets
- Each item must be:
{{
"theme":  string,
"facet_point_indices":  array of integers
}}
- facet_point_indices refers to zero-based indices in the facet_points
array.
5) temporal_info
- Each item must be:
{{
"subject":  string,
"time_expression":  string,
"normalized_time":  string or null,
"relation":  string
}}
- relation examples:  "before", "after", "during", "at".
- normalized_time should use ISO-8601 when explicit enough, otherwise
null.
- For relative time references (e.g., "yesterday", "last week"), use the
conversation timestamp shown in the chunk header to compute an absolute
date for normalized_time.
Rules:
- Output valid JSON only.  No markdown fences, no explanation text.
- If information is missing, use empty arrays or null values as
appropriate.
- Do not invent unsupported facts.
- Prefer extracting MORE facet_points with specific details over fewer
generic ones.
```

Fallback extraction `You are a lightweight information extraction engine.`

```
Task:
Extract only a concise episode summary and entity list from the text
chunk.
Text chunk:
{chunk}
Return a single JSON object with exactly these keys:
- episode_summary: string
- entities: array of objects
Each entities item must be:
{{
"name": string,
"entity_type": string
}}
Rules:
- entity_type should be one of: "person", "organization", "place",
"concept", "event", "other".
- Output valid JSON only. No markdown fences, no explanation text.
- If no entities are present, return an empty array.
```

E.2 Reader and Judge

Multi-hop QA reader (main span conditions) `You are answering a multi-hop question from retrieved passages. The passages carry the intermediate steps as well as the final fact; report only the final fact.`

```
Example
Question: When did the country that produced the film X declare war on
the US?
Wrong answer: Japan, December 8, 1941 (this lists the chain)
Right answer: December 8, 1941 (this answers what was asked)
Passages:
{passages}
Question: {question}
Reply with one span and nothing else - no chain, no list, no sentence, no
trailing period.
If the passages do not settle it, still give the single most likely span
from them.
Answer:
```

LoCoMo reader `You are a helpful assistant answering questions based on the provided context.`

```
Context:
```

```
{context}

Question: {question}

Instructions:

1. Answer the question using the provided context. Be specific and cite
concrete details from the context.

2. For time-related questions, follow these steps:

Step 1: Find the conversation date from the header (e.g., [1:56 pm on 8
May, 2023] means the conversation date is 8 May 2023).

Step 2: Identify the relative time expression (e.g., "yesterday", "last
week", "last Saturday").

Step 3: Calculate the actual date. "yesterday" = conversation date
minus 1 day. "last week" = approximately 7 days before. "last Saturday"
= the most recent Saturday before the conversation date. "next month" =
the month after the conversation month.

Step 4: State the calculated date in your answer.

3. When multiple events of the same type exist (e.g., multiple camping
trips, multiple beach visits), distinguish between them using their
dates.

4. Prefer quoting specific details (names, dates, objects, places) from
the context over paraphrasing.

5. If the context contains partial but relevant information, provide the
best answer you can.

6. Only say you cannot answer if the context truly contains NO relevant
information at all.

Answer:
```

LoCoMo judge

```
You are an evaluation judge. Compare the generated answer
with the gold answer and determine if the generated answer is correct.

Be lenient with format differences. For example:

- "May 7th" and "7 May" are the same date -> CORRECT

- "Caesar salad" and "She ordered a Caesar salad" -> CORRECT (same
meaning)

- Partial but accurate answers are CORRECT

Question: {question}

Gold answer: {gold_answer}

Generated answer: {generated_answer}

First, provide a short (one sentence) explanation of your reasoning, then
finish with CORRECT or WRONG.

Return your response in JSON format:

{{"reasoning": "your explanation", "label": "CORRECT or WRONG"}}
```

E.3 Evidence-set selector: prompt

In a single LLM call, the selector returns an ordered list of up to ten evidence passages from the RRF-preselected pool of at most 30 candidates (Section 3.3); the reader receives the first five passages.

At each step the model sees the question, the passages selected so far (relabelled S1, S2, ...; a selected passage cannot be picked again), and the remaining candidates, whose ids are their ranks in the window. It states what the selected passages establish, names the missing hop as a sub-question, labels the leading candidates, and picks the next passage. Selection stops when the model reports that nothing is missing; any remaining slots follow RRF order. System message: `Return strict JSON only.` Temperature 0.

```
You are assembling the evidence for a multi-hop question, one passage at
↪  a time.

Question: {question}

The answer requires chaining several passages: an early passage names
↪  something the
question refers to only indirectly, and a later passage answers the
↪  question about that
thing. You are building that chain, not ranking passages one by one. The
↪  best next passage
supplies the link the selected passages are still missing. A passage
↪  that restates a fact
already selected adds nothing, however relevant it looks on its own.

Selected so far ({n} of at most {budget}):
{selected}

Candidates:
{candidates}

[set_links only:]
Under each candidate, "links" lists the rare names it shares with the
↪  question or with a
selected passage, rarest first. "(in n passages)" says how rare the name
↪  is in the whole
collection; the quotes show the sentence that uses the name on each
↪  side. A shared name is
only a lead. It is a real bridge only if
  (a) both passages mean the same thing by it -- the same person, place,
  ↪  work or
      organization, not a namesake, another sense of the word, or
      ↪  another edition, season
      or year; and
  (b) the candidate says something new about it that the chain still
  ↪  needs.
A shared name with a different referent, or a candidate that repeats
↪  what the selected
passage already says about it, is not a bridge. A candidate with no link
↪  can still be the
right next step.

Work in this order:
1. "state": one sentence with only what the selected passages themselves
↪  say ("nothing yet"
   if none) -- not what you know or expect.
2. "missing": the next single fact the chain needs, as a sub-question
↪  about something
   specific; do not restate the question. "none" if the selected
   ↪  passages already settle it.
   Example -- question: When did the country that produced the film X
   ↪  declare war on the US?
   With nothing selected: state "nothing yet", missing "Which country
   ↪  produced the film X?"
   With "X is a 1953 Japanese film" selected: state "X was produced in
   ↪  Japan.", missing
   "When did Japan declare war on the US?"
```

```
3. "judgments": for up to five of the most promising candidates, the id,
↪  the name the
   connection to the question or to a selected passage runs through (or
   ↪  null), and a role:
     anchor     resolves what the question names directly (only while
     ↪  nothing is selected)
     bridge     supplies the missing fact, about the same referent
     redundant  restates facts the selected passages already give
     namesake   shares a name with a selected passage but means
     ↪  something else by it
     off_chain  none of the above
4. "pick": the id to add next, normally the best anchor or bridge, from
↪  the candidate ids
   {valid} (selected passages cannot be picked again). If no candidate
   ↪  supplies the missing
   fact, pick the one most likely to lead toward it. null only when
   ↪  "missing" is "none".

Return JSON only:
{"state": "...", "missing": "...", "judgments": [{"id": 7, "via":
↪  "Fujian", "role": "bridge"}], "pick": 7}
```

Placeholders. `{selected}` lists `[S`j`]` `<passage>`, or `(none yet)`. `{candidates}` lists `[`i`]` `<passage>`; in the `set_links` variant each candidate is followed by a line `links: <link>; ...` (at most three links, rarest first) or `links: none`. A link is either `"E" (in` d `passages) is named in the question` or `"E" (in` d `passages) shared with [S`j`]: "<sentence of S`j` using E>" | here: "<sentence of the candidate using E>"`, restricted to entities with $\mathrm{df}(e) \leq \tau$ (Eq. 1), with at most two links per selected passage. `{valid}` lists the remaining candidate ids. The `set` variant removes the link paragraph and all `links` lines.

Link fields for other retrievers. A candidate's link field is computed only from the candidate passage, the passages already selected, the question, and a corpus-level entity index; it never uses the candidate's retrieval provenance. When the selector is applied to candidates from the graph-free hybrid retriever or HippoRAG 2, the field is therefore filled by the same rule from the same index (built once over the corpus and counted as part of the selector's offline cost). We do not leave the field empty: `links: none` asserts that the candidate shares no rare name with the question or the selected passages, and writing it for every baseline candidate would bias the selector against them. Nor do we drop the field for baselines only, which would compare two different selectors.