# REORG: RELATION-GUIDED EVIDENCE ORGANIZATION FOR RETRIEVAL-AUGMENTED GENERATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Post-retrieval context construction for retrieval-augmented generation (RAG) must preserve connections among facts, not merely select relevant passages. Retrieved passages can interleave shared information with unique, complementary facts, so passage-level selection or independent compression may discard evidence whose value emerges through joint reading. We call this problem **Evidence Entanglement** and propose **REORG** (Relation-Guided Evidence Organization), which establishes joint-reading scopes before removing content. REORG extracts local cross-passage attention associations under complementary passage orders and aggregates them into sparse sentence and passage graphs to form overlapping reading groups. These cues guide comparison rather than directly determine semantic relation types. Within each group, an organizer uses source text and association cues to jointly select evidence, returning only sentence identifiers and relation and role labels. Programmatic source recovery, deduplication, and rendering under a complete answer-input budget separate evidence selection from factual wording. Across the reported results on HotpotQA, 2WikiMQA, MuSiQue, and NQ-Open, REORG records unweighted mean EM/F1 of 36.39/46.06, exceeding CPC, the strongest baseline by these averages, by 2.36/2.41 points. Our code is available at https://anonymous.4open.science/r/REORG-2D15.

## 1 INTRODUCTION

Retrieval-augmented generation (RAG) provides large language models with the knowledge needed to answer questions by introducing external sources (Lewis et al., 2020). Fusion-in-Decoder (FiD) combines information from multiple retrieved passages in the decoder (Izacard & Grave, 2021). For complex questions that require the synthesis of multiple facts, relevant information is often distributed across documents and must be combined to form a complete answer (Yang et al., 2018). Obtaining relevant retrieval results is therefore only the first step. How their information is organized into a context that the model can use also affects whether the retrieved knowledge can be used effectively (Liu et al., 2024). This paper focuses on this post-retrieval context-construction process and studies how to reduce unnecessary content while preserving evidence that can jointly support an answer.

Recent context-construction methods refine both selection and representation. Provence unifies query-aware pruning with reranking (Chirkova et al., 2025), while attention-guided compression makes retention decisions over semantic units (Zhao et al., 2025). Refiner jointly extracts and organizes source text (Li et al., 2024), and LongRefiner exploits document hierarchies for query-aware refinement (Jin et al., 2025). SARA combines retained text with compressed vectors (Jin et al., 2026), while LogicRAG guides adaptive retrieval and context summarization with a query-dependency graph (Chen et al., 2026). Our focus is a particular condition within this landscape: retrieved passages are not independent information units. Different passages may share information while each contributes other facts, conditions, or details. Whether a passage should be retained therefore cannot be determined entirely by its own relevance, nor can the substitutability of two passages be judged solely from their overall similarity.

More specifically, shared information and source-specific information are often interleaved within the same passage, while the connections required by an answer cross passage boundaries. We call this condition **Evidence Entanglement**. The left panel of Figure 1 illustrates this condition and the corresponding relation-first perspective. Shared information is not inherently background that



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.

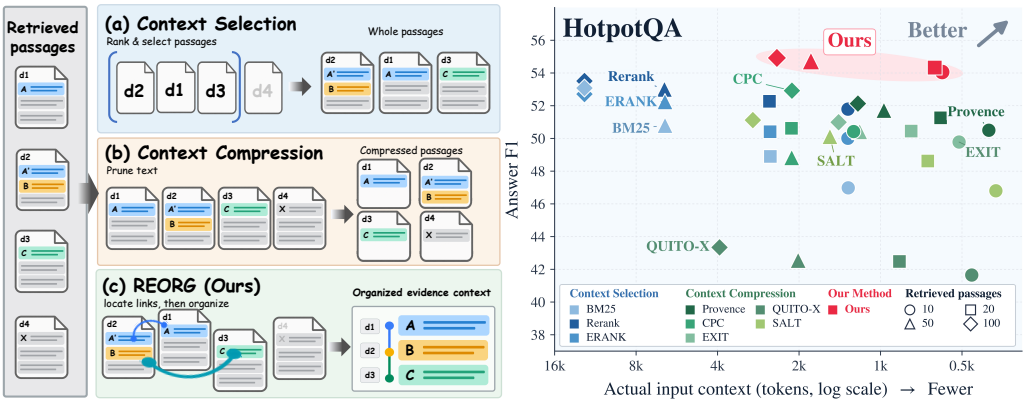


Figure 1: Paired views of relation-first evidence organization and answer quality. **Left:** Retrieved chunks may interleave shared content and complementary facts; relation-first grouping exposes local correspondences before joint selection. **Right:** Reported HotpotQA answer F1 versus actual input-context length (log scale, decreasing to the right). Marker shapes denote retrieved-chunk counts.

should be removed: it may directly support the answer or provide the basis for connecting other facts. Removing content according to whole-passage similarity can discard other supporting facts while reducing repeated expressions, whereas independently compressing each passage can overlook how a detail functions when used together with other documents. The problem is thus not only how much information to retain, but whether the selection process recognizes the roles that different pieces of information play together. This leads to our first motivation: *M#1: Let relations among cross-document content participate in evidence selection, rather than making retention decisions from each passage in isolation.*

Asking a language model to read and organize all retrieved material at once is a natural way to consider these connections. This strategy, however, requires the model to perform two tasks within a single organization process: locate content that is distributed across different positions but should be understood jointly, and decide which of that content should be retained. When multiple passages contain both shared content and distinct additions, evidence selection depends on repeated comparisons among them. We therefore separate the two tasks: first search the complete source text for potential local associations, and then perform joint evidence selection around those associations. This gives our second motivation: *M#2: Before any content is removed, first establish a scope for joint reading so that subsequent decisions can directly compare mutually related source text.*

Based on these motivations, we propose **Relation-First Evidence Organization**: first determine which content should be compared together, and then decide which evidence to retain and how to present the relations among the retained evidence. Realizing this principle requires addressing two challenges.

**Challenge #1: How can we locate cross-document content worth reading together?** A useful connection may involve only a few words or sentences within a long passage, which whole-passage relevance or similarity cannot localize. At the same time, exhaustive semantic comparison of every candidate pair adds substantial processing overhead as the candidate set grows. The system therefore needs association cues that point to concrete content in the unabridged source text and define what should be read together, without prematurely deciding which passages can replace one another.

**Challenge #2: How can multiple connections be used for joint evidence selection?** A passage may connect to several sources through different content. Assigning it to one fixed group would deny some of that content the opportunity to be assessed jointly with the relevant sources. Moreover, the existence of a connection does not imply content equivalence: related material may repeat the same fact or provide information that must be retained together. The system must therefore allow a passage to participate in multiple reading groups and assess retention from the source text within each group, rather than deduplicating or filtering directly according to association strength.

To address these challenges, we propose **REORG** (Relation-guided Evidence Organization), as summarized in Figure 2. **For Challenge #1,** REORG uses a language model to read the question and the complete retrieval candidates, extracts local cross-passage associations from attention, and aligns them to sentences in the source text. These associations indicate which content is worth reading together; they are not treated as confirmed instances of redundancy, complementarity, or reasoning relations. **For Challenge #2,** REORG uses the local associations to construct overlapping reading groups, so the same passage can participate in different comparisons. An organizer model considers the question, group source text, and association cues to jointly select answer-supporting sentences and label their relations.

At the output stage, REORG separates content selection from factual wording. The organizer model outputs only source-sentence identifiers and relation labels; a program recovers the corresponding source text instead of generating a factual summary that replaces it. The final context consists of original evidence and a relational structure that refers to that evidence, with a sentence presented only once when it is selected by multiple reading groups. This preserves the result of joint evidence selection while keeping evidence wording consistent with its source, without rewriting facts merely to express connections among evidence.

Our contributions are threefold:

- **Conceptually,** we identify context selection under interleaved shared and complementary information, and propose using cross-document connections as the basis for joint evidence selection.
- **Technically,** REORG combines local association discovery, overlapping reading groups, and extractive joint selection to construct answer contexts that preserve source evidence and its relations.
- **Experimentally,** REORG records the highest EM/F1 on each of four datasets in the reported comparison, with unweighted mean gains of 2.36/2.41 points over CPC, the strongest baseline by these averages (Table 1). HotpotQA F1 rises from 54.06 to 54.92 across 10–100 candidates, while EM is non-monotonic (Section 4.3). Sections 4.4 and 4.5 report descriptive component scores and parameter sweeps, rather than causal component or robustness estimates.

## 2 Related Work

### 2.1 Context Compression

Context compression reduces input through selective retention or compact representations. LLMLingua uses a small language model for coarse-to-fine, iterative token compression (Jiang et al., 2023). Recent methods couple pruning with reranking, as in Provence (Chirkova et al., 2025), or make context-aware sentence decisions, as in CPC and EXIT (Liskavets et al., 2025; Hwang et al., 2025). Attention guides semantic-unit pruning in Zhao et al. (2025) and adaptive document selection in the distinct AttnComp of Luo et al. (2025). SARA combines retained text with compression vectors and iterative evidence reranking (Jin et al., 2026); in personalization, Attn-GS uses attention-marked profile sentences to guide generative compression (Zeng et al., 2026). These approaches differ in selection granularity and representation. REORG's proposed design instead focuses on which cross-document spans to compare jointly before removal, since relevance alone does not identify which evidence must be retained together across sources.

### 2.2 Context Restructuring

Context restructuring organizes dispersed evidence for joint use. Refiner sections verbatim extracts and necessary context by interconnectedness (Li et al., 2024), while EDC$^2$-RAG forms non-overlapping document clusters before query-aware within-cluster compression (Li et al., 2025a). StructRAG chooses task-adaptive structural representations, whereas LongRefiner selects content using query scope and document hierarchies (Li et al., 2025b; Jin et al., 2025). Beyond refinement of a fixed candidate set, KG2RAG uses a knowledge graph to expand and organize chunks (Zhu et al., 2025), whereas LogicRAG builds query-dependency graphs for adaptive retrieval and rolling context summarization (Chen et al., 2026). REORG instead forms overlapping reading groups from local cross-document associations, jointly selects source sentences, and retains relation references. Its contrast concerns the scope and stage of joint reading, not the presence of relations or verbatim preservation alone. Appendix D expands these mechanism-level comparisons.

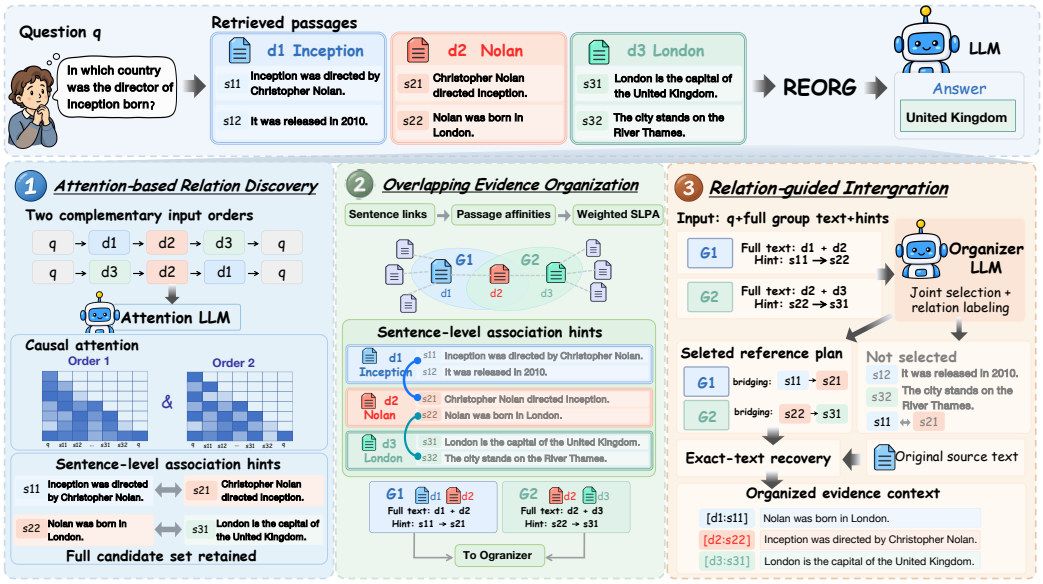


Figure 2: Overview of REORG. Attention-derived local associations are aligned to source sentences and organized into overlapping reading groups for joint selection and relation labeling. The final context recovers the selected source text rather than replacing it with a factual summary.

# 3 METHOD

## 3.1 PROBLEM FORMULATION

Given a question $q$, a set of retrieved passages $\mathcal{D} = \{d_i\}_{i=1}^{N}$, and an answer-input budget $B$, REORG transforms the retrieval results into a source-constrained evidence context. Let $\mathcal{S}_i$ denote the selectable body sentences in passage $d_i$, and let $\mathcal{S} = \bigcup_i \mathcal{S}_i$. The framework selects a sentence set $\mathcal{U} \subseteq \mathcal{S}$ and constructs a relation set $\mathcal{R}$ that refers to these sentences:

$$\begin{aligned} \mathcal{C}_B &= \mathrm{Render}(\mathcal{U}, \mathcal{R}), \\ \ell_{\mathrm{chat}}(q, \mathcal{C}_B) &\le B. \end{aligned} \tag{1}$$

Here, $\ell_{\mathrm{chat}}$ measures the complete answer-model input, including the task instruction, question, and rendered evidence.

To handle shared information interleaved with unique and complementary facts, REORG constructs the context in three connected stages. Attention-based Relation Discovery extracts sentence-level association hints; Overlapping Evidence Organization forms overlapping reading groups for joint assessment; and Relation-guided Integration selects evidence, labels relations, and recovers the source text. Relation discovery operates on the complete candidate set. Reranking signals determine the within-group reading order and the priority of evidence in the final output.

## 3.2 ATTENTION-BASED RELATION DISCOVERY

**Two complementary input orders.** The visible history of causal attention depends on input order. To examine cross-passage connections from complementary directions, we construct

$$\begin{aligned} x^{+} &= [q, d_1, \ldots, d_N, q], \\ x^{-} &= [q, d_N, \ldots, d_1, q], \end{aligned} \tag{2}$$

while preserving the within-passage order and the original source identifiers. Let $\pi_p(u)$ be the position of source token $u$ under ordering $p \in \{+, -\}$, and let $\mathcal{I}$ be the set of attention channels used for relation extraction. If $A_c^p$ denotes the attention probability of channel $c$, normalized over the complete causal history, we define

$$\alpha_p(u, v) = \max_{c \in \mathcal{I}} A_c^p\big(\pi_p(u), \pi_p(v)\big), \quad v \in \mathcal{K}_p(u), \tag{3}$$

where $\mathcal{K}_p(u)$ contains the visible tokens from passages other than the passage containing $u$. The full-attention layers, Q/K recomputation, and blockwise extraction procedure are deferred to Appendix A.3.

For every token $u$ acting as an attention query, we retain only its $k$ strongest cross-passage connections. Connections from the two orderings are then mapped to a common source coordinate system:

$$\begin{aligned} \overrightarrow{\mathcal{E}}_p &= \left\{ (u, v) : v \in \mathrm{TopK}_{v' \in \mathcal{K}_p(u)} \alpha_p(u, v') \right\}, \\ w_T(\{u, v\}) &= \max_{\substack{p \in \{+,-\},\ (a,b) \in \overrightarrow{\mathcal{E}}_p \\ \{a,b\} = \{u,v\}}} \alpha_p(a, b). \end{aligned} \tag{4}$$

Let $\mathcal{E}_T$ be the set of undirected token pairs supported by at least one retained directed connection. This merge converts local associations from the complementary orders into traceable token relations.

**Sentence links and passage affinities.** To connect fine-grained positions to joint-reading scopes, we define a strong-local-link aggregation operator

$$\mathrm{Agg}_r(X) = \frac{1}{r_X} \sum_{a \in \mathrm{Top}_{r_X}(X)} a, \qquad r_X = \min(r, |X|), \tag{5}$$

where $X$ is a nonempty multiset of edge weights. The operator combines the strongest links so that a local relation can be supported by multiple strong signals rather than a single maximum.

We first aggregate token links between the same pair of source sentences and then aggregate the retained sentence links between different passages:

$$\begin{aligned} w_U(s, t) &= \mathrm{Agg}_r\left( \left\{ w_T(\{u, v\}) : \substack{\{u,v\} \in \mathcal{E}_T \\ u \in s,\ v \in t} \right\} \right), \\ w_D(d_i, d_j) &= \mathrm{Agg}_r\left( \left\{ w_U(s, t) : \substack{\{s,t\} \in \mathcal{E}_U \\ s \in \mathcal{S}_i,\ t \in \mathcal{S}_j} \right\} \right). \end{aligned} \tag{6}$$

Here, $\mathcal{E}_U$ is the sentence-edge set retained after sparsifying the sentence relations. Each sentence node nominates its $m_s$ strongest neighbors, and each passage node nominates its $m_d$ strongest neighbors. At each level, the union of the nominated edges forms the corresponding sparse graph; unsupported node pairs receive no edge. The resulting passage relation graph is

$$\mathcal{G}_D = (\mathcal{D}, \mathcal{E}_D, w_D).$$

Each passage edge also stores its supporting high-weight sentence pairs, which keeps the construction of reading scopes traceable to local source positions.

### 3.3 Overlapping Evidence Organization

A passage may participate in several evidence connections through different local content. We therefore apply a weighted variant of the speaker–listener label propagation algorithm (SLPA) (Xie et al., 2011) to $\mathcal{G}_D$, using label memories to construct overlapping reading scopes. Let $c_u(\lambda)$ be the memory count of label $\lambda$ at node $u$, initialized as $c_u(\lambda) = \mathbf{1}[\lambda = u]$. During each asynchronous update, every neighbor of receiving node $u$ samples a label from its current memory distribution:

$$p_v(\lambda) = \frac{c_v(\lambda)}{\sum_\mu c_v(\mu)}, \qquad \lambda_v \sim \mathrm{Categorical}(p_v), \quad v \in \mathcal{N}(u). \tag{7}$$

The receiving node aggregates these votes with the passage-relation weights and updates its label memory:

$$\begin{aligned} \lambda_u^\star &= \arg\max_\lambda \sum_{v \in \mathcal{N}(u)} w_D(u, v) \mathbf{1}[\lambda_v = \lambda], \\ c_u(\lambda_u^\star) &\leftarrow c_u(\lambda_u^\star) + 1. \end{aligned} \tag{8}$$

After a preset number of rounds, let $p_u^{\mathrm{end}}$ be the final label frequency distribution. The retained memberships and their initial groups are

$$\begin{aligned} \Lambda_u &= \{\lambda : p_u^{\mathrm{end}}(\lambda) \geq \tau\}, \\ G_\lambda^{(0)} &= \{u : \lambda \in \Lambda_u\}. \end{aligned} \tag{9}$$

If $\Lambda_u$ is empty, the most frequent label is retained. Because a node may retain more than one label, the corresponding passage can participate in multiple joint-reading scopes.

We further decompose disconnected initial groups, adjust their sizes, and complete passage coverage to obtain $\mathfrak{G} = \{G_g\}_{g=1}^{M}$, where

$$\bigcup_{g=1}^{M} G_g = \mathcal{D}.$$

Oversized groups are recursively split, smaller groups are expanded through strong neighbors, and uncovered passages receive additional reading groups. The exact post-processing algorithm and its parameters are specified in Appendix A.3. For each $G_g$, we collect the sentence-ID pairs supporting its internal passage edges to form the sentence-level association hint set $\mathcal{H}_g$. The reading group defines the scope of joint assessment, while $\mathcal{H}_g$ identifies positions within that scope that warrant direct comparison.

### 3.4 Relation-guided Integration

**Joint selection and relation labeling.** Let $\mathcal{S}_g = \bigcup_{d_i \in G_g} \mathcal{S}_i$ be the selectable sentences in group $G_g$. Given the question, the group source text, and the sentence-level association hints, the organizer $F_\theta$ returns a selected reference plan consisting of typed source-sentence references:

$$\begin{aligned} \mathcal{Z}_g &= F_\theta(q, G_g, \mathcal{H}_g) = \{(\rho_{g,j}, I_{g,j})\}_{j=1}^{J_g}, \\ I_{g,j} &\subseteq \mathrm{ID}(\mathcal{S}_g) \times \mathcal{A}. \end{aligned} \tag{10}$$

Here, $\rho_{g,j}$ is a relation type and $\mathcal{A}$ is the set of evidence roles; $I_{g,j}$ associates source-sentence identifiers with their roles. The relation inventory covers duplicate, complementary, bridging, comparative, conflicting, and single-source evidence. Association hints guide local comparison, but content selection and relation labeling are decided from the group source text. The exact labels, output constraints, and organizer prompt are deferred to Appendix A.4.

**Exact-text recovery.** After validity checking, every reference is recovered through the source map. For sentence $s$, let $d(s)$ be its source passage and let $[a_s, b_s)$ be its character span. Source recovery requires

$$\mathrm{Quote}(s) = d(s)[a_s : b_s] = \mathrm{Text}(s). \tag{11}$$

Evidence wording is therefore determined by the source text, while the organizer is responsible for sentence selection and relation organization. Taking the union of sentence identifiers across groups gives the unique sentence set $\mathcal{U}$. Relation records with the same type and referenced identifier set are merged into $\mathcal{R}$. If one sentence receives multiple roles, the highest-priority role is retained. A sentence may participate in several relation sets, but its text is rendered only once.

**Organized evidence context.** We order $\mathcal{U}$ by evidence role, reranking position, and stable source order, obtaining $(s_1, \ldots, s_n)$. Define $\mathcal{U}_j = \{s_1, \ldots, s_j\}$, and let $\mathcal{R}|_{\mathcal{U}_j}$ remove references to omitted sentences and any relation records that become empty. The final context retains the longest prefix that satisfies the complete input budget:

$$\begin{aligned} j^\star = \max\Big\{ & j \in \{0, \ldots, n\} : \\ & \ell_{\mathrm{chat}}\big(q, \mathrm{Render}(\mathcal{U}_j, \mathcal{R}|_{\mathcal{U}_j})\big) \le B \Big\}, \\ \mathcal{C}_B = \mathrm{Render}(\mathcal{U}_{j^\star}, & \mathcal{R}|_{\mathcal{U}_{j^\star}}). \end{aligned} \tag{12}$$

We assume that $B$ can accommodate the complete prompt containing the question and empty evidence. Selection operates on complete sentences, and relation references are updated together with the retained evidence. The answer model then generates

$$\hat{y} = \mathrm{Answer}(q, \mathcal{C}_B).$$

Table 1: Reported EM/F1 and compression ratios on four QA datasets. **Bold** and underlining mark the highest and second-highest EM/F1, respectively.

| Method | Venue | HotpotQA | | | 2WikiMQA | | | MuSiQue | | | NQ | | | AVG | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | EM | F1 | CR | EM | F1 | CR | EM | F1 | CR | EM | F1 | CR | EM | F1 | CR |
| ***Vanilla Method*** | | | | | | | | | | | | | | | | |
| Vanilla | | 20.89 | 28.50 | – | 20.90 | 23.22 | – | 5.79 | 15.19 | – | 15.29 | 23.55 | – | 15.72 | 22.62 | – |
| COT | NeurIPS'22 | 25.16 | 34.63 | – | 29.43 | 34.44 | – | 8.19 | 15.89 | – | 22.24 | 33.85 | – | 21.26 | 29.70 | – |
| ***Context Selection*** | | | | | | | | | | | | | | | | |
| BM25 | TREC'94 | 36.88 | 48.90 | 5.0× | 42.55 | 48.27 | 4.9× | 8.77 | 17.73 | 5.0× | 33.13 | 44.57 | 5.0× | 30.33 | 39.87 | 5.0× |
| Qwen3-Reranker | arXiv'25 | 39.62 | 52.27 | 5.0× | 45.94 | 51.71 | 4.9× | 10.96 | 20.31 | 5.0× | 37.56 | 49.58 | 5.0× | 33.52 | 43.47 | 5.0× |
| ERANK | AAAI'26 | 38.03 | 50.41 | 5.7× | 42.63 | 48.60 | 5.6× | 9.85 | 19.05 | 5.6× | 34.82 | 46.97 | 5.7× | 31.33 | 41.26 | 5.7× |
| ***Context Compression*** | | | | | | | | | | | | | | | | |
| PROVENCE | ICLR'25 | 40.11 | 52.13 | 10.2× | 40.43 | 45.57 | 12.1× | 10.88 | 19.16 | 15.0× | 39.83 | 50.79 | 7.7× | 32.81 | 41.91 | 11.3× |
| CPC | AAAI'25 | 40.62 | 52.92 | 5.8× | 45.16 | 50.87 | 6.1× | 11.79 | 21.13 | 5.8× | 38.53 | 49.69 | 5.8× | 34.03 | 43.65 | 5.9× |
| EXIT | ACL'25 | 38.33 | 50.99 | 8.7× | 42.45 | 48.12 | 10.0× | 9.47 | 18.31 | 10.1× | 35.79 | 47.90 | 4.5× | 31.51 | 41.33 | 8.3× |
| QUITO-X | EMNLP'25 | 32.55 | 43.34 | 3.2× | 36.12 | 41.42 | 3.1× | 8.27 | 17.31 | 3.1× | 32.66 | 43.61 | 3.1× | 27.40 | 36.42 | 3.1× |
| SALT | EMNLP'26 | 39.26 | 51.12 | 4.2× | 47.05 | 52.72 | 4.1× | 10.43 | 20.42 | 4.2× | 37.92 | 48.83 | 4.3× | 33.67 | 43.27 | 4.2× |
| ***Our*** | | | | | | | | | | | | | | | | |
| **REORG** | | **42.39** | **54.92** | 5.1× | **49.69** | **55.26** | 6.4× | **12.83** | **22.10** | 4.9× | **40.64** | **51.96** | 3.8× | **36.39** | **46.06** | 5.1× |

# 4 EXPERIMENTS

The evaluation addresses four questions: whether REORG improves answer quality across QA tasks (RQ1), how its behavior changes with retrieval scale (RQ2), which components contribute (RQ3), and how sensitive answer quality is to the parameter settings (RQ4).

## 4.1 EXPERIMENTAL SETUP

**Datasets.** We evaluate on HotpotQA (Yang et al., 2018), 2WikiMQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and NQ-Open (Kwiatkowski et al., 2019; Lee et al., 2019), covering multi-hop and open-domain short-answer QA. Dataset descriptions are in Appendix B.3.

**Baselines.** We compare with context-selection baselines BM25 (Robertson et al., 1995), Qwen3-Reranker (Zhang et al., 2025), and ERANK (Cai et al., 2026); context-compression baselines PROVENCE (Chirkova et al., 2025), CPC (Liskavets et al., 2025), EXIT (Hwang et al., 2025), QUITO-X (Wang et al., 2025), and SALT (Mamo et al., 2026); and the Vanilla and CoT (Wei et al., 2022) reference conditions. Baseline descriptions are in Appendix B.4; evaluation and implementation details are in Appendix B.

## 4.2 MAIN RESULTS (RQ1)

Table 1 summarizes the reported comparison on HotpotQA, 2WikiMQA, MuSiQue, and NQ (NQ-Open). Compression ratio (CR) is expressed as a multiplicative factor. AVG is the unweighted mean of the four dataset values; CR AVG is a mean of ratios, not a pooled token ratio. Dashes indicate that no compression ratio is reported for the Vanilla and CoT reference conditions.

REORG records average EM/F1 of 36.39/46.06, compared with 34.03/43.65 for CPC, the strongest baseline by these reported averages. Among selection methods, Qwen3-Reranker records 33.52/43.47. The leading compression comparator varies across datasets: CPC records HotpotQA F1 of 52.92, SALT records 2WikiMQA F1 of 52.72, and PROVENCE records NQ-Open F1 of 50.79; the corresponding REORG values are 54.92, 55.26, and 51.96. These are descriptive numerical comparisons, not estimates of statistical significance.

The quality comparison must also account for retained length. On NQ-Open, REORG and PROVENCE have reported compression factors of 3.8× and 7.7×, respectively. The main table compares complete pipelines at their reported operating points, not at matched realized lengths. Its CR values are descriptive length indicators rather than measures of whole-pipeline computational cost. The component scores in Section 4.4 are likewise reported separately from the main quality comparison.

### 4.3 Effect of Retrieval Scale (RQ2)

Table 2: REORG on HotpotQA at four retrieval scales. Reported input tokens are rounded to the nearest integer. The full comparison is in Appendix C.1.

| **Metric** | $K = 10$ | $K = 20$ | $K = 50$ | $K = 100$ |
|---|---|---|---|---|
| **EM** | 41.61 | 41.99 | 42.57 | 42.39 |
| **F1** | 54.06 | 54.33 | 54.73 | 54.92 |
| Tokens | 592 | 631 | 1,808 | 2,415 |

Table 2 reports REORG on HotpotQA with $K \in \{10, 20, 50, 100\}$ retrieval candidates. The comparison tracks EM, F1, and reported input tokens, separating candidate count from the length passed downstream. The complete comparison with all baselines is in Appendix C.1.

From $K = 10$ to $K = 100$, F1 increases from 54.06 to 54.92, a gain of 0.86 points, while reported input length grows from 591.9 to 2,414.5 tokens. Thus, a tenfold increase in candidate count corresponds to a $4.08$-fold increase in reported input tokens, rather than proportional length growth. The last increase, from 50 to 100 candidates, adds 0.19 F1 points and 606.9 tokens. EM instead peaks at 42.57 for $K = 50$ and decreases to 42.39 for $K = 100$, so the larger pool does not improve both quality metrics uniformly.

These observations describe REORG's quality–length behavior at the reported operating points, not whole-pipeline efficiency or the causal effect of more candidates. Candidate count and downstream input length are distinct: the larger pools accompany both longer inputs and different answer scores. The comparison therefore does not isolate candidate count from realized length.

### 4.4 Component Ablation (RQ3)

Table 3 lists the four reported variants alongside the existing NQ-Open REORG result from Table 1. The latter serves as a main-table reference; the differences below are descriptive comparisons, not paired estimates of component effects.

Table 3: Reported NQ-Open component scores. REORG uses the Table 1 reference; differences are descriptive rather than paired component-effect estimates.

| **Variant** | **EM** | **F1** |
|---|---|---|
| **REORG** | 40.64 | 51.96 |
| w/o group overlap (A1) | 39.92 | 51.70 |
| w/o Organizer relation hints (A2) | 39.70 | 51.39 |
| w/o Answer relation structure (A3) | 39.86 | 51.29 |
| w/o group-wise organization (A4) | 39.53 | 50.97 |

A1 changes overlapping reading scopes to non-overlapping groups. A2 removes the Organizer's relation hints, targeting guidance during evidence selection. A3 removes the relation block supplied to Answer, targeting the presentation of selected evidence rather than sentence selection. Evidence roles determine sentence order and are not printed as separate tags. A4 organizes all Top-100 candidates in one Organizer call rather than separate reading groups. Appendix C.2 describes the corresponding interventions.

Relative to REORG's reference F1 of 51.96, A1–A4 are lower by 0.26, 0.57, 0.67, and 0.99 points, respectively. A1 records the highest scores among the four variants, while A4 records the lowest EM/F1, 39.53/50.97. This ordering describes the reported point estimates, not the causal importance of the components. In particular, the comparison does not isolate grouping from relation-hint access, input-window effects, or processing cost.

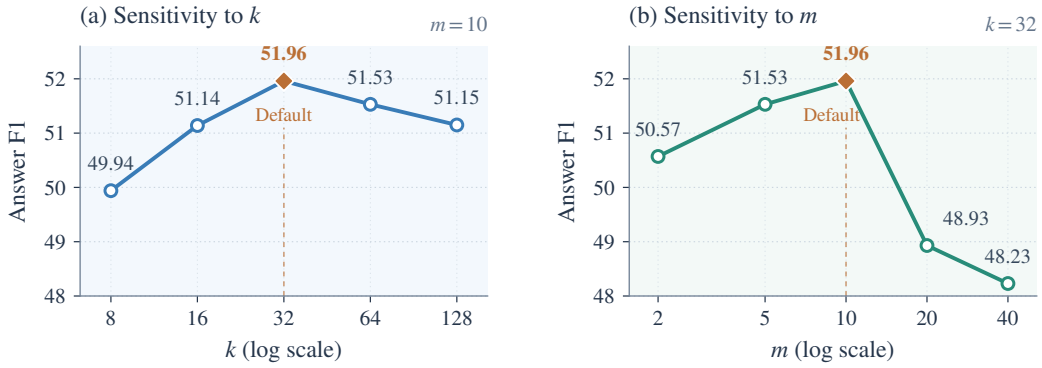


Figure 3: Reported Answer F1 sensitivity curves: $k$ with $m = 10$ (left), and $m$ with $k = 32$ (right). Highlighted points mark the reference configuration $(k, m) = (32, 10)$ in the two sweeps.

### 4.5 Parameter Sensitivity (RQ4)

Figure 3 presents the reported Answer F1 curves for $k \in \{8, 16, 32, 64, 128\}$ with $m = 10$, and $m \in \{2, 5, 10, 20, 40\}$ with $k = 32$. Both panels use logarithmic horizontal axes and highlight $(k, m) = (32, 10)$. We retain the sweep labels of the reported curves; the plotted $m$ is not identified with either of the separately defined neighbor counts $m_s$ and $m_d$.

The curves describe answer-score variation over the displayed settings. Their highlighted configuration marks the experimental setting $(k, m) = (32, 10)$. These point estimates do not establish broad robustness, causal graph/group effects, or processing efficiency. Appendix C.3 explains the dependencies between graph parameters and downstream reading groups.

## 5 Limitations, Future Work and Conclusion

REORG uses preset neighbor counts rather than adapting graph sparsity to the evidence density of each question. Future work will explore question-adaptive sparsity while retaining source-aligned association hints. Selecting complete sentences can retain irrelevant clauses alongside useful facts. Future work will explore clause-level extraction with exact source offsets, preserving attribution and the context needed to interpret each fact.

We presented REORG for post-retrieval context construction under Evidence Entanglement: shared information and complementary facts interleaved across retrieved passages. Local attention associations establish overlapping reading groups before an organizer jointly selects source sentences and labels their relations. Exact-text recovery and budgeted rendering assemble the final context, separating evidence selection and organization from factual wording. Across four QA datasets, REORG's reported unweighted mean EM/F1 is 36.39/46.06, versus 34.03/43.65 for CPC, the strongest baseline by these averages. On HotpotQA, F1 increases across the tested retrieval scales, while EM is non-monotonic. Component and sensitivity comparisons provide descriptive diagnostics, not causal component effects or broad robustness estimates. The design principle is to determine what to read together before selecting evidence and preserve its source wording.

## AI USE STATEMENT

During the preparation of this manuscript, large language models (LLMs) assisted with manuscript drafting and language refinement, related-work literature search, coding, figure preparation, and LaTeX formatting. The authors are responsible for reviewing and verifying AI-assisted content, including text, references, code, and figures. The research design, experimental analysis, and all scientific conclusions were carried out independently by the authors without LLM involvement. The authors made all final decisions regarding the research design, experimental analysis, and scientific conclusions, and take full responsibility for the final content of this work.

## ETHICS STATEMENT

This work studies post-retrieval context construction for question answering using existing benchmarks and a Wikipedia-based retrieval corpus. Potential ethical risks include biased or inaccurate source material, exposure of sensitive information, and overreliance on generated answers. Preserving source sentences and relation references does not guarantee factual correctness, evidence completeness, or fairness. Use and redistribution of datasets, retrieved text, and model artifacts should comply with applicable licenses and access conditions. Downstream applications should protect personal information, validate outputs for the intended domain, and retain appropriate human oversight. The reported question-answering evaluations do not establish safety for high-stakes deployment.

## REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we describe the overall REORG algorithm in Appendix A.2, implementation details and experimental hyperparameters in Appendix A.3, and Organizer and Answer prompts and output formats in Appendix A.4. Wikipedia preprocessing, chunking, and BM25 indexing are described in Appendices B.1 and B.2. Dataset and baseline descriptions appear in Appendices B.3 and B.4; model settings and evaluation protocols are discussed in Appendices B.5 and B.6. Additional results and interpretation details are provided in Appendix C. Our core implementation is available at https://anonymous.4open.science/r/REORG-2D15. The experimental settings are specified in Table 5 and Appendix B.5.

## REFERENCES

Yuzheng Cai, Yanzhao Zhang, Dingkun Long, Mingxin Li, Pengjun Xie, and Weiguo Zheng. ERank: Fusing supervised fine-tuning and reinforcement learning for effective and efficient text reranking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 30121–30129, 2026.

Shengyuan Chen, Chuang Zhou, Zheng Yuan, Qinggang Zhang, Zeyang Cui, Hao Chen, Yilin Xiao, Jiannong Cao, and Xiao Huang. You don't need pre-built graphs for RAG: Retrieval augmented generation with adaptive reasoning structures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 30270–30278, 2026.

Nadezhda Chirkova, Thibault Formal, Vassilina Nikoulina, and Stéphane Clinchant. Provence: Efficient and robust context pruning for retrieval-augmented generation. In *International Conference on Learning Representations*, 2025.

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.

Taeho Hwang, Sukmin Cho, Soyeong Jeong, Hoyun Song, SeungYoon Han, and Jong C. Park. EXIT: Context-aware extractive compression for enhancing retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 4895–4924, 2025.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 874–880, 2021.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376, 2023.

Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yongkang Wu, Zhonghua Li, Ye Qi, and Zhicheng Dou. Hierarchical document refinement for long-context retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3502–3520, 2025.

Yiqiao Jin, Kartik Sharma, Vineeth Rakesh, Yingtong Dou, Menghai Pan, Mahashweta Das, and Srijan Kumar. SARA: Selective and adaptive retrieval-augmented generation with context compression. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14508–14528, 2026.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6086–6096, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, pp. 9459–9474, 2020.

Weitao Li, Xiangyu Zhang, Kaiming Liu, Xuanyu Lei, Weizhi Ma, and Yang Liu. Efficient dynamic clustering-based document compression for retrieval-augmented-generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 9833–9849, 2025a.

Zhonghao Li, Xuming Hu, Aiwei Liu, Kening Zheng, Sirui Huang, and Hui Xiong. Refiner: Restructure retrieved content efficiently to advance question-answering capabilities. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 8548–8572, 2024.

Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, and Yongbin Li. StructRAG: Boosting knowledge intensive reasoning of LLMs via inference-time hybrid information structurization. In *International Conference on Learning Representations*, 2025b.

Barys Liskavets, Maxim Ushakov, Shuvendu Roy, Mark Klibanov, Ali Etemad, and Shane K. Luke. Prompt compression with context-aware sentence encoding for fast and improved LLM inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 24595–24604, 2025.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

Lvzhou Luo, Yixuan Cao, and Ping Luo. AttnComp: Attention-guided adaptive context compression for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 8456–8472, 2025.

Oteo Mamo, Hyunjin Yi, Joydhriti Choudhury, Shangqian Gao, and Weikuan Yu. SALT: Salience-aware lexical trie for long-context compression. *arXiv preprint arXiv:2607.17486*, 2026.

Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Overview of the Third Text REtrieval Conference*, pp. 109–126, 1995.

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.

Yihang Wang, Xu Huang, Bowen Tian, Yueyang Su, Lei Yu, Huaming Liao, Yixing Fan, Jiafeng Guo, and Xueqi Cheng. QUITO-X: A new perspective on context compression from the information bottleneck theory. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6841–6856, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, pp. 24824–24837, 2022.

Jierui Xie, Boleslaw K. Szymanski, and Xiaoming Liu. SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *IEEE International Conference on Data Mining Workshops*, pp. 344–349, 2011.

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

Shenglai Zeng, Tianqi Zheng, Chuan Tian, Dante Everaert, Yau-Shian Wang, Yupin Huang, Michael J. Morais, Rohit Patki, Jinjin Tian, Xinnan Dai, Kai Guo, Monica Xiao Cheng, and Hui Liu. Attn-GS: Attention-guided context compression for efficient personalized LLMs. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35925–35946, 2026.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Yunlong Zhao, Haoran Wu, and Bo Xu. Leveraging attention to effectively compress prompts for long-context LLMs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 26048–26056, 2025.

Xiangrong Zhu, Yuexiang Xie, Yi Liu, Yaliang Li, and Wei Hu. Knowledge graph-guided retrieval augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8912–8924, 2025.

# A METHOD DETAILS

This appendix specifies the REORG implementation, its experimental settings, and its source-reference interface. Knowledge-base construction, dataset descriptions and metric conventions are discussed in Appendix B.

## A.1 NOTATION

Table 4 collects the main symbols used in Section 3. They include data-dependent objects as well as hyperparameters; fixed numerical settings are listed separately in Table 5. Passage and sentence indices follow their source order. The symbols $u, v$ denote tokens during relation discovery and passage nodes during label propagation, as indicated by the corresponding graph. $\mathrm{TopK}$ and $\mathrm{Top}_r$ select the specified number of strongest entries, capped by availability.

Table 4: Main notation in REORG. Retrieval scale K differs from token-link count k; candidate count N differs from selected-sentence count n.

| Symbol | Meaning | Location |
|---|---|---|
| **Problem and input/output** | | |
| $q$ | Question to be answered. | Sec. 3.1 |
| $\mathcal{D},\ d_i,\ N$ | Retrieved passage set, its i-th passage, and candidate count. | Sec. 3.1 |
| $K$ | Retrieval scale in the experiments; the candidate count N for that setting, not a baseline's retained count. | Sec. 4.3 |
| $\mathcal{S}_i,\ \mathcal{S}$ | Selectable body sentences in passage i, and their union over passages. | Sec. 3.1 |
| $B,\ \ell_{\mathrm{chat}}$ | Complete Answer-input budget and its token-counting function, including instructions, question and chat formatting. | Eq. 1 |
| $\mathcal{U},\ \mathcal{R}$ | Selected source-sentence set and relation records referring to those sentences. | Eq. 1 |
| $\mathcal{C}_B,\ \hat{y}$ | Budgeted evidence context and final generated answer. | Eq. 12 |
| **Local associations and sparse graphs** | | |
| $x^+,\ x^-,\ p$ | Complementary passage orders; p identifies the forward or reverse pass. | Eq. 2 |
| $\pi_p(u)$ | Position of canonical source token u in pass p. | Eq. 3 |
| $\mathcal{I},\ A_c^p$ | Selected attention channels and a channel's causal-history-normalized attention probabilities. | Eq. 3 |
| $\mathcal{K}_p(u)$ | Causally visible keys from passages other than the passage containing u. | Eq. 3 |
| $\alpha_p(u,v),\ k$ | Maximum channel attention for a cross-passage token pair, and retained keys per query token. | Eq. 3 |
| $\overrightarrow{\mathcal{E}}_p$ | Directed top-k token connections retained in pass p. | Eq. 4 |
| $\mathcal{E}_T,\ w_T$ | Merged undirected token-edge set and edge weights. | Eq. 4 |
| $\mathrm{Agg}_r,\ r,\ r_X$ | Mean of the strongest $r_X$ weights in nonempty multiset $X$, with $r_X = \min(r, \lvert X\rvert)$. | Eq. 5 |
| $\mathcal{E}_U,\ w_U$ | Retained sentence edges and token-link-aggregated sentence weights. | Eq. 6 |
| $m_s,\ m_d$ | Neighbor nominations per sentence and per passage. Union symmetrization can give a larger realized degree. | Eq. 6 |
| $\mathcal{E}_D,\ w_D,\ \mathcal{G}_D$ | Sparse passage edges, their sentence-aggregated weights, and the passage graph. | Sec. 3.2 |
| **Overlapping reading groups** | | |
| $\mathcal{N}(u),\ \lambda$ | Neighbors of a passage node and a propagated community label. | Eq. 7 |
| $c_u(\lambda)$ | Count of label lambda in node u's memory. | Eq. 7 |
| $p_v(\lambda),\ \lambda_v$ | Speaker label distribution and the label sampled by neighbor v. | Eq. 7 |

Continued on next page

| Symbol | Meaning | Location |
|---|---|---|
| $\lambda_u^\star$ | Label receiving the greatest edge-weighted vote at listener u. | Eq. 8 |
| $p_u^{\mathrm{end}},\ \tau$ | Final label-frequency distribution and membership-retention threshold. | Eq. 9 |
| $\Lambda_u,\ G_\lambda^{(0)}$ | Retained labels at u and the initial group associated with label lambda. | Eq. 9 |
| $\mathfrak{G},\ G_g,\ M$ | Final group family, its g-th reading group, and number of groups. | Sec. 3.3 |
| $\mathcal{H}_g,\ \mathcal{S}_g$ | Group-local sentence-pair hints and all selectable sentences in group g. | Eq. 10 |
| **Selection, recovery and rendering** | | |
| $F_\theta,\ \mathcal{Z}_g$ | Organizer model and its typed source-reference plan for group g. | Eq. 10 |
| $\rho_{g,j},\ I_{g,j},\ J_g$ | Relation type, sentence-ID/role pairs, and number of returned relation records. | Eq. 10 |
| $\mathcal{A},\ \mathrm{ID}$ | Evidence-role inventory and the mapping from source sentences to stable identifiers. | Eq. 10 |
| $d(s),\ [a_s, b_s)$ | Source passage of sentence s and its half-open character interval in that passage. | Eq. 11 |
| $\mathrm{Quote},\ \mathrm{Text}$ | Recovered source substring and stored sentence text, required to be equal. | Eq. 11 |
| $\mathrm{Render}$ | Assembly of selected source text and surviving relation references. | Eq. 1 |
| $s_1, \ldots, s_n,\ n$ | Priority-ordered selected sentences and their count before budget trimming. | Eq. 12 |
| $\mathcal{U}_j,\ \mathcal{R}\vert_{\mathcal{U}_j}$ | First j sentences and relation records restricted to their retained identifiers. | Eq. 12 |
| $j^\star$ | Longest sentence prefix whose complete rendered Answer input satisfies B. | Eq. 12 |

### A.2 Overall Algorithm

Algorithm 1 summarizes the post-retrieval procedure. The input passages retain stable source identities throughout both attention passes and all reading groups. Reranking changes reading and output priorities, not membership of the relation-discovery candidate pool. Offline corpus construction is outside this algorithm and is specified in Appendix B.1.

**Algorithm 1** REORG: relation-guided evidence context construction

**Require:** Question $q$, passages $\mathcal{D}$, complete-input budget $B$, configuration
**Ensure:** Source-grounded evidence context $\mathcal{C}_B$

Assign stable passage, sentence and token IDs; retain source spans
Score all passages with the reranker without removing candidates
*Attention-based Relation Discovery*
**for** $p \in \{+, -\}$ **do**
    Construct $x^p$, preserving within-passage order and canonical IDs
    Normalize attention over each query's full causal history
    Retain top-$k$ cross-passage keys by $\alpha_p$
**end for**
Merge pass edges into $(\mathcal{E}_T, w_T)$ in source coordinates
Aggregate token links and sparsify the sentence graph using $m_s$
Aggregate retained sentence links and sparsify $\mathcal{G}_D$ using $m_d$
*Overlapping Evidence Organization*
Obtain overlapping label memberships by weighted SLPA
Decompose, split and expand groups; complete passage coverage
Check coverage and configured group-size bounds
*Relation-guided Integration*
**for** each reading group $G_g \in \mathfrak{G}$ **do**
    Order group passages by reranking position; collect hints $\mathcal{H}_g$
    Obtain typed sentence references $\mathcal{Z}_g = F_\theta(q, G_g, \mathcal{H}_g)$
    Validate references and recover exact source sentences
**end for**
Merge selected sentence IDs and relation records into $(\mathcal{U}, \mathcal{R})$
Order selected sentences by role priority, reranking and stable source order
Select the longest budget-feasible prefix and restrict relation references
**return** $\mathcal{C}_B = \mathrm{Render}(\mathcal{U}_{j^\star}, \mathcal{R}|_{\mathcal{U}_{j^\star}})$

The answer model subsequently receives $(q, \mathcal{C}_B)$. Reference validation and budget checks are part of construction, not evidence that every required fact has been retained. Their failure behavior and the remaining relation-validity boundary are described below.

### A.3 Implementation Details and Hyperparameters

**Source coordinates and attention extraction.** The implementation assigns passage IDs in the original BM25 order, keeps titles as separate units, and segments each retrieved body into sentences at punctuation or newline boundaries while preserving character offsets. Tokenization records offsets within each unit and canonical token IDs. Only passage order reverses between the two passes; question and separator tokens carry no selectable source identity.

The attention adapter targets the Qwen3.5-family gated full-attention modules. It selects layers marked `full_attention` in the checkpoint configuration and excludes linear-attention layers. Hooks capture layer inputs and rotary-position values; Q/K projections, Q/K normalization, rotary embeddings and grouped-query head mapping are reconstructed with the native modules. The adapter takes the maximum attention probability over heads, retains per-layer cross-passage top-$k$ keys, and merges layers by maximum weight before applying the same top-$k$ cap. Thus $\mathcal{I}$ contains all heads of the selected full-attention layers; its concrete layer indices depend on the checkpoint.

Query rows are processed in blocks of 256 tokens. For each row, softmax uses all legal causal keys before same-passage and non-source keys are filtered. Blocking therefore partitions the computation, not the causal history into independent windows. Causally invisible positions are excluded from sparse top-$k$ even when fewer than $k$ cross-passage keys exist. Forward/reverse edges retain their source coordinates and pass/layer/head provenance.

**Sparse graphs and reading-group repair.** The strongest token links are averaged for each sentence pair. Sentence nodes nominate their $m_s$ strongest neighbors, and the union of these nominations forms the undirected graph. Passage weights similarly average strong retained body-sentence links and use $m_d$-neighbor nomination. Title units do not contribute to passage-edge aggregation. The two aggregation counts are separate configuration fields; both are set to three. Neither neighbor setting is an upper bound on realized undirected degree.

Weighted SLPA follows the label-memory operations in Section 3.3, based on Xie et al. (2011). Each round shuffles listener order, speakers sample from their memories, and an equal weighted vote is resolved by the smallest label. Isolated nodes retain their initial labels. Thresholded communities are decomposed into connected components; duplicates and strict subsets are removed. Oversized groups are recursively split using weighted Louvain, with resolution 1.2 at the first level and an increment of 0.5 per recursion level. If a split returns only one community, nodes are ordered by decreasing within-group weighted degree, with node identity breaking ties, and cut into target-sized blocks. Small groups expand through the strongest available neighbor edge, with source index breaking ties. Coverage completion adds groups for uncovered passages, followed by duplicate/subset removal.

The base seed controls SLPA. Group-size repair uses the base seed plus 1009 times the sample index, with offsets of 101 times the parent group index and the recursion depth for splitting. These indices are zero-based in the implementation. The wrapper checks complete candidate coverage and the configured minimum/maximum group sizes; infeasible bounds, including small disconnected components, stop processing rather than silently changing the settings. Reranking order is applied only after group membership has been fixed.

Table 5: REORG hyperparameter settings used in the experiments unless otherwise specified. Dashes mark controls without a separate main-text symbol.

| Control | Symbol | Value | Scope |
|---|---|---|---|
| Token links per query | $k$ | 32 | Relation discovery |
| Strong-edge aggregation | $r$ | 3 / 3 | Token-to-sentence / sentence-to-passage |
| Sentence neighbors | $m_s$ | 10 | Sentence graph |
| Passage neighbors | $m_d$ | 10 | Passage graph |
| Attention query block | – | 256 | Tokens; full causal key history |
| SLPA rounds | – | 50 | Asynchronous updates |
| Membership threshold | $\tau$ | 0.1 | Label-memory frequency |
| Group minimum / target / maximum | – | 5 / 10 / 20 | Passages per group |
| Base random seed | – | 20260918 | SLPA and group repair |
| Complete Answer input | $B$ | 8,000 | Tokens including the chat template |

**Exact recovery, merging and budget handling.** Every selected sentence must belong to the supplied group. Its character span is checked against the original chunk, and the recovered substring must equal the stored sentence text. Cross-group selections are deduplicated by sentence ID; relation records are merged when both their type and member ID set agree. Role priority is `direct`/`comparison` > `bridge` > `constraint` > `context`. Equal-priority assignments retain the first encountered role in the ordered group traversal.

The renderer lists each retained sentence once, with its ID and title, then lists relation types and their member IDs. It does not print a separate role tag beside each source sentence; roles affect ordering. Starting from the complete priority-ordered sentence list, the implementation removes sentences from its end and re-tokenizes the complete chat prompt until the budget is met. An over-budget empty-evidence prompt raises an error. References to dropped sentences and empty relation records are removed. A nonempty singleton can still retain a multi-source relation label: the renderer does not enforce type-dependent minimum membership. Exact quotation therefore does not imply completeness of the displayed relation.

### A.4 Prompts and Output Formats

The following English prompts are used in the experiments. The Organizer version is `ours-high-recall-ids-v1`; the Answer version is `ours-exact-quotes-answer-v1`. Braced fields in the input templates denote runtime substitutions, not literal instructions. The ablation variants modify the corresponding inputs as described in Appendix C.2.

**Organizer System Prompt**

```
You are an evidence organization model.

Organize and compress evidence from one group of retrieved Wikipedia
chunks
for a given question.

Rules:
1. Use only information explicitly stated in the supplied source
chunks.
2. Do not answer the question. Produce high-recall evidence for a
later answering model.
3. Prioritize recall over aggressive compression. Preserve every fact
that could
   participate in a direct or multi-hop answer, including identity,
   aliases,
   relationships, dates, quantities, locations, and bridge entities.
4. Merge genuinely equivalent evidence sets, but do not merge evidence
merely
   because it mentions the same entity or topic.
5. Preserve meaningful differences in entities, dates, quantities,
negation,
   conditions, and scope.
6. Put contradictory evidence in a conflict evidence set.
7. Every retained item must cite a valid sentence ID. Do not generate
statements
   or copy quotes; the program will recover exact source text from the
   ID.
8. Relation hints only identify sentences worth comparing. They are
not evidence
   and may be ignored when unsupported by source text.
9. Do not introduce outside knowledge.
10. Return valid JSON only, without a Markdown fence or extra fields.
11. Return at most 6 evidence_sets, with at most one set for each
relation type.
    Prefer compact multi-source sets, but do not
    discard a potentially useful bridge fact merely because it does
    not answer
    the question by itself.
12. A group that mentions any entity explicitly named in the question
is useful.
    Retain at least one sentence for every such question entity
    present in the group,
    using role=context if no stronger role applies.
13. Cite each sentence ID at most once and cite at most 16 sentence
IDs total.
14. To keep the JSON compact, combine items with the same
relation_type into one
    evidence set. Never emit multiple single_source sets. Do not add
    whitespace
    for readability.

Output schema:
{
  "group_id": "the supplied group ID",
  "status": "useful or no_relevant_evidence",
```

**Organizer System Prompt** (continued)

```
  "evidence_sets": [
    {
      "relation_type": "single_source, duplicate, complementary,
      bridge, comparison, or conflict",
      "items": [
        {
          "sentence_id": "D001-U001",
          "role": "direct, bridge, comparison, constraint, or context"
        }
      ]
    }
  ]
}

Use status=no_relevant_evidence only when none of the chunks contains
direct,
indirect, identity, or bridge evidence for the question. Do not force
relation
hints into the output, but inspect both endpoints before deciding they
are irrelevant.
```

**Organizer User-Input Template**

```
Organize this chunk group into compact evidence for the question.

Question:
{question}

Group ID:
{group_id}

Sentence relation hints, ordered approximately from stronger to
weaker:
{sentence_relation_hints_or_none}

Source chunks:
{source_chunks_with_sentence_ids}

Return the JSON object specified by the system instructions.
```

Each source chunk is enclosed in a `chunk` element with its ID and title; its body lists `[sentence_id]` followed by the exact sentence. Hints are distinct sentence-ID pairs ordered approximately by supporting passage-edge strength; an empty hint list is rendered as `(none)`. All source sentences in the group remain available to the Organizer, not only the hinted endpoints.

**Labels and validation.** Table 6 explains the fixed label inventory. These are intended semantic readings, not truth conditions checked by the parser. The system prompt requests at most six sets, one per relation type, and at most sixteen sentence IDs in total. The JSON schema enforces at most six sets and one to sixteen items *per set*, with fixed labels and no additional fields; the one-set-per-type and total-ID limits are prompt constraints, not stronger validator guarantees.

The output schema requires `group_id`, `status` and `evidence_sets`. The validator rejects unknown or out-of-group sentence IDs, invalid labels and unexpected fields, and removes repeated sentence IDs in traversal order. The supplied group ID and the status derived from retained sets are authoritative and may normalize those two redundant fields; sentence IDs, roles and relation types are not guessed. An invalid output stops the stage; there is no automatic retry or replacement evidence. The prompt's JSON block is a format specification, not a measured example or a source of factual evidence.

Table 6: Intended readings of the fixed relation labels and evidence roles. Priority applies to roles; schema validation does not certify semantic correctness.

| Label | Intended reading | Priority |
|---|---|---|
| *Relation types* | | |
| `single_source` | Standalone evidence without an asserted cross-source relation. | – |
| `duplicate` | Equivalent factual content, not merely a shared entity or topic. | – |
| `complementary` | Distinct facts that can contribute jointly to an answer. | – |
| `bridge` | Evidence connecting entities or intermediate reasoning steps. | – |
| `comparison` | Facts relevant to a comparison required by the question. | – |
| `conflict` | Source claims with a meaningful contradiction. | – |
| *Evidence roles* | | |
| `direct` | Direct answer-supporting fact. | 5 |
| `comparison` | Fact used in a requested comparison. | 5 |
| `bridge` | Intermediate connection needed for a possible answer. | 4 |
| `constraint` | Condition, scope or qualification of a fact. | 3 |
| `context` | Supporting identity or contextual information. | 2 |

**Answer System Prompt**

```
Answer the question using the supplied exact source evidence.
Give the most likely final answer supported by the evidence.
Return only the final short answer string. Do not explain.
```

**Answer User-Input Template**

```
Question: {question}

Source Sentences:
{source_sentence_block}

Evidence Groups:
{evidence_group_block}

Answer:
```

The source-sentence block expands to each retained ID, title and exact quote. The relation block expands to a record ID, relation type and retained member IDs joined by `<->`. Empty blocks use the markers shown in the implementation: `(no selected source sentences)` and `(none)`. The complete chat template, including these blocks, is counted against $B$; no reference answer is supplied.

**Reranker Task Instruction**

```
Given a multi-hop question, rank Wikipedia passages by how useful they
are for answering the question. A passage is useful if it directly
answers part of the question or provides bridge evidence needed to
connect entities across passages.
```

The reranker receives this instruction with the question, title and body, and scores the relative yes/no logits. Its instruction is a ranking input, not an instruction to generate replacement evidence.

# B KNOWLEDGE BASE AND EXPERIMENTAL SETUP

## B.1 WIKIPEDIA SOURCE AND PREPROCESSING

The external knowledge source is the March 1, 2026 English Wikipedia snapshot released by Wikimedia, identified as `enwiki-20260301` and the `pages-articles-multistream` dump. Corpus construction retains main-namespace entries and excludes redirects. Light WikiText cleaning removes comments, reference tags, common templates, and file and category links; internal links are converted to readable text and whitespace is normalized. Entries that become empty are discarded. The construction record reports 7,142,535 retained articles.

This corpus supplies external retrieval candidates rather than the benchmark-provided supporting documents. Corpus preprocessing is performed before REORG and is separate from its source-sentence selection procedure.

## B.2 CHUNKING AND BM25 INDEXING

Cleaned articles are divided into non-overlapping chunks, preferring sentence and paragraph boundaries. The target length is 500 characters, the soft lower limit is 256 characters, and the hard upper limit is 1,024 characters. Overlong sentences are split when necessary; a short tail is merged when doing so does not exceed the hard limit. The soft lower limit is not a guarantee on every resulting chunk. The corpus contains 61,284,624 chunks, with mean length 441.57 characters, as summarized in Table 7. These are character lengths, not model token lengths or the attention query-block size.

The chunk-level BM25 inverted index is built with Tantivy and divided into 16 shards. It stores article title, page ID, chunk ID and text so retrieved evidence can retain its source identity. The accompanying retrieval adapter also reads a unique candidate `id`. It queries the indexed `title` and `chunk_text` fields, obtains up to $K$ hits from each shard and selects the final $K$ candidates from their merged scores. If query parsing fails, its fallback retains lowercased word tokens. The adapter uses each stored index's configuration and does not recompute global BM25 statistics across shards; no equivalence to a monolithic globally normalized index is assumed.

Table 7: Wikipedia corpus and BM25-index statistics from the construction record. All chunk lengths are in characters, not model tokens.

| Statistic | Value |
|---|---|
| Snapshot identifier | `enwiki-20260301` |
| Dump family | `pages-articles-multistream` |
| Retained articles | 7,142,535 |
| Text chunks | 61,284,624 |
| Target chunk length | 500 |
| Soft lower / hard upper length | 256 / 1,024 |
| Mean chunk length | 441.57 |
| Index shards | 16 |

The release opens existing indexes read-only; the corpus cleaner and index builder are not bundled. Its subsequent sentence segmentation operates within retrieved chunks and is not the corpus-chunking procedure above. Non-overlapping corpus chunks are compatible with overlapping reading groups, which can place the same retrieved chunk in several joint-reading scopes.

## B.3 EVALUATION DATASETS

The following descriptions characterize the four QA benchmarks. The external corpus in Appendix B.1 is distinct from these QA datasets.

**HotpotQA.** HotpotQA (Yang et al., 2018) contains Wikipedia-based questions requiring information from multiple documents, including bridge and comparison questions. It provides sentence-level supporting facts alongside the answers.

**2WikiMQA.** 2WikiMQA, short for 2WikiMultiHopQA (Ho et al., 2020), combines Wikipedia text with structured information from Wikidata to construct multi-hop questions. Its evidence annotations include reasoning paths linking the question to its answer.

**MuSiQue.** MuSiQue (Trivedi et al., 2022) constructs two- to four-hop questions by composing connected single-hop questions. The construction links reasoning steps so that later steps depend on information established by earlier ones.

**NQ-Open.** NQ-Open draws on real search queries from Natural Questions (Kwiatkowski et al., 2019). The open-domain formulation retains short-answer questions and removes the supplied evidence document (Lee et al., 2019), requiring evidence to be retrieved rather than given with each question.

### B.4 Baselines

The following descriptions summarize the baseline methods used in our comparison. The BM25 context-selection baseline retains retrieved passages for answering and is distinct from the offline knowledge-base construction above.

**Vanilla.** Vanilla provides the reference answer-generation condition in Table 1.

**CoT.** Chain-of-thought prompting (Wei et al., 2022) elicits intermediate reasoning steps before the final answer. We include CoT as an additional reference condition in Table 1.

**BM25.** BM25 (Robertson et al., 1995) ranks query–passage relevance using lexical term matches, with term-frequency and document-length weighting. As a context-selection baseline, it retains whole passages rather than pruning their contents.

**Qwen3-Reranker.** Qwen3-Reranker (Zhang et al., 2025) takes a query, document, and task instruction and formulates relevance as a generative yes/no decision. The relative likelihoods of these responses provide the score for passage ranking.

**ERANK.** ERANK (Cai et al., 2026) generates fine-grained relevance scores for individual query–document pairs. It is trained by supervised fine-tuning followed by reinforcement learning with a listwise-derived ranking reward, while retaining pointwise scoring at inference.

**PROVENCE.** PROVENCE (Chirkova et al., 2025) formulates query-conditioned context pruning as sequence labeling to identify relevant sentences. It combines pruning and reranking in one model, supporting both passage ranking and within-passage content reduction.

**CPC.** CPC (Liskavets et al., 2025) learns context-aware sentence representations through contrastive training. At inference, it ranks sentences by similarity to the query, selects them under a compression budget, and restores their original order.

**EXIT.** EXIT (Hwang et al., 2025) classifies each candidate sentence using the query, its full original document, and the target sentence. Sentences passing a relevance threshold are reassembled in original order to form the compressed context.

**QUITO-X.** QUITO-X (Wang et al., 2025) uses query-conditioned encoder–decoder cross-attention to estimate token importance under an information-bottleneck formulation. The scores are aggregated and smoothed at word level before words are selected for the compressed context.

**SALT.** SALT (Mamo et al., 2026) organizes sentence keywords in a trie ordered by sentence frequency. It allocates selection budget across lexical-theme branches and scores each sentence by the additional keyword coverage it provides, accounting for what has already been selected.

### B.5 Models and Experimental Settings

Relation extraction, reranking, Organizer and Answer have separate model roles. The attention adapter uses native Qwen3.5 full-attention modules. Checkpoint locations and served-model identifiers are runtime inputs rather than fixed algorithm parameters. Complete-input accounting requires the local Answer tokenizer to match the served model.

Organizer and Answer requests both use temperature 0.0 with thinking disabled. Their output caps are 1,024 and 64 tokens, respectively. The reranker uses batch size 128 and maximum input length 512 tokens; relation-extraction forwards use SDPA. The experimental hyperparameters, including the 8,000-token complete Answer-input cap, are listed in Table 5. Exact prompts are in Appendix A.4. Parameter sweeps and component ablations vary the corresponding settings as described in Sections 4.4 and 4.5.

The initial candidate pool is distinct from the passages retained by a selection baseline or the text retained by a compressor. The 8,000-token cap applies to REORG's complete Answer input. Realized input lengths for the retrieval-scale comparison are reported in Tables 2 and 8.

### B.6 Metrics and Evaluation Protocol

**Answer metrics and aggregation.** Answer quality is reported as exact match (EM) and token-overlap F1. The main and retrieval-scale tables use a 0–100 scale; the case-study table explicitly uses a 0–1 scale. Cross-dataset AVG assigns equal weight to the four dataset values rather than pooling questions. EM/F1 averages are rounded to two decimal places; CR averages are rounded to one.

**Length reporting.** Table 1 retains the reported multiplicative compression factors. Its CR AVG is the arithmetic mean of four dataset factors, not a pooled token ratio. No CR is reported for the Vanilla and CoT rows. Tables 2 and 8 report the input-token counts rounded to the nearest integer for display. These counts describe downstream input length and exclude processing costs from earlier pipeline stages.

The implementation's complete Answer-input budget is a separate, explicitly defined quantity. It includes the question, instructions, chat formatting, source identifiers, exact source text and rendered relation records, as required by Equation 1. The implementation counts this complete prompt with the Answer tokenizer before generation. The case-study tokens in Table 9 also refer to complete final-answer prompts; they are not processing costs for earlier pipeline stages.

Reference answers and supporting-fact labels are evaluation-only inputs. The input loader retains the question and candidate fields for inference, not gold labels. A failed stage produces a partial output and a failure manifest rather than silently treating completed rows as a full evaluation. The execution record distinguishes completed predictions from failed instances.

# C Additional Results and Interpretation

## C.1 Full Retrieval-Scale Results

Table 8 gives all nine methods at the four candidate counts, with EM/F1 and rounded input-token counts. This experiment evaluates each method at $K \in \{10, 20, 50, 100\}$, complementing the main comparison in Table 1.

Table 8: Full reported HotpotQA comparison across retrieval scales. Each $K$ contains EM, F1, and reported input tokens (rounded to the nearest integer). Tokens are descriptive input-length quantities, not whole-pipeline processing costs or proof of matched-length comparisons.

| Method | $K = 10$ | | | $K = 20$ | | | $K = 50$ | | | $K = 100$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | EM | F1 | Tokens | EM | F1 | Tokens | EM | F1 | Tokens | EM | F1 | Tokens |
| BM25 | 35.38 | 46.98 | 1,316 | 36.88 | 48.90 | 2,553 | 38.51 | 50.80 | 6,255 | 40.28 | 53.08 | 12,413 |
| Rerank | 39.28 | 51.79 | 1,319 | 39.62 | 52.27 | 2,572 | 40.27 | 53.01 | 6,286 | 40.70 | 53.51 | 12,413 |
| ERANK | 38.14 | 50.01 | 1,320 | 38.03 | 50.41 | 2,559 | 39.89 | 52.25 | 6,258 | 40.15 | 52.71 | 12,413 |
| EXIT | 37.60 | 49.78 | 513 | 37.89 | 50.45 | 771 | 38.16 | 50.42 | 1,194 | 38.33 | 50.99 | 1,428 |
| SALT | 35.68 | 46.80 | 375 | 37.18 | 48.62 | 669 | 38.47 | 50.12 | 1,538 | 39.26 | 51.12 | 2,963 |
| PROVENCE | 38.68 | 50.50 | 398 | 39.41 | 51.26 | 600 | 39.80 | 51.72 | 972 | 40.11 | 52.13 | 1,210 |
| CPC | 38.16 | 50.42 | 1,257 | 38.14 | 50.62 | 2,133 | 36.72 | 48.83 | 2,129 | 40.62 | 52.92 | 2,125 |
| QUITO-X | 31.37 | 41.65 | 460 | 32.09 | 42.48 | 851 | 32.07 | 42.55 | 2,015 | 32.55 | 43.34 | 3,934 |
| **REORG** | 41.61 | 54.06 | 592 | 41.99 | 54.33 | 631 | 42.57 | 54.73 | 1,808 | 42.39 | 54.92 | 2,415 |

REORG has the highest reported EM and F1 at each $K$. Its F1 gap to the highest-scoring baseline, Rerank, narrows from 2.27 points at $K = 10$ to 1.41 at $K = 100$, rather than growing with the candidate pool. This is a comparison of reported scores, not a matched-length or significance claim.

## C.2 Component-Ablation Details

The four variant labels distinguish reading-scope overlap (A1), Organizer relation hints (A2), Answer-facing relation structure (A3), and group-wise versus one-shot organization (A4). A1 and A2 concern the information available during joint selection, whereas A3 concerns the expression of selected evidence. A4 changes the scope of an Organizer call. These distinctions describe the intervention targets, not matched-control or equal-cost claims.

The REORG row in Table 3 is reused from the main table, so its differences from A1–A4 are descriptive score differences, not paired component-effect estimates. The Answer renderer shows relation types and member IDs in the relation block, while evidence roles determine source ordering and are not printed as per-sentence tags. A3 removes this relation block from the Answer input.

## C.3 Parameter-Sensitivity Details

The graph dependencies determine what can be held fixed in a sensitivity comparison. Changing token-link count $k$ requires recomputing the retained token edges and their downstream sentence/passage graphs and reading groups; the sentence graph cannot generally remain fixed. Changing only $m_d$ can reuse the sentence graph when $k$, $r$ and $m_s$ are unchanged. A change to $m_s$, or to a coupled sentence/passage-neighbor setting, requires rebuilding the affected sentence sparsification and downstream objects. The plotted $m$ in Figure 3 is kept distinct from $m_s$ and $m_d$; equal parameter values do not establish that they are the same parameter.

The figure reports Answer F1 rather than graph degrees, group sizes, overlap, or Organizer workload. Answer-score variation is therefore not treated as a measurement of those intermediate quantities. The highlighted configuration provides a reference within the displayed sweeps, not a statistical robustness or optimality guarantee.

### C.4 Case Study: Preserving a Season Constraint

**Question and Reference**

HotpotQA fullwiki validation (Yang et al., 2018) Top-100

**Question.** What was the name of the storm that was the fifth hurricane of the season where Miami, Florida became the new National Hurricane Center?

**Reference answer:** Hurricane Faith ID: `5ae10572554299422ee995cd`

**Retrieved Evidence: Support and Distraction**

**A. 1966 Atlantic hurricane season** Rank 1; `D027-U004`

"During the year, the Miami, Florida, Weather Bureau office was re-designated the National Hurricane Center."

**B. Hurricane Faith** Rank 19; `D058-U001`

"Hurricane Faith was a long-lived Cape Verde hurricane and was the sixth named storm and fifth hurricane of the 1966 Atlantic hurricane season."

**C. 1948 Miami hurricane (Fox)** Rank 2

The passage names Fox and repeats "fifth hurricane of the 1948 season" (`D001-U001`, `D001-U002`). Matching "Miami" and "Florida" does not satisfy the **1966** constraint.

**Required connection.** A's title resolves "During the year"; B identifies the fifth hurricane, not the fifth named storm.

Miami office redesignation → 1966 season → fifth hurricane → Hurricane Faith

Table 9: Recorded Top-100 inputs and answers. EM/F1 use a 0–1 scale; tokens cover the complete final-answer prompt, not earlier pipeline stages.

| Rerank | CPC | REORG |
|---|---|---|
| All 100 candidates reach the reader. Both A and B are present, alongside Fox; supporting passages are not missing. | A's sentence survives, but its title and B's sentence are omitted. Neither "1966" nor "Faith" remains. Both Fox sentences are retained. | **Select:** A in `G0002`; B in `G0008`. **Recover:** exact source text by sentence ID, with A's title. **Integrate:** across groups; Fox is still retained. |
| **Answer:** Fox | **Answer:** Fox | **Answer:** Hurricane Faith |
| **EM:** 0 **F1:** 0.00 | **EM:** 0 **F1:** 0.00 | **EM:** 1 **F1:** 1.00 |
| **Tokens:** 11,577 | **Tokens:** 2,121 | **Tokens:** 3,879 |

**Interpretation.** Table 9 shows that REORG preserves the season anchor and complementary fact with less input than Rerank, but more than CPC. Its context contains 54 source-sentence units and 19 evidence groups, with no selected sentences dropped for the input budget. A and B enter through different groups, not one recorded evidence set joining them. The cards summarize excerpts, not complete prompts; the chain is a source-supported explanation, not a model reasoning trace. This single Top-100 case does not isolate component effects or establish failure at other depths or for unreported baselines.

# D Extended Related Work

## D.1 Context Compression

LLMLingua allocates budgets across prompt components and uses a small language model for iterative token compression (Jiang et al., 2023). Provence conditions sequence labeling on query–passage pairs and unifies pruning with reranking (Chirkova et al., 2025). At sentence granularity, compression can also incorporate surrounding context. CPC learns context-aware sentence representations through contrastive training on questions and positive and negative sentences. It ranks sentences by question similarity, selects them under a compression budget, and restores their original order (Liskavets et al., 2025). EXIT instead predicts each sentence's relevance from the question, the complete source document, and the target sentence, then reassembles the retained sentences in source order (Hwang et al., 2025). Both methods condition sentence selection on more than isolated sentence content: CPC does so through contextual representations, and EXIT through the classifier input. The further question posed by Evidence Entanglement is which sentences from different sources need to be considered together, rather than only whether each sentence is relevant in its original context.

Attention-guided compression illustrates why the decision unit matters as well as the scoring signal. The method of Zhao et al. (2025) uses query-to-context attention to score tokens, builds a dependency graph from self-attention, and applies maximum-spanning-tree extraction and Louvain grouping to form semantic units before pruning. The distinct AttnComp of Luo et al. (2025) aggregates query attention into instruction and document scores. Its cumulative Top-P selection starts from the instruction score and adds ranked documents until the threshold is reached or the next document falls below a minimum score. The former groups dependent tokens for retention; the latter adapts how many whole documents are retained. Attention and grouped selection are therefore already present in this literature; REORG's proposed use of attention concerns localized cross-document comparison scopes rather than the introduction of attention as a compression signal.

Compression methods also differ in the form of evidence passed to the answer model. SARA retains selected passages as natural language, represents the remaining evidence as compression vectors, and uses those vectors for iterative evidence reranking (Jin et al., 2026). Its context therefore combines explicit text with compressed representations instead of merely discarding every passage outside the text-retained subset. Attn-GS addresses personalization: a marking model aggregates attention to identify important profile sentences, and a summary model consumes the marked context to generate a compressed user profile within a token limit (Zeng et al., 2026). The attention-marking stage guides generation rather than requiring its output to be a concatenation of source sentences. These hybrid and generative designs differ from REORG's source-sentence selection and relation-reference interface, which separates decisions about evidence organization from factual wording.

## D.2 Context Restructuring

Restructuring can operate on extracted content or on documents awaiting compression. Refiner couples verbatim extraction of query-relevant content and necessary context with sectioning according to interconnectedness (Li et al., 2024). It thus already combines source-preserving extraction with organization across document chunks. EDC$^2$-RAG instead chooses a query-relevant document as a cluster root, adds documents similar to that root, removes assigned documents from the remaining pool, and repeats the process before query-aware compression within each cluster (Li et al., 2025a). The resulting groups are non-overlapping in document membership, which does not imply that their factual content is disjoint. The relevant comparison is between organizing extracted material and establishing document-level compression scopes, not between using relations and ignoring them.

Other approaches choose a structural representation or exploit a document's hierarchy. StructRAG selects among table, graph, algorithm, catalogue, and text-chunk formats using the question and document information. It reconstructs the material in that format and uses question decomposition and knowledge extraction to reason over the structured evidence (Li et al., 2025b). LongRefiner combines local–global query analysis with a hierarchy of sections, subsections, and paragraphs. It scores hierarchy nodes from local and global perspectives and selects content within a token budget (Jin et al., 2025). These approaches make the representation required by the task or the coverage required within a long document explicit. REORG's organization unit is instead a joint-reading group

built from localized associations across retrieved passages, allowing one passage to enter more than one comparison scope.

Relational organization also appears in pipelines that change retrieval itself. KG2RAG associates chunks with corpus-extracted triples offline, expands seed retrieval through their graph, and filters and arranges linked chunks using connected components and maximum spanning trees (Zhu et al., 2025). LogicRAG instead builds a directed acyclic graph of query subproblems at inference time, updates it when retrieval is insufficient, and resolves subproblems in a topological order. It maintains retrieved information through a rolling summary (Chen et al., 2026). These methods explicitly use relations, but their graphs represent corpus facts or query dependencies. REORG operates on a given candidate set and uses local source associations to define overlapping joint-selection scopes.

This distinction is motivated by the Evidence Entanglement setting: shared background and unique facts may occur in the same passage, while different spans connect that passage to different sources. Whole-passage similarity or membership in one group does not directly determine which facts are substitutable and which must be retained together. REORG proposes locating cross-document associations on the source text before deletion, forming overlapping reading groups, and jointly selecting evidence within those groups. Association cues indicate where to compare; they do not certify redundancy or semantic relation types. The organizer decides selection and relation labels from the group text and returns source-sentence references, from which a program recovers and deduplicates the original evidence. The proposed distinction lies in localizing overlapping comparison scopes and using them for joint selection, rather than in source preservation or relation-aware organization alone.