READER-ORIENTED EVIDENCE RESHAPING: IMPROVING RAG FROM THE READER'S PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-Augmented Generation (RAG) enhances large language models by incorporating external evidence. However, existing post-retrieval processing methods remain largely evidence-centric approaches that tend to ignore how the target Reader interprets and utilizes the resulting representation. Insights from prompt engineering suggest that different representations of the same information lead to significant variations in model outputs. Therefore, we propose Reader-Oriented Evidence Reshaping (ROER), a framework designed to explicitly align the representation of evidence with the intrinsic reasoning preferences of the target Reader. Instead of relying on a fixed processing strategy, our framework represents evidence organization as executable Evidence Organization Programs (EOPs) and evolves candidate programs guided by feedback from the target Reader. These evolved programs dynamically determine how retrieved evidence is selected, structured, and presented. To support this, we introduce a multi-view retrieval module that integrates evidence from multiple perspectives, supplying diverse evolutionary materials for subsequent evidence reshaping. Experiments on CosmosQA, QuALITY, and LongBench-v2 demonstrate that ROER delivers consistent improvements over competitive baselines across short-context, document-level, and ultra-long-context reasoning settings. In particular, ROER achieves a 21.21% relative accuracy improvement over the strongest baseline in the ultra-long-context setting, highlighting its particular strength in complex long-context reasoning.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated remarkable capabilities in language understanding and reasoning (Brown et al., 2020; Wei et al., 2022; Achiam et al., 2023; Chen et al., 2024; 2025b). However, for knowledge-intensive tasks, LLMs still suffer from insufficient parametric knowledge and a limited ability to reliably solve problems requiring external information (Lewis et al., 2020; Mallen et al., 2023; Chen et al., 2025a). Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) alleviates this limitation by retrieving relevant evidence from external knowledge sources and providing it as additional context for language models (Gao et al., 2023).

As RAG research increasingly shifts from retrieving relevant knowledge to effectively utilizing retrieved knowledge, post-retrieval evidence processing has become an important component of modern RAG systems (Gao et al., 2023; Zhao et al., 2026; Huang & Huang, 2026). Existing approaches can be broadly categorized into three groups. (1) Extractive Evidence Compression (Hwang et al., 2025; Chirkova et al., 2025; Choi et al., 2024; Jiang et al., 2024) selects and prunes tokens, sentences, or passages to preserve query-related information while reducing redundancy. (2) Abstractive Evidence Compression (Yoon et al., 2024; Kim et al., 2024; Jung et al., 2024) employs generation-based rewriting or summarization to produce compact evidence representations. (3) Structured Evidence Reconstruction (Zhu et al., 2025; Wang et al., 2025) reorganizes retrieved evidence through reordering, or explicit relation modeling. Although these approaches improve evidence compactness and information density, they mainly optimize the evidence itself and rarely consider whether the resulting representation is suitable for the target Reader (Huang & Huang, 2026; Zhao et al., 2026). In practice, the effectiveness of evidence depends not only on its intrinsic quality but also on the characteristics and reasoning requirements of the Reader model (Liu et al., 2024; Lee et al., 2025; Vladika & Matthes, 2025). Different Readers may utilize identical semantic information differently due to

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.

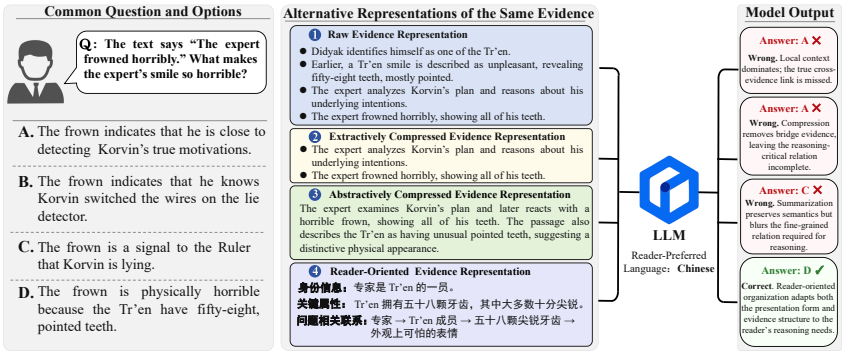


Figure 1: Evidence should be reshaped to match the reader's reasoning needs, rather than merely processed for evidence.

variations in training distributions, linguistic abilities, and reasoning behaviors (Kaplan et al., 2020; Wei et al., 2022). This observation aligns with findings in Prompt Engineering (Sahoo et al., 2024), demonstrating that modifying the presentation format can substantially affect model outputs even when the underlying semantics remain unchanged (Wei et al., 2022; Kojima et al., 2022). Therefore, evidence processing should move beyond optimizing the evidence itself and consider how to present retrieved knowledge in a form compatible with the target Reader.

Figure 1 illustrates this phenomenon. We fix the question, candidate options, and underlying evidence while changing only the forms of evidence presentation. Using ERNIE (Sun et al., 2021), a large language model that inherently favors Chinese contexts, as the Reader, original evidence, extractive compression, and abstractive compression fail to provide an effective reasoning input. In contrast, Reader-oriented reshaping enables the model to establish clearer entity-attribute relations and produce the correct prediction. This indicates that evidence utility is Reader-dependent. Specifically, effective evidence representation is determined not only by what information is provided, but also by whether the representation matches the target Reader.

Motivated by this observation, we propose **Reader-Oriented Evidence Reshaping (ROER)**, a framework that reformulates post-retrieval processing from the perspective of the Reader. Unlike existing evidence-centric methods, our framework considers the alignment between evidence representation and Reader requirements as the optimization objective. As illustrated in Figure 2, the framework consists of a Retrieval Module and a Reshape Module. The Reshape Module introduces Evidence Organization Programs (EOPs) to represent evidence transformations as executable programs and evolves them according to Reader utility. The Retrieval Module provides complementary and compact evidence spaces to support efficient evidence reshaping.

The main contributions of this work are summarized as follows:

1. Unlike existing evidence-centric post-retrieval processing methods, we introduce a Reader-oriented perspective that adapts evidence representations to the reasoning requirements of the target Reader, enabling retrieved evidence to be utilized more effectively.
2. We propose the ROER, a framework that formulates evidence organization as executable programs to dynamically construct representations explicitly aligned with the reasoning requirements of the target Reader.
3. Experiments on CosmosQA, QuALITY, and LongBench-v2 demonstrate that ROER delivers consistent improvements over competitive baselines across short-context, document-level, and ultra-long-context reasoning settings. In particular, ROER achieves a 21.21% relative accuracy improvement over the strongest baseline in the ultra-long-context setting, highlighting its particular strength in complex long-context reasoning.

2 Related Work

Post-retrieval evidence processing methods in RAG generally diverge into three distinct paradigms based on their underlying mechanisms. (1) Extractive Evidence Compression isolates relevant content from retrieved contexts. To achieve this, certain approaches evaluate information entropy via

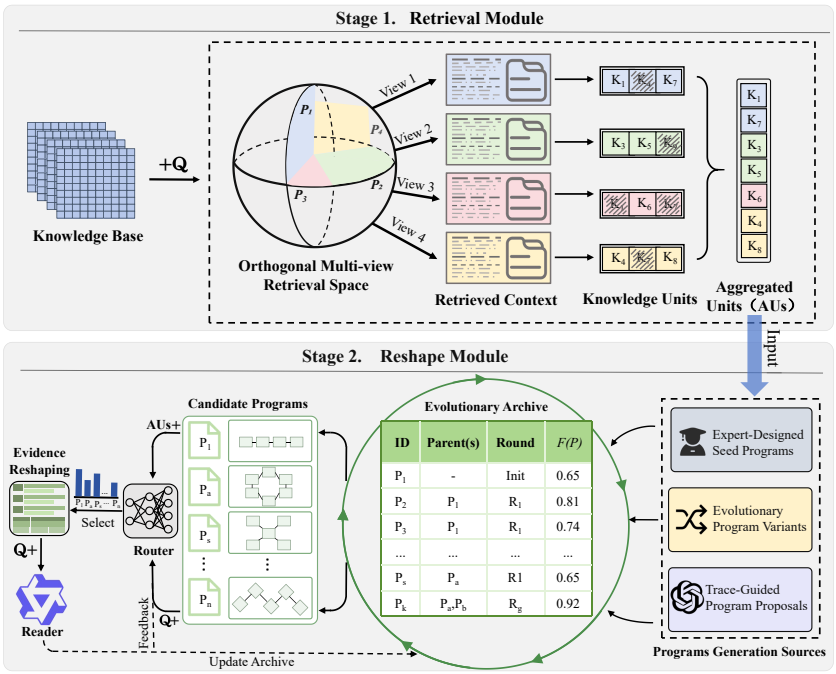


Figure 2: Overview of Reader-Oriented Evidence Reshaping for improving RAG from the Reader's perspective.

small causal language models to filter out uninformative tokens (Jiang et al., 2024). Alternative methods train dedicated span selectors with contrastive learning objectives or cross-attention mechanisms to extract query-aligned sentences (Hwang et al., 2025; Chirkova et al., 2025; Choi et al., 2024). Although these techniques can substantially reduce irrelevant or redundant information, the retained evidence remains strictly constrained by the original textual organization and may fragment critical relations required for downstream reasoning. (2) Abstractive Evidence Compression produces condensed evidence through generative rewriting or summarization. A common strategy is to perform query-aware progressive summarization to integrate scattered information across multiple documents (Yoon et al., 2024). Beyond this, other methods further reformulate retrieved contexts into more compact evidence representations through generative compression (Kim et al., 2024; Jung et al., 2024). However, such generative approaches remain susceptible to information omission, hallucination, and semantic deviation.

(3) Structured Evidence Reconstruction modifies evidence organization via explicit structural transformations. Techniques in this category typically convert flat texts into graph-based or topological representations. For instance, specific baselines dynamically construct knowledge graphs from retrieved contexts to guide reasoning (Zhu et al., 2025) or reorganize evidence chunks into specialized structural networks for enhanced information routing (Wang et al., 2025). While enhancing structural clarity, these strategies conventionally rely on predefined static topologies or task-specific organization patterns, which inherently limits their flexibility to adapt to diverse reasoning paths.

Overall, existing methods remain predominantly evidence-centric, optimizing information preservation or structure without modeling whether the output aligns with the target Reader. In contrast, Reader-Oriented Evidence Reshaping directly adapts evidence presentation to specific Reader requirements, ensuring a more compatible reasoning input.

3 Method

3.1 Problem Formulation and Overview

Given a query q, an external knowledge source $\mathcal{K}$, and a fixed target Reader $\mathcal{M}_\psi$, we formulate the problem of explicitly reorganizing retrieved knowledge to maximize downstream reasoning utility.

Different from conventional RAG frameworks that primarily optimize evidence-side properties, such as relevance, compactness, or structural organization, our objective is to discover Evidence Organization Programs (EOPs) whose induced representation maximizes the utility of the target Reader. The overall framework consists of two stages: (1) constructing a diverse evidence space through multi-view retrieval, and (2) optimizing the evidence transformation process according to Reader-derived utility. Specifically, the Retrieval Module first constructs an aggregated evidence space:

$$E_{\text{agg}} = \mathcal{R}_{\text{mv}}(q, \mathcal{K}), \tag{1}$$

where $\mathcal{R}_{\text{mv}}(\cdot)$ denotes the multi-view retrieval process and E_{agg} represents the collected evidence space. Based on the retrieved evidence, we formulate evidence organization as an executable program optimization problem. Starting from an initial program population $\mathcal{P}^{(0)}$, Reader-guided evolution explores the EOP space and maintains an evolutionary archive:

$$\mathcal{H} = \text{Evolve}(\mathcal{P}^{(0)}, \mathcal{D}, \mathcal{M}_\psi), \tag{2}$$

where $\mathcal{D}$ denotes the training dataset used for EOP evolution and selector optimization, and $\mathcal{H}$ represents the archive containing evaluated programs and their corresponding Reader utilities. After evolution, effective programs are selected from the archive to construct the final EOP bank:

$$\mathcal{B} = \text{Select}_B(\mathcal{H}), \tag{3}$$

where $\mathcal{B}$ denotes the retained EOP candidates used for subsequent program selection. For an evidence organization program $P \in \mathcal{B}$, the execution procedure applies the corresponding organizational operations to the retrieved evidence:

$$E_P = \text{Exec}(P; q, E_{\text{agg}}), \tag{4}$$

where $\text{Exec}(\cdot)$ denotes the executable transformation procedure and E_P represents the evidence representation generated by program P. During inference, a program selector predicts the most suitable organizational strategy according to the query and retrieval characteristics:

$$P^* = \arg\max_{P \in \mathcal{B}} s_\theta(P, q, E_{\text{agg}}), \tag{5}$$

where s_θ represents the predicted suitability score of each candidate program. The selected program generates the final Reader input:

$$\hat{y} = \mathcal{M}_\psi(q, E_{P^*}), \tag{6}$$

where $\hat{y}$ denotes the prediction generated by the Reader. The overall framework is summarized in Algorithm 1. Implementation details are provided in Appendix A.

3.2 Retrieval Module

The Retrieval Module aims to construct a diverse and informative evidence space that provides sufficient knowledge coverage for subsequent evidence reshaping. Instead of relying on a single semantic representation, we employ multiple complementary query views to alleviate retrieval bias caused by limited semantic perspectives. Given query q, we construct a set of semantic views:

$$\mathcal{Q} = \{q^{(1)}, ..., q^{(V)}\}, \tag{7}$$

where each view captures a different semantic perspective of the original query. The external knowledge source is segmented into retrievable evidence units:

$$\mathcal{C} = \{c_1, \ldots, c_N\}, \tag{8}$$

where each c_j denotes an individual evidence unit. For each query view and evidence unit, we compute their compatibility score:

$$s_{v,j} = \text{sim}(\phi(q^{(v)})W_v, \phi(c_j)W_v), \tag{9}$$

where $\phi(\cdot)$ denotes the shared encoder, W_v represents the projection matrix associated with view v, and $\text{sim}(\cdot, \cdot)$ denotes the similarity function. The candidate evidence retrieved from different semantic views is subsequently aggregated into the evidence space defined in Eq. 1. The Retrieval Module focuses on improving evidence diversity and coverage, while the final evidence representation is determined by the subsequent Reshape Module. Detailed retrieval implementation is provided in Appendix A.1.

3.3 RESHAPE MODULE

Given the aggregated evidence space, the Reshape Module aims to discover evidence organization strategies that better support the reasoning behavior of the target Reader. Rather than adopting a manually predefined evidence format, we formulate evidence organization as an executable program space, where each program specifies a sequence of transformations applied to retrieved evidence.

3.3.1 EVIDENCE ORGANIZATION PROGRAM

An Evidence Organization Program (EOP) represents a compositional evidence transformation strategy:

$$P = (\mathcal{S}, \eta), \tag{10}$$

where $\mathcal{S}$ denotes an ordered sequence of executable organization operators and η represents the associated program parameters. Each EOP specifies how retrieved evidence should be selected, organized, and transformed before being provided to the Reader. The operators are defined within a constrained domain-specific language (DSL), allowing each generated program to correspond to a valid executable evidence transformation. Different combinations and parameterizations of operators result in different organization behaviors, enabling the exploration of diverse Reader-oriented evidence representations. The execution procedure follows Eq. 4. The complete DSL design and operator definitions are described in Appendix A.2.

3.3.2 READER-ORIENTED EOP EVOLUTION

Different Readers interpret and utilize the same evidence representation in different ways. Therefore, we optimize the EOP space according to the downstream utility induced by the target Reader. Given a candidate program P, its utility is evaluated according to the Reader behavior induced by the transformed evidence:

$$U_\psi(P) = \mathbb{E}_{y \sim p_\psi(y|q,E_P)}[r(q,y)] - \lambda_c \Omega(E_P), \tag{11}$$

where $p_\psi(y|q, E_P)$ denotes the output distribution of the target Reader, $r(q, y)$ represents the task-dependent evaluation function, and $\Omega(E_P)$ measures the representation cost of the generated evidence. Starting from the initial program set $\mathcal{P}^{(0)}$, the evolution process iteratively generates and evaluates executable EOP candidates according to the utility defined above. The optimization maintains an archive of evaluated programs and expands the EOP space through program-level transformations, including mutation, crossover, and feedback-guided proposals. The retained high-utility programs constitute the EOP bank $\mathcal{B}$ used during inference. Detailed evolution operators and optimization procedures are provided in Appendix A.3.

3.3.3 ADAPTIVE EOP SELECTION

The evolved EOP bank contains multiple Reader-compatible evidence organization strategies. Since different queries may benefit from different representations, we introduce a program selector to adaptively select an appropriate EOP for each input. For each candidate program $P \in \mathcal{B}$, the selector predicts its suitability based on information available before evidence transformation:

$$s_\theta(P, q, E_{\text{agg}}) = f_\theta(x_P), \tag{12}$$

where x_P denotes the feature representation derived from the query, retrieval state, and program characteristics. The final program selection follows Eq. 5, enabling query-adaptive evidence reshaping for the target Reader. The selector architecture and training procedure are described in Appendix A.4.

4 THEORETICAL ANALYSIS

In this section, we analyze why evidence representation should be optimized for the target Reader and when the optimal representation may vary across Readers. For a fixed reasoning instance and retrieved evidence pool, let $E \in \Omega$ denote an admissible organization of the same underlying evidence. Let $\mathcal{L}(E; \mathcal{M}_\psi)$ denote the task-level objective used to evaluate the evidence representation E for Reader $\mathcal{M}_\psi$, where lower values indicate better performance, and assume that each Reader

Algorithm 1 Reader-Oriented Evidence Reshaping

Input: Training dataset $\mathcal{D}$, knowledge source $\mathcal{K}$, fixed Reader $\mathcal{M}_\psi$, initial EOPs $\mathcal{P}^{(0)}$
Output: EOP bank $\mathcal{B}$, program selector f_θ

Construct evidence space: $E_i^{\text{agg}} \leftarrow \mathcal{R}_{\text{mv}}(q_i, \mathcal{K})$ ▷ Details in Appendix A.1
Initialize archive: $\mathcal{H} \leftarrow \text{InitArchive}(\mathcal{P}^{(0)})$
Evolve EOPs: $\mathcal{H} \leftarrow \text{Evolve}(\mathcal{H}, \mathcal{D}, \mathcal{M}_\psi)$ ▷ Details in Appendix A.3
Construct EOP bank: $\mathcal{B} \leftarrow \text{Select}_B(\mathcal{H})$
Train selector: $f_\theta \leftarrow \text{TrainSelector}(\mathcal{B}, \mathcal{D})$ ▷ Details in Appendix A.4
Inference:
Retrieve evidence: $E_{\text{agg}} \leftarrow \mathcal{R}_{\text{mv}}(q, \mathcal{K})$
Select program: $P^* \leftarrow \arg\max_{P \in \mathcal{B}} s_\theta(P, q, E_{\text{agg}})$
Generate evidence: $E^* \leftarrow \text{Exec}(P^*; q, E_{\text{agg}})$
Predict: $\hat{y} \leftarrow \mathcal{M}_\psi(q, E^*)$
return $\hat{y}$

admits at least one optimum in Ω. The optimal evidence representation set for Reader $\mathcal{M}_\psi$ is defined as:

$$\Omega_\psi^* = \arg\min_{E \in \Omega} \mathcal{L}(E; \mathcal{M}_\psi). \tag{13}$$

Consider two Readers $\mathcal{M}_{\psi_1}$ and $\mathcal{M}_{\psi_2}$, and let $E_1^* \in \Omega_{\psi_1}^*$. For any alternative representation $E \in \Omega$, we define its optimality gap under $\mathcal{M}_{\psi_1}$ as:

$$G_{\psi_1}(E; E_1^*) = \mathcal{L}(E; \mathcal{M}_{\psi_1}) - \mathcal{L}(E_1^*; \mathcal{M}_{\psi_1}) \geq 0. \tag{14}$$

We further define the Reader-induced relative preference shift:

$$S_{\psi_1 \to \psi_2}(E; E_1^*) = \Big[\mathcal{L}(E_1^*; \mathcal{M}_{\psi_2}) - \mathcal{L}(E_1^*; \mathcal{M}_{\psi_1})\Big] - \Big[\mathcal{L}(E; \mathcal{M}_{\psi_2}) - \mathcal{L}(E; \mathcal{M}_{\psi_1})\Big]. \tag{15}$$

A positive value indicates that switching from $\mathcal{M}_{\psi_1}$ to $\mathcal{M}_{\psi_2}$ favors E relative to E_1^*.

Theorem 1 (Reader-dependent Optimal Evidence Representation)**.** Let $E_1^* \in \Omega_{\psi_1}^*$ be an optimal representation for Reader $\mathcal{M}_{\psi_1}$. If there exists an alternative representation $\widetilde{E} \in \Omega$ such that

$$S_{\psi_1 \to \psi_2}(\widetilde{E}; E_1^*) > G_{\psi_1}(\widetilde{E}; E_1^*), \tag{16}$$

then E_1^* is no longer optimal for Reader $\mathcal{M}_{\psi_2}$:

$$E_1^* \notin \Omega_{\psi_2}^*. \tag{17}$$

Since $E_1^* \in \Omega_{\psi_1}^*$, it follows that

$$\Omega_{\psi_1}^* \neq \Omega_{\psi_2}^*. \tag{18}$$

Theorem 1 establishes that the optimal representation of the same retrieved evidence can vary across Readers, providing the theoretical motivation for Reader-Oriented Evidence Reshaping.

5 EXPERIMENT

In this section, we conduct experiments to systematically evaluate the ROER from three complementary perspectives. **First**, we examine whether optimizing evidence representations with respect to the target Reader consistently improves reasoning performance over evidence-centric processing methods across different context scales and reasoning difficulties. **Second**, we investigate whether the observed improvements arise from the proposed evidence reshaping mechanism rather than simply from increased retrieval coverage or computational cost. **Third**, we analyze the Reader-dependent behavior of the framework by studying component contributions, inference efficiency, performance across different Reader models, and representative evidence-reshaping cases.

We first describe the experimental setup, including implementation details, datasets, baseline methods, and evaluation metrics. The main comparison results are then presented in Section 5.2, followed by detailed analyses in Section 5.3. Additional experimental configurations and reproducibility details are provided in Appendix C. Furthermore, the code and data of our framework are made available for research purposes[1], and all experiments run on a single NVIDIA A6000 GPU.

[1]The source code is included in the supplementary material and will be publicly released upon acceptance.

Table 1: Performance comparison of ROER against existing baselines across the CosmosQA, QuALITY, and LongBench-v2 datasets. We report the prediction accuracy (%) for individual benchmarks alongside the average accuracy. The best result in each column is highlighted in bold, with the second-best underlined.

Method	CosmosQA	QuALITY	LongBench-v2	Average
EXIT	56.50	41.50	20.00	39.33
R2C	67.50	60.50	31.50	53.17
LongLLMLingua	55.50	61.50	27.50	48.17
Provence	61.00	63.00	27.00	50.33
CompAct	60.50	43.50	32.50	45.50
SuRe	53.50	62.00	26.00	47.17
FaviComp	61.00	48.00	<u>33.00</u>	47.33
KG^2RAG	<u>69.50</u>	<u>66.00</u>	24.00	53.17
PECAN	69.00	**67.00**	26.00	<u>54.00</u>
Ours	**71.00**	65.00	**40.00**	**58.67**

5.1 Experiment Setups

Implementation Details. In the main experiments, we employ Qwen3-8B (Yang et al., 2025) as the target Reader, which remains frozen throughout all evaluations. Our framework first constructs an aggregated evidence space through multi-view retrieval, performs offline Reader-oriented EOP evolution, and then trains a pre-execution selector over the resulting EOP bank. Given that varying queries may demand distinct Reader-oriented representations, the trained selector adaptively selects a suitable EOP for each input. Inference then executes only this chosen program, thereby reducing computational overhead and accelerating the overall reasoning process.

We evaluate the proposed framework on three publicly available multiple-choice reasoning benchmarks consisting of CosmosQA (Huang et al., 2019), QuALITY (Pang et al., 2022), and LongBench-v2 (Bai et al., 2025). Following the taxonomy in Section 2, we compare representative post-retrieval evidence processing methods across three distinct paradigms: *extractive compression* (e.g., Hwang et al. (2025); Choi et al. (2024); Jiang et al. (2024); Chirkova et al. (2025)), *abstractive compression* (e.g., Yoon et al. (2024); Kim et al. (2024); Jung et al. (2024)), and *structured reconstruction* (e.g., Zhu et al. (2025); Wang et al. (2025)). Further details on the datasets, implementation settings, prompt configurations, baseline methods, and training configurations are provided in Appendix C.

Evaluation Metrics. Following (Jiang et al., 2024; Kim et al., 2024; Hwang et al., 2025), we use prediction accuracy to evaluate the individual datasets and the average accuracy for overall performance comparison.

5.2 Main Results

In this section, we provide a comprehensive comparison with various baseline methods on the CosmosQA, QuALITY, and LongBench-v2 datasets. Table 1 presents the experimental results. ROER demonstrates robust adaptability across different context scales, achieving the highest average accuracy of 58.67% to establish an 8.65% relative improvement over the strongest baseline, PECAN (54.00%). This highest average accuracy, coupled with securing the top performance on two of the three individual datasets, demonstrates the effectiveness of aligning evidence representations with the target Reader.

The most substantial gain emerges in the ultra-long-context setting of LongBench-v2, where our method achieves 40.00% accuracy to establish a 21.21% relative improvement over the strongest baseline, FaviComp (33.00%). LongBench-v2 presents extreme challenges characterized by overwhelming noise and scattered evidence. Under such conditions, existing methods exhibit distinct vulnerabilities: extractive compression severs inter-evidence logical flow, abstractive compression risks severe information loss, and structured reconstruction baselines fail to extract reliable topological relations from fragmented spaces. In contrast, our framework dynamically evolves programs to restructure evidence into a Reader-tailored format, significantly boosting comprehension. Further-

Table 3: Inference latency per example in seconds. Lower values denote higher efficiency. The best result in each column is highlighted in bold, with the second-best underlined.

Method	CosmosQA	QuALITY	LongBench-v2
EXIT	<u>0.39</u>	133.94	1571.63
R2C	0.58	**1.57**	47.89
LongLLMLingua	0.43	7.07	42.73
Provence	**0.30**	<u>2.57</u>	83.79
CompAct	3.59	27.28	56.32
SuRe	25.56	42.81	52.86
FaviComp	3.84	8.41	<u>15.12</u>
KG^{2}RAG	7.59	90.84	763.11
PECAN	4.55	63.95	2545.71
Ours	1.92	3.92	**13.43**

Table 4: Performance with diverse target Readers on LongBench-v2 dataset. We report accuracy (%). The best result in each column is highlighted in bold, with the second-best underlined.

Method	Qwen3	DeepSeek	GLM-4
EXIT	20.00	22.50	31.00
R2C	31.50	29.00	29.00
LongLLMLingua	27.50	30.50	27.50
Provence	27.00	27.50	23.50
CompAct	32.50	28.50	<u>34.50</u>
SuRe	26.00	25.50	23.50
FaviComp	<u>33.00</u>	32.00	26.50
KG^{2}RAG	24.00	24.00	34.00
PECAN	26.00	<u>34.00</u>	30.00
Ours	**40.00**	**36.50**	**37.00**

more, the benefits of Reader-oriented reshaping extend beyond ultra-long-context scenarios. On the short-context CosmosQA benchmark, our approach achieves the highest accuracy of 71.00%, outperforming KG^{2}RAG (69.50%). Since evidence coverage is rarely the primary bottleneck here, this gain indicates that formatting evidence to match the Reader's reasoning preferences optimizes utilization, confirming that representation compatibility is as vital as information completeness. On the document-level QuALITY dataset, our method yields 65.00% accuracy, closely trailing the structured baselines PECAN (67.00%) and KG^{2}RAG (66.00%). Because QuALITY predominantly features narrative-heavy documents with inherently well-organized entity relations, explicit structured methods can naturally exploit these native textual topologies to construct effective reasoning graphs.

Overall, these results validate that shifting from evidence-centric processing to Reader-oriented reshaping is highly effective. Our framework excels particularly in complex scenarios where evidence is structurally misaligned with the target Reader, fragmented, and obscured by noise.

5.3 Analysis

Ablation Study. To evaluate the specific contributions of the core components within our framework, we conduct ablation studies on the LongBench-v2 dataset.

Specifically, we independently remove the Reshaping Module and the Retrieval Module to isolate their respective contributions to downstream performance. As shown in Table 2, ablating the Reshaping Module reduces accuracy by an absolute 9.50%, while removing the Retrieval Module results in a smaller absolute drop of 3.50%. The substantially larger degradation associated with the Reshaping Module indicates that Reader-oriented evidence reshaping is a primary source of the overall performance gains. This observation further supports our central premise that retrieved evidence should be organized according to the requirements of the target Reader, enabling more effective evidence utilization.

Table 2: Ablation study of core components on the LongBench-v2 dataset, where absolute accuracy drops are computed against the full framework.

Variant	Accuracy(%)	Drop (%)
Ours	**40.00**	–
w/o Reshaping Module	30.50	9.50 ↓
w/o Retrieval Module	36.50	3.50 ↓

Efficiency Analysis. To evaluate the computational efficiency of our framework, we measure the average inference latency per example. As reported in Table 3, our framework achieves highly competitive inference efficiency across diverse benchmarks. Notably, it performs particularly well in the ultra-long-context setting of LongBench-v2, achieving the lowest inference latency of $13.43\,\mathrm{s}$. This computational advantage stems from the selector, which adaptively selects a suitable EOP in advance, thereby avoiding exhaustive search overhead during inference.

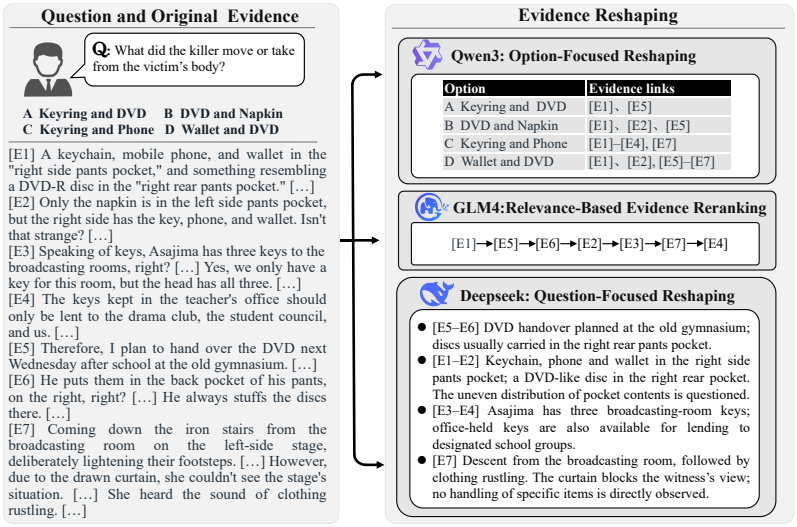


Figure 3: Case study of Reader-dependent evidence reshaping. Given the same retrieved evidence, different Readers prefer different organization strategies.

Different Reader Models. To assess the effectiveness of our framework across diverse target Readers, we further evaluate its performance on the LongBench-v2 dataset using Qwen3-8B (Yang et al., 2025), DeepSeek-8B (Guo et al., 2025), and GLM4-9B (Glm et al., 2024).

As reported in Table 4, our framework consistently achieves the best performance across all three Readers, reaching accuracies of 40.00%, 36.50%, and 37.00%, respectively. Notably, the strongest competing method shifts dramatically across different Readers: FaviComp dominates with Qwen3-8B, PECAN with DeepSeek-8B, and CompAct with GLM-4-9B. This striking variance highlights a critical limitation, demonstrating unequivocally that static, fixed evidence-centric processing strategies are far from Reader-independent. Different Readers harbor distinct structural and reasoning preferences. In contrast, our approach dynamically adapts evidence organization via evolved EOPs, successfully reconciling cross-model robustness with Reader-specific preferences and thoroughly validating our theoretical insight that optimal evidence representation inherently varies with the target Reader.

Case Study. To illustrate the Reader-dependent nature of evidence representation, we present a representative LongBench-v2 example in Figure 3. For the same question and identical retrieved raw evidence, different Readers exhibit distinct structural and organizational preferences. Specifically, in this illustrative example, Qwen3-8B prefers an option-focused representation that explicitly links each candidate option with its supporting evidence, facilitating comparative reasoning; GLM-4-9B favors a relevance-ordered representation that prioritizes evidence units most relevant to the question while preserving their original contextual relationships; and in contrast, DeepSeek-8B benefits from a compact question-focused representation that retains query-related reasoning signals while pruning irrelevant noise. These differences demonstrate that even for the same reasoning instance, the optimal evidence representation can vary across Readers, highlighting the need to adapt evidence organization to the target Reader.

6 Conclusion

In this work, we introduce ROER, a novel post-retrieval paradigm that shifts the focus from evidence-centric processing to Reader-oriented evidence reshaping. By leveraging evolved EOPs, our framework dynamically reshapes retrieved evidence to match the Reader's reasoning preferences, significantly enhancing comprehension and utilization efficiency. Theoretical insights and extensive evaluations across CosmosQA, QuALITY, and LongBench-v2 confirm that this Reader-oriented strategy delivers substantial performance gains and superior inference efficiency, particularly in ultra-long-context regimes, offering a promising perspective for post-retrieval RAG optimization.

REFERENCES

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

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, et al. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3639–3664, 2025.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Dong Chen, Yueting Zhuang, Shuo Zhang, Jinfeng Liu, Su Dong, and Siliang Tang. Data shunt: Collaboration of small and large models for lower costs and better performance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11249–11257, 2024.

Dong Chen, Fei Gao, Shuo Zhang, Yueting Zhuang, Siliang Tang, Qidong Liu, Hua Wang, Xin Yang, and Mingliang Xu. Improving large models with small models: Lower costs and better performance. *Neural Networks*, pp. 108276, 2025a.

Dong Chen, Shilin Zhang, Fei Gao, Yueting Zhuang, Siliang Tang, Qidong Liu, and Mingliang Xu. Logic distillation: Learning from code function by function for decision-making tasks. In James Kwok (ed.), *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 7338–7346. International Joint Conferences on Artificial Intelligence Organization, 8 2025b. doi: 10.24963/ijcai.2025/816. URL https://doi.org/10.24963/ijcai.2025/816. Main Track.

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*, volume 2025, pp. 38082–38106, 2025.

Eunseong Choi, Sunkyung Lee, Minjin Choi, Jun Park, and Jongwuk Lee. From reading to compressing: Exploring the multi-document reader for prompt compression. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 14734–14754, 2024.

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

Team Glm, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. 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. 2391–2401, 2019.

Yizheng Huang and Jimmy Xiangji Huang. A survey on retrieval-augmented text generation for large language models. *ACM Computing Surveys*, 58(12):1–38, 2026.

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.

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

Dongwon Jung, Qin Liu, Tenghao Huang, Ben Zhou, and Muhao Chen. Familiarity-aware evidence compression for retrieval-augmented generation. *arXiv preprint arXiv:2409.12468*, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. Sure: Summarizing retrievals using answer candidates for open-domain qa of llms. In *International Conference on Learning Representations*, volume 2024, pp. 15213–15245, 2024.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Youngwon Lee, Seung-won Hwang, Daniel F Campos, Filip Gralinski, Zhewei Yao, and Yuxiong He. Inference scaling for bridging retrieval and augmented generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 7339–7354, 2025.

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

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.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9802–9822, 2023.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, et al. Quality: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5336–5358, 2022.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 2024.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*, 2021.

Juraj Vladika and Florian Matthes. On the influence of context size and model choice in retrieval-augmented generation systems. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6739–6751, 2025.

Xinyu Wang, Yanzheng Xiang, Lin Gui, and Yulan He. Pecan: Llm-guided dynamic progress control with attention-guided hierarchical weighted graph for long-document qa. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 13317–13335, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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.

Chanwoong Yoon, Taewhoo Lee, Hyeon Hwang, Minbyul Jeong, and Jaewoo Kang. Compact: Compressing retrieved documents actively for question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21424–21439, 2024.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. Retrieval-augmented generation for ai-generated content: A survey. *Data Science and Engineering*, 11(1):1–29, 2026.

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 Framework Details

This section provides additional details of the proposed Reader-Oriented Evidence Reshaping framework. We describe the multi-view retrieval procedure, the Evidence Organization Program design, the Reader-guided evolution process, the utility formulation, and the adaptive program selection mechanism.

A.1 Multi-view Retrieval

The Retrieval Module aims to construct a diverse evidence space by integrating multiple semantic perspectives of the original query. Instead of directly optimizing a single retrieval representation, we first transform the original query into several complementary semantic views. For each query view, the retriever encodes both the query representation and candidate evidence units into a shared semantic space. The relevance scores introduced in Eq. 9 are computed between each query view and candidate evidence unit. To integrate evidence from different views, we apply a budget-constrained aggregation procedure. Specifically, the aggregation process considers both relevance scores and cross-view coverage to select complementary evidence units. The objective is to maintain sufficient semantic diversity while avoiding excessive redundancy caused by repeatedly retrieving similar evidence. The aggregated candidates from different views are merged into the evidence space used by the subsequent Reshape Module. The retrieval procedure only determines the available evidence space and does not impose any predefined organizational format. The final presentation form is determined by the evolved Evidence Organization Programs.

A.2 Evidence Organization Program

An Evidence Organization Program (EOP) represents an evidence organization strategy as an executable program defined over a constrained domain-specific language (DSL). Rather than constraining evidence organization to a single static and predefined form, the DSL decomposes it into a collection of composable operators, each capturing a distinct organization behavior. In our implementation, the admissible operator space is defined as

$$\mathcal{O} = \{o_{\text{raw}}, o_{\text{focus}}, o_{\text{option}}, o_{\text{contrast}}, o_{\text{temp-causal}}, o_{\text{source}}, o_{\text{structured}}, o_{\text{code}}, o_{\text{head-tail}}\}, \tag{19}$$

where the operators respectively preserve the retrieved evidence in its original form, emphasize question-relevant content, organize evidence around candidate options, highlight contrastive information, capture temporal and causal relations, preserve source diversity, emphasize structured or

numerical content, retain code-related evidence, and supplement local evidence with broader contextual information. A complete EOP consists of an ordered sequence of organization operators:

$$\mathcal{S} = [o_1, o_2, \ldots, o_m], \tag{20}$$

where $o_i \in \mathcal{O}$ denotes the i-th organization operator. Each operator is associated with a local parameter configuration η_i that controls its organization behavior. In addition, an EOP contains global execution parameters governing the overall evidence construction process. Accordingly, the complete parameter configuration can be expressed as:

$$\eta = (\eta_1, \eta_2, \ldots, \eta_m, B, \lambda_{\text{MMR}}), \tag{21}$$

where B denotes the maximum evidence budget and λ_{MMR} controls the relevance–redundancy trade-off during evidence selection. Together, $\mathcal{S}$ and η determine which organization behaviors are applied and how they are configured and combined.

To ensure valid execution throughout evolutionary search, each generated EOP is validated against the predefined DSL constraints before Reader evaluation. Programs containing inadmissible operators, invalid parameter configurations, or structurally invalid compositions are excluded from evaluation. This constrained representation allows the search process to explore diverse compositional evidence-organization strategies while preserving program validity.

A.3 Reader-Oriented EOP Evolution

The evolution procedure optimizes the EOP space according to the Reader utility defined in Eq. 11. Instead of directly optimizing evidence content, the evolution procedure modifies the organization strategy applied to the same retrieved evidence space. During optimization, the evolutionary archive maintains previously evaluated programs, their Reader-derived utilities, and lineage information. Specifically, each archive entry records the evaluated EOP, its corresponding utility score, and the generation history used to trace program evolution. At each evolution iteration, candidate programs are sampled from the current archive and transformed through three types of program-level operations.

Mutation. Mutation performs local modifications on an existing program while preserving DSL validity. Given an existing EOP, mutation may adjust operator parameters, replace compatible operators, insert new operators, reorder the operator sequence, or modify the evidence budget. To make this local exploration more failure-aware, diagnostic failure tags are used to bias certain modifications toward operators that are potentially relevant to the observed failure. For example, a coverage-related failure may guide the insertion or replacement of the `head_tail_context` operator, which supplements locally retrieved evidence with contextual information from the beginning and end of the original input. This mechanism enables targeted local exploration around existing effective organization strategies while maintaining program executability.

Crossover. Crossover combines complementary structures from two existing programs. The operation exchanges valid operator subsequences between parent programs and produces new executable candidates. By recombining different organization patterns, crossover enables exploration beyond the local neighborhood of existing solutions.

Reader-feedback-guided proposal. The proposal operation introduces new organization strategies according to previously observed program outcomes. Failed or suboptimal execution trajectories provide additional signals for generating alternative executable programs. Importantly, the proposal process operates only in the EOP space and does not modify Reader outputs or model parameters. The prompt used for Reader-feedback-guided EOP proposal is provided in Appendix C.5.

After candidate generation, each executable program is evaluated according to its Reader utility. The archive is subsequently updated by retaining effective programs and their lineage information. The final EOP bank is obtained by selecting high-quality programs from the optimized archive. Algorithm 2 summarizes the complete Reader-guided eop evolution process.

A.4 The EOP Selector

The evolved EOP bank contains multiple organization strategies. To enable adaptive evidence reshaping, we train a selector that predicts the most suitable program before evidence transformation.

Algorithm 2 Reader-Guided EOP Evolution

Input: Initial EOP population $\mathcal{P}^{(0)}$, training set $\mathcal{D}$, Reader $\mathcal{M}_\psi$
Output: Evolution archive $\mathcal{H}$
1: Initialize archive: $\mathcal{H} \leftarrow \mathrm{InitArchive}(\mathcal{P}^{(0)})$
2: **for** each evolution iteration **do**
3: Sample candidate programs: $\mathcal{P}_g \leftarrow \mathrm{Sample}(\mathcal{H})$
4: Generate evidence representations: $E_P \leftarrow \mathrm{Exec}(P)$
5: Evaluate Reader utility: $U_\psi(P) \leftarrow \mathrm{Evaluate}(P, \mathcal{D}, \mathcal{M}_\psi)$
6: Update archive: $\mathcal{H} \leftarrow \mathrm{Update}(\mathcal{H}, P, U_\psi(P))$
7: Generate new candidates: $\mathcal{P}_{g+1} \leftarrow \mathrm{Evolve}(\mathcal{H})$ ▷ Mutation, crossover, proposal
8: **end for**
9: **return** $\mathcal{H}$

For each candidate program, the selector receives three categories of information: (1) query-level features describing the reasoning requirements of the input query; (2) retrieval-state features characterizing the retrieved evidence space; (3) program-level descriptors representing the organization behavior of the candidate EOP. The feature representation is formulated as:

$$x_P = [f_q(q); f_e(E_{\mathrm{agg}}); d(P)], \tag{22}$$

where $f_q(\cdot)$ extracts query-related features, $f_e(\cdot)$ summarizes the characteristics of the retrieved evidence, and $d(P)$ represents the intrinsic descriptor of the candidate program. During training, the selector uses the Reader-derived utilities collected during EOP evolution as supervisory signals. These utilities indicate the relative preference among candidate programs and provide supervision for learning the suitability estimation function. At inference time, the selector evaluates candidate programs in $\mathcal{B}$ and selects the program with the highest predicted suitability according to Eq. 5.

B Proof of Theorem 1

We provide the complete proof of Theorem 1. The key idea is to characterize how switching the target Reader changes the relative preference over alternative evidence representations. Specifically, we compare the original optimality gap under $\mathcal{M}_{\psi_1}$ with the Reader-induced relative preference shift defined in Eq. 14 and Eq. 15.

For conciseness, denote

$$L_1(E) = \mathcal{L}(E; \mathcal{M}_{\psi_1}), \qquad L_2(E) = \mathcal{L}(E; \mathcal{M}_{\psi_2}). \tag{23}$$

Since $E_1^* \in \Omega_{\psi_1}^*$, it is optimal under $\mathcal{M}_{\psi_1}$, and therefore $G_{\psi_1}(E; E_1^*) \geq 0$ for every $E \in \Omega$.

We first establish the key relationship between the optimality gap and the Reader-induced relative preference shift. Using Eqs. 14 and 15, for any $E \in \Omega$ we obtain

$$\begin{aligned} & G_{\psi_1}(E; E_1^*) - S_{\psi_1 \to \psi_2}(E; E_1^*) \\ &= L_1(E) - L_1(E_1^*) - \Big([L_2(E_1^*) - L_1(E_1^*)] - [L_2(E) - L_1(E)] \Big) \\ &= L_2(E) - L_2(E_1^*). \end{aligned} \tag{24}$$

Equation 24 directly connects the two quantities in Theorem 1 to the relative performance of two evidence representations under Reader $\mathcal{M}_{\psi_2}$. Here, $G_{\psi_1}(E; E_1^*)$ measures the disadvantage of E relative to E_1^* under the original Reader $\mathcal{M}_{\psi_1}$, whereas $S_{\psi_1 \to \psi_2}(E; E_1^*)$ measures how much switching to $\mathcal{M}_{\psi_2}$ favors E relative to E_1^*. Their difference, therefore, determines whether the relative preference between the two representations remains unchanged or is reversed after switching the target Reader.

Assume that the condition in Theorem 1 holds. Then there exists an alternative representation $\widetilde{E} \in \Omega$ satisfying Eq. 16.

$$S_{\psi_1 \to \psi_2}(\widetilde{E}; E_1^*) > G_{\psi_1}(\widetilde{E}; E_1^*). \tag{25}$$

Applying Eq. 24 to $\widetilde{E}$ immediately gives:

$$L_2(\widetilde{E}) - L_2(E_1^*) = G_{\psi_1}(\widetilde{E}; E_1^*) - S_{\psi_1 \to \psi_2}(\widetilde{E}; E_1^*) < 0. \tag{26}$$

Hence,

$$L_2(\widetilde{E}) < L_2(E_1^*). \tag{27}$$

Because lower objective values indicate better performance, $\widetilde{E}$ achieves a strictly better objective value than E_1^* under Reader $\mathcal{M}_{\psi_2}$. Therefore, E_1^* cannot be an optimal evidence representation for $\mathcal{M}_{\psi_2}$, and thus

$$E_1^* \notin \Omega_{\psi_2}^*. \tag{28}$$

By construction, $E_1^* \in \Omega_{\psi_1}^*$. Hence, there exists at least one evidence representation that belongs to $\Omega_{\psi_1}^*$ but not to $\Omega_{\psi_2}^*$. The two optimal representation sets, therefore, cannot be identical:

$$\Omega_{\psi_1}^* \neq \Omega_{\psi_2}^*. \tag{29}$$

This proves that an evidence representation optimal for one Reader need not remain optimal for another. Consequently, even for the same reasoning instance and the same retrieved evidence, the optimal evidence representation can vary across Readers. This completes the proof of Theorem 1.

C Experiment Details

C.1 Dataset Statistics

We utilize the original public training split of each dataset for both EOP optimization and selector training. For evaluation, we randomly sample 200 examples from the corresponding public validation splits.

Table 5: Statistics of the evaluation datasets. Context length is measured using the Qwen3-8B tokenizer.

Dataset	Context Category	Mean Context Length
CosmosQA	Short-context	76.61
QuALITY	Document-level	5,672.18
LongBench-v2	Ultra-long-context	236,272.22

As shown in Table 5, the three datasets encompass substantially different context scales. CosmosQA features short-context reasoning, whereas QuALITY demands document-level comprehension over contexts averaging more than 5K tokens. Furthermore, LongBench-v2 introduces an ultra-long-context environment with an average length exceeding 236K tokens. This extensive span of context settings enables a comprehensive evaluation regarding the effectiveness of ROER across diverse reasoning scenarios.

C.2 Implementation Details

This section details the concrete configurations used to instantiate the Retrieval Module and the Reshape Module introduced in Section 3.

Retrieval Module. We use `BAAI/bge-small-en-v1.5` as the shared dense encoder. Each source context is segmented into evidence units of at most 180 words, with a 30-word overlap between consecutive units. The multi-view retriever uses four semantic views, each projected into a 96-dimensional latent space. The learned projection matrices remain fixed throughout all downstream experiments.

For each semantic view, we retain up to the top-32 evidence units based on view-specific relevance scores. This broad initial retrieval establishes a high-recall candidate pool to prevent the premature discarding of potentially useful evidence. The scores from different views are subsequently aggregated to obtain a global multi-view ranking. We utilize this global ranking to construct the evidence pool for subsequent EOP execution. Neighboring units are retained when available to preserve local coherence. To ensure computational efficiency, both the intermediate candidate packing and the final evidence representation passed to the Reader are strictly restricted to an $8{,}000$-character budget.

Table 6: Description of the eight initial EOPs used to initialize Reader-guided evolution. The final EOP bank is selected from the complete evolutionary archive and is not restricted to these initial programs.

Initial Program	Organization Pattern	Evidence Organization Strategy
Raw Evidence	Original preservation	Preserves top-ranked retrieved evidence with minimal transformation, establishing a baseline representation for evaluating subsequent reshaping effects.
Question-Focused Evidence	Query-driven filtering	Extracts evidence units strongly aligned with the query to systematically eliminate redundancy and enhance overall textual compactness.
Option-Aligned Evidence	Option-centric structuring	Categorizes evidence based on candidate options, allowing the Reader to efficiently contrast supporting and conflicting signals across alternatives.
Contrastive Evidence	Discriminative prioritization	Isolates evidence that actively distinguishes between competing options to highlight critical decision-making information.
Temporal-Causal Evidence	Temporal-causal alignment	Highlights sequential dependencies and causal links to facilitate multi-step reasoning over complex event transitions.
Multi-View Aligned Evidence	Cross-view integration	Aggregates diverse perspectives from distinct retrieval views into a unified and logically coherent representation.
Structured Evidence	Format-aware extraction	Prioritizes numerical, tabular, or structurally formatted evidence and aligns these elements explicitly with the reasoning objective.
Code-Aware Evidence	Code-centric organization	Foregrounds repository-level structures and code snippets alongside their surrounding contextual documentation.

Reshape Module. Each EOP is formulated as an ordered sequence of at most four executable DSL operators alongside program-level parameters. Whenever an operator selects multiple evidence units, we employ a greedy Maximal Marginal Relevance (MMR) strategy to balance operator-specific relevance against redundancy among the selected units. The MMR coefficient is fixed at 0.78 throughout all experiments, placing greater emphasis on relevance while explicitly discouraging highly redundant evidence. Furthermore, the maximum output length of any program is strictly limited to $8{,}000$ characters. Reader-oriented evolution is initialized with eight executable EOPs. These initial configurations are intentionally designed to cover substantially different organization behaviors. They serve purely as starting points for program evolution and do not constrain the diversity of the final EOP bank. Table 6 details the exact initialization programs.

The initial EOPs differ both in their prioritized evidence units and in how the retained information is organized for the Reader. Rather than functioning as a fixed program bank, they explicitly serve as starting points for Reader-guided evolution. Throughout this process, any EOP can undergo extensive modification through parameter adjustments, operator substitutions, sequence reordering, and crossover recombination.

Beyond mutation and crossover, the framework employs a Reader-feedback-guided proposal mechanism to discover new evidence-organization strategies from concrete failure cases and execution traces. Rather than directly modifying Reader outputs, the proposal model analyzes successful and unsuccessful program trajectories to identify potentially useful organization behaviors. It then generates complete DSL-compatible EOP candidates composed only of admissible organization operators. These proposed programs are subsequently evaluated by the target Reader and participate in the same evolutionary selection process as other candidate EOPs.

C.3 Training Details

Multi-view Projector Training. The multi-view projection matrices are optimized offline prior to all downstream experiments. We utilize HotpotQA (Yang et al., 2018) for this pre-training phase,

as its multi-hop questions and explicit supporting-evidence annotations provide ideal supervision for learning complementary retrieval views. Each training instance pairs a query with its designated supporting evidence alongside diverse distractor contexts. The projector is optimized using a learning rate of 2×10^{-4} and a batch size of 128 for a single epoch. Both diversity and orthogonality regularization coefficients are set to 0.05. Crucially, the shared dense encoder remains strictly frozen, and the identical learned projection checkpoint is uniformly applied across all downstream evaluation datasets.

Reader-Guided EOP Evolution. Evolution is conducted offline utilizing a maximum of 128 training examples per dataset to maintain computational efficiency. The population size is set to 24, with the top 8 programs retained at each generation based entirely on Reader utility. The procedure spans two evolutionary update rounds, yielding three evaluated populations when incorporating the initial set. Each update generates up to 12 mutation candidates and 8 crossover candidates. To drive the Reader-feedback-guided proposal operation, we retain a maximum of 12 failure cases from unsuccessful reasoning trajectories. For the multiple-choice evaluation, we instantiate the general Reader utility defined in Eq. 11 through the following computable formulation.

$$\widehat{U}_{\psi}(P) = A(P) + \lambda_f F(P) + \lambda_o O(P) + \lambda_n N(P) - \lambda_h H(P) - \lambda_l L(P), \tag{30}$$

where $A(P)$ represents downstream accuracy, $F(P)$ is the correction rate relative to the fallback EOP, and $O(P)$ captures the unique-correctness rate, measuring the proportion of examples solved exclusively by P among the current EOP population. Furthermore, $N(P)$ quantifies the net correction gain, $H(P)$ tracks the harmful-modification rate, and $L(P)$ serves as an evidence-length penalty. The target evidence length governing this penalty is $6{,}000$ characters. The corresponding coefficients are strictly configured as $\lambda_f = 0.40$, $\lambda_o = 0.35$, $\lambda_n = 0.20$, $\lambda_h = 0.60$, and $\lambda_l = 0.05$.

EOP Selector Training. The selector is parameterized as a two-layer MLP with a hidden dimension of 128 and a dropout rate of 0.15. Its input strictly follows the feature construction detailed in Appendix A.4, encompassing query-level signals, retrieval states, and static EOP descriptors. We optimize the selector for 20 epochs using the Adam optimizer with a learning rate of 5×10^{-4}, a weight decay of 10^{-3}, and a batch size of 128. A 20% validation split is reserved to determine the final selection threshold. The training objective combines pointwise utility regression with pairwise program-ranking supervision. We sample three EOP pairs per training instance and apply a pairwise loss coefficient of 0.35. By design, exhaustive program execution is strictly confined to the offline training phase to construct supervision signals for the selector. This strategy guarantees that standard inference performs purely lightweight pre-execution selection, requiring exactly one EOP execution and a single Reader call per query, thereby improving overall inference efficiency.

C.4 Baseline Details

This section details the core operational mechanisms of the evaluated baseline methods.

Extractive Evidence Compression.

- **EXIT** (Hwang et al., 2025) executes context-aware extractive compression by estimating sentence relevance while explicitly preserving document-level dependencies. The retained sentences are reassembled in their original textual order before being forwarded to the Reader.
- **R2C** (Choi et al., 2024) derives importance signals utilizing cross-attention mechanisms within a Fusion-in-Decoder architecture to conduct hierarchical prompt compression across varying context granularities.
- **LongLLMLingua** (Jiang et al., 2024) implements question-aware coarse-to-fine prompt compression. This method actively increases the density of query-aligned information by combining coarse-grained context selection with token-level compression while specifically modeling information position within long texts.
- **Provence** (Chirkova et al., 2025) formulates context pruning strictly as a sequence-labeling objective integrated with context reranking. The framework dynamically dictates the optimal proportion of retrieved context to retain for any given query.

Abstractive Evidence Compression.

- **CompAct** (Yoon et al., 2024) applies an iterative compression procedure designed to condense retrieved documents while safeguarding the essential evidence required for complex multi-hop reasoning tasks.
- **SuRe** (Kim et al., 2024) generates specific answer candidates and subsequently summarizes retrieved passages conditioned on each alternative. These candidate-conditioned summaries function as explicit reasoning rationales for downstream answer assessment.
- **FaviComp** (Jung et al., 2024) is a training-free compression method that exploits the language model's parametric familiarity to reconstruct retrieved contexts into more familiar representations. Unlike ROER, it focuses on familiarity-aware context compression rather than explicitly adapting evidence organization to the reasoning requirements of the target Reader.

Structured Evidence Reconstruction.

- **KG^{2}RAG** (Zhu et al., 2025) augments standard semantic retrieval via knowledge-graph-guided chunk expansion. The framework subsequently reorganizes the retrieved chunks according to graph-derived topological relations to produce a structurally coherent evidence representation.
- **PECAN** (Wang et al., 2025) integrates large-language-model-guided dynamic progress control with an attention-driven hierarchical weighted graph. This strategy adaptively dictates the volume of retrieved information while exploiting graph-structured dependencies to facilitate long-document question answering.

C.5 Prompt Details

This section reports the prompts used for downstream Reader evaluation and trace-guided EOP proposal. These prompts clarify the information provided to each component and support reproducibility.

Reader Prompt. Across all dataset evaluations, we employ a unified prompt for the target Reader. The Reader is provided with the evidence representation produced by the selected EOP, together with the question and candidate options, and is required to return only a single answer label. Using a fixed prompt across all datasets ensures that performance differences are attributable to the supplied evidence rather than changes in the Reader instruction.

```
You are a multiple-choice QA reader.  Use ONLY the provided evidence.
Choose the option best supported by the evidence.  Return only one
uppercase letter:  A, B, C, or D.

Evidence:
{EVIDENCE}

Question:
{QUESTION}

Options:
A. {OPTION_A}
B. {OPTION_B}
C. {OPTION_C}
D. {OPTION_D}

Answer:
```

Here, {EVIDENCE} denotes the evidence representation generated by the selected EOP. The Reader therefore makes its prediction solely based on this representation, allowing downstream performance to reflect how effectively the selected evidence organization supports the target Reader.

Reader-Feedback-Guided Proposal Prompt. During offline EOP evolution, we use GPT-5.5 as the proposal model for the Reader-feedback-guided proposal operation. Rather than relying solely on aggregate program-level performance, the proposal model is provided with concrete failure cases and their associated execution traces, enabling it to identify organization behaviors that may improve downstream Reader performance. Successful alternative traces reveal potentially beneficial

organization strategies, whereas persistent failures and harmful transformations provide signals for behaviors that may require modification or avoidance.

The proposal model is not used to answer downstream questions or generate rewritten evidence. Rather, it generates complete DSL-compatible EOP candidates composed only of admissible organization operators, thereby constraining the proposal process to the predefined EOP search space.

```
You are the proposal operator inside a persistent evolutionary search
over evidence-organization programs.
Do NOT solve the MCQs and do NOT generate evidence text.  Propose
executable DSL genotypes only.
Each proposal MUST be justified by one or more concrete failure
traces below.  Prefer small edits with a clear causal hypothesis.
Use successful alternative traces to infer what organization behavior
helped; use all-fail traces to add missing organization behavior;
avoid operators that caused harm.

Generation:
{GENERATION}

Schema:
{PROGRAM_SCHEMA}

Current elites:
{ELITE_PROGRAMS}

Failure cases / execution traces:
{FAILURE_TRACES}

Return JSON only:
{"programs":  [up to {N} programs]}
```

Here, the {GENERATION} field indicates the current evolution round, while {PROGRAM_SCHEMA} specifies the admissible organization operators and structural constraints that each proposed EOP must satisfy. The {ELITE_PROGRAMS} field provides the high-performing programs retained from the current population, serving as references for subsequent program exploration. The {FAILURE_TRACES} field contains concrete training-time failure cases and their associated execution outcomes, enabling the proposal model to identify potentially useful organization behaviors or avoid harmful ones. Finally, {N} specifies the maximum number of candidate programs returned in each proposal step. Generated proposals are subsequently validated against the DSL constraints and evaluated by the target Reader in the same manner as other candidate EOPs.