

Uncertainty-Aware Hybrid Retrieval for Long-Document RAG

Hoin Jung, Xiaoqian Wang

Elmore Family School of Electrical and Computer Engineering
Purdue University

West Lafayette, IN 47907

{jung414, joywang}@purdue.edu

Abstract

Retrieval-augmented generation (RAG) depends critically on the quality and granularity of retrieved evidence. Large retrieval units preserve context but often introduce irrelevant content, which can dilute answer-bearing evidence and worsen long context utilization. Fine-grained units are more compact, but they may be difficult to retrieve reliably because short chunks can lack semantic, lexical, or bridging cues needed to match the query. We propose **Uncertainty-aware Multi-Granularity RAG (UMG-RAG)**, a training-free hybrid retrieval framework that adapts retrieval across chunk granularities based on query-specific reliability estimates. Instead of training a new retriever or modifying the generator, UMG-RAG uses existing dense and sparse retrievers as complementary experts across multiple chunk granularities. For each query, it converts each expert-granularity score list into an evidence distribution, estimates reliability from distribution entropy, and fuses candidates according to query-specific semantic, lexical, and granularity confidence. We further introduce **UMGP-RAG**, a parent promotion variant that uses fine-grained hits to locate relevant evidence while returning broader non-redundant parent chunks for local coherence. Experiments on question answering benchmarks show that uncertainty-aware fusion and parent promotion improve generation quality in a training-free, plug-and-play retrieval pipeline.¹

1 Introduction

Retrieval-augmented generation (RAG) (Lewis et al., 2020) has become a standard framework for grounding large language models in external knowledge. In a typical pipeline, a retriever first selects relevant text units from a database, and a generator then produces an answer conditioned on the retrieved context. Although this framework reduces

the need for fully parametric knowledge storage, its effectiveness depends heavily on the granularity and precision of retrieval (Zhang et al., 2025b). When retrieved units are too large, the generator receives a mixture of relevant and irrelevant content. This can push the model into long context regimes where useful evidence may be diluted or placed in positions that are poorly utilized by the reader, often called the lost-in-the-middle phenomenon (Liu et al., 2024). When retrieved units are too small, such as isolated sentences, the retrieved evidence may be compact but difficult to recover reliably, because short units may lack semantic cues, entity aliases, or bridging context needed for matching the query. We frame this tension as a retrieval-stage granularity tradeoff between coarse units that preserve context but introduce noise and fine-grained units that provide compact evidence but risk being missed or fragmented.

Figure 1 illustrates this tradeoff. Coarse retrieval can return a long context that contains the answer but also includes substantial irrelevant information, making the generator less likely to use the answer-bearing evidence and potentially reintroducing positional bias in long-context reading. Fine-grained retrieval can avoid this noise, but the exact local evidence may be missed when the query requires surrounding information that is not present in the small unit. This suggests that neither coarse nor fine retrieval units are universally preferable. Instead, an effective retriever should be able to exploit both: fine-grained units are useful when they provide high-confidence evidence, while coarser units are useful when local context is needed to make the evidence retrievable or coherent.

Existing solutions often address only one side of this tradeoff. Long-context interventions calibrate positional attention (Hsieh et al., 2024; Jung and Wang, 2026), adjust positional encodings (Zhang et al., 2024), reorder documents (Wang et al., 2025), or compress prompts (Jiang et al., 2024a) to reduce

¹<https://github.com/HoinJung/Uncertainty-aware-Multi-Granularity-RAG>

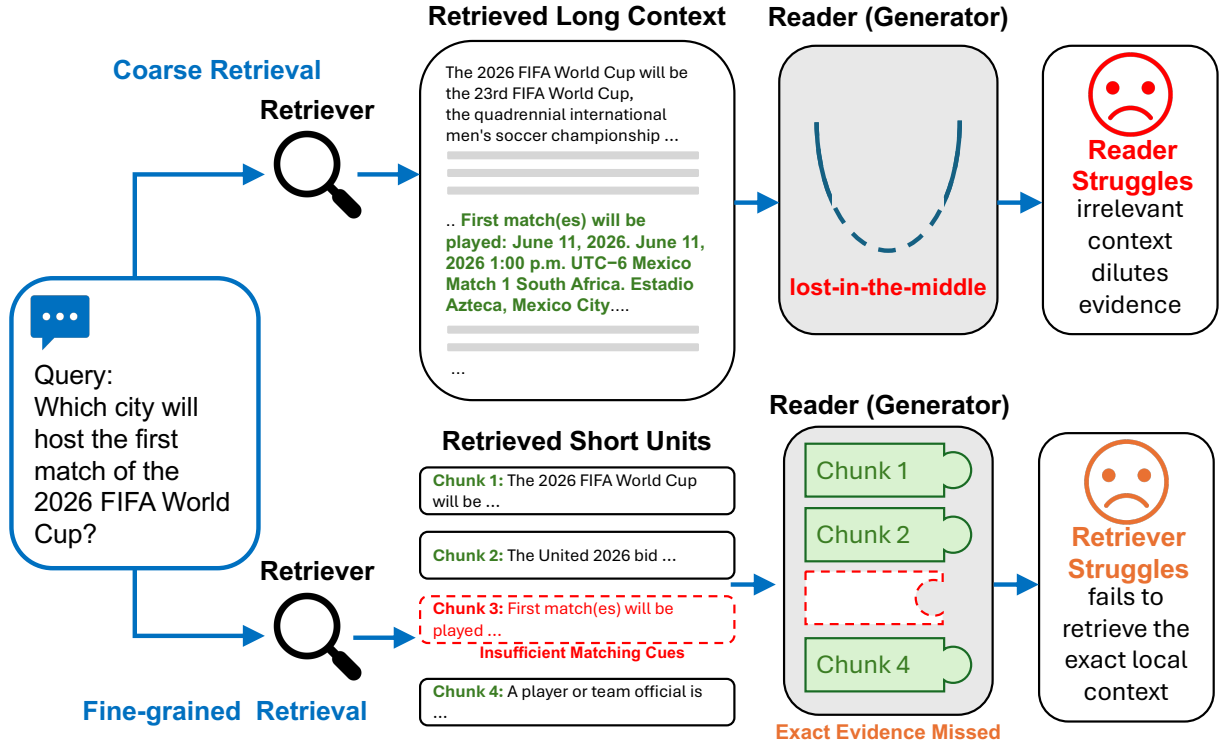


Figure 1: **Motivation of retrieval granularity tradeoff in RAG.** Coarse retrieval units preserve document-level context but may introduce irrelevant content that dilutes answer-bearing evidence. Fine-grained units are more compact, but they may lack sufficient semantic or lexical cues, causing the retriever to miss the exact local evidence.

the effect of lost-in-the-middle. These methods are valuable, but they generally assume that the relevant evidence has already been retrieved and placed somewhere in the prompt. As a result, they do not directly resolve the retrieval-stage decision of whether the system should retrieve a compact local chunk, a broader parent context, or a combination of both. Advanced RAG systems address fragmentation by constructing knowledge graphs (Guo et al., 2025), retrieving at multiple abstraction levels (Zheng et al., 2025), or inferring sentence-level relations (Tao et al., 2025). However, these methods often require graph construction, abstraction-level indexing, or post-retrieval sentence linking, and they do not explicitly estimate query-specific reliability across retrieval sources and granularities.

In this work, we propose Uncertainty-aware Multi-Granularity RAG (UMG-RAG), a training-free hybrid retrieval framework for compact long-document RAG. Instead of training a new retriever or modifying the generator, UMG-RAG uses existing dense and sparse retrievers as complementary experts. Dense retrieval captures semantic similarity, while sparse retrieval captures lexical and entity-level matching. Each expert retrieves candidates across multiple chunk granularities, and the

method estimates the confidence of each expert-granularity pair from the sharpness of its candidate score distribution. A peaked distribution indicates that the expert has a clear retrieval preference for the query, while a flat distribution indicates higher retrieval uncertainty. UMG-RAG uses this confidence to fuse dense and sparse evidence into a single ranked list, as shown in Figure 2.

We further introduce UMGP-RAG, a parent-promotion extension that improves local coherence without giving up fine-grained retrieval precision. When a highly ranked chunk comes from a fine-grained unit, UMGP-RAG promotes it to a broader parent chunk and removes redundant overlaps. This design allows fine-grained retrieval to act as a precise locator of relevant regions, while the final context returned to the generator remains locally coherent. Importantly, the framework is training-free and can be applied on top of existing dense and sparse retrieval backbones.

Our contributions are as follows:

- We formalize a retrieval granularity tradeoff in long-document RAG, where coarse units improve contextual coverage but add distracting content, while fine-grained units are compact but harder to retrieve reliably.

- We propose UMG-RAG, a training-free uncertainty-aware hybrid retrieval method that estimates query-specific reliability for each expert and granularity pair, and further introduce UMGP-RAG, a parent-promotion extension that converts fine-grained hits into broader non-redundant contexts for generation.
- We evaluate the proposed methods on long-document RAG benchmarks with multiple dense retrievers and generators, showing that the complete pipeline improves generation quality without retriever or generator training.

2 Related Work

2.1 Long Context Interventions for Lost in the Middle

Prior work has shown that language models often fail to use relevant evidence when it appears in the middle of a long prompt (Liu et al., 2024). Recent methods address this issue mainly at the reader or prompt level. FITM (Hsieh et al., 2024) calibrates positional attention bias using dummy documents. MS-PoE (Zhang et al., 2024) modifies rotary positional encodings to improve long-context utilization. PINE (Wang et al., 2025) reassigns document positions based on content importance. LongLLMLingua (Jiang et al., 2024a) compresses long prompts through question-aware pruning and reordering. These approaches are complementary to ours, but they generally assume that relevant evidence has already been retrieved. In contrast, we intervene earlier in the RAG pipeline by constructing a compact, high-signal retrieval context before generation.

2.2 Retrieval Granularity and Hybrid Retrieval in RAG

Another line of work modifies the retrieval unit or retrieval structure. LongRAG (Jiang et al., 2024b) increases the retrieval unit size to preserve document-level context, while LightRAG (Guo et al., 2025) constructs a knowledge graph to retrieve connected entities and relations. MAL-RAG (Zheng et al., 2025) retrieves from multiple abstraction levels, including summaries, paragraphs, and sentences. SAKI-RAG (Tao et al., 2025) uses sentence-level attention knowledge to recover logically connected evidence. These methods highlight the importance of retrieval granularity, but they often rely on predefined retrieval structures, graph

construction, or additional evidence-connection mechanisms.

Hybrid dense-sparse retrieval is also widely used in RAG because dense retrievers capture semantic similarity while sparse retrievers are effective for lexical and entity-level matching. Prior RAG systems such as HyPA-RAG (Kalra et al., 2024) combine dense and sparse retrieval, often using Reciprocal Rank Fusion as a simple fusion rule (Cormack et al., 2009). However, fixed fusion does not account for query-specific reliability. A lexical query may favor sparse evidence, while a paraphrastic query may favor dense evidence, and both may require different chunk granularities. Our method differs by estimating confidence for each expert-granularity pair from its score distribution, allowing dense-sparse fusion and retrieval granularity to adapt per query without training a new retriever or modifying the generator.

3 Proposed Method

3.1 Overview and Problem Setup

We propose an uncertainty-aware hybrid retrieval framework for retrieval-augmented generation. Unlike training-based methods, our framework does not introduce a new retriever, train an adapter, or modify the generator. Instead, it uses existing pre-trained dense and sparse retrievers as complementary retrieval experts and adaptively selects evidence across multiple chunk granularities without any training.

As motivated in Figure 1, RAG faces a retrieval granularity tradeoff. Coarse units preserve local context but may introduce distractors, while fine-grained units are compact but can be harder to retrieve reliably. We address this tradeoff by retrieving from multiple granularities and estimating, for each query, which retrieval source and granularity are reliable.

Let $\mathcal{D} = \{d_1, \dots, d_N\}$ denote a document corpus and let q denote an input query. Each document is segmented into overlapping sentence-level chunks with multiple granularities. We define the granularity set as $\mathcal{G} = \{2, 4, 8, 16, 32\}$, where $g \in \mathcal{G}$ denotes the number of sentences in a retrieval unit. Let $\mathcal{U}_g$ be the set of chunks generated at granularity g , and let $\mathcal{U} = \bigcup_{g \in \mathcal{G}} \mathcal{U}_g$ be the complete multi-granularity retrieval space.

We use two retrieval experts,

$$\mathcal{E} = \{\text{dense_retriever}, \text{sparse_retriever}\}.$$

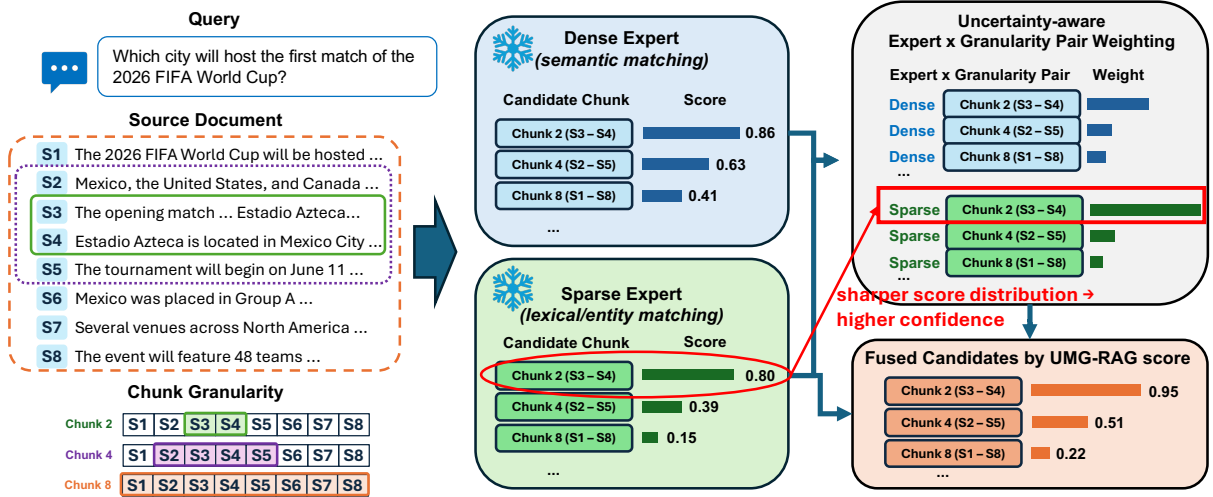


Figure 2: **Overview of uncertainty-aware multi-granularity RAG.** A source document is segmented into overlapping chunks of different granularities. Existing dense and sparse experts retrieve candidates from these granularities. UMG-RAG estimates query-specific confidence for each expert–granularity pair from the sharpness of its score distribution, then fuses the candidates into a single ranked list.

The dense expert captures semantic similarity, while the sparse expert captures lexical and entity-level matching. The framework is model-agnostic and only requires each expert to score a query-chunk pair.

For each expert $e \in \mathcal{E}$ and granularity $g \in \mathcal{G}$, the retriever assigns a raw relevance score

$$s_{e,g}(q, u) \in \mathbb{R}, \quad u \in \mathcal{U}_g.$$

A central challenge is that these scores are not directly comparable. Dense scores, sparse scores, and scores from different chunk granularities may have different scales and distributions. Therefore, fixed fusion does not explicitly account for query-specific differences in retriever and granularity reliability. A fixed weight assumes that the same retriever and granularity are equally reliable for every query, although lexical, entity-centric, and paraphrastic queries may require different retrieval behavior.

Therefore, instead of combining raw scores with a fixed dense–sparse weight, we estimate the query-specific confidence of each expert–granularity pair. This confidence is derived from the uncertainty of its candidate score distribution and determines how much that retrieval source contributes to the final context.

3.2 Uncertainty-Aware Multi-Granularity RAG

We now describe UMG-RAG. As shown in Figure 2, the method retrieves candidates from each

expert–granularity pair, converts their scores into normalized evidence distributions, estimates uncertainty using entropy, and fuses candidates into a single ranked list.

For a query q , each expert–granularity pair (e, g) first retrieves an intermediate candidate set

$$\mathcal{C}_{e,g}(q) = \text{TopM}_{u \in \mathcal{U}_g} s_{e,g}(q, u),$$

where M is the number of candidates retained from each expert–granularity pair before fusion. In our implementation, we set $M = 100$. This value is intentionally larger than the final retrieval size K so that the fusion stage can compare a sufficiently diverse set of candidates from both dense and sparse experts across granularities. At the same time, $M = 100$ keeps the intermediate pool computationally manageable, since the final context is selected only after uncertainty-aware fusion.

To compare score distributions across experts and granularities, we normalize the raw scores within each candidate set:

$$\tilde{s}_{e,g}(q, u) = \frac{s_{e,g}(q, u) - \mu_{e,g}(q)}{\sigma_{e,g}(q) + \epsilon},$$

where $\mu_{e,g}(q)$ and $\sigma_{e,g}(q)$ are the mean and standard deviation of the scores in $\mathcal{C}_{e,g}(q)$, and ϵ is a small constant for numerical stability.

We then convert the normalized scores into a score-induced evidence distribution:

$$p_{e,g}(u | q, \mathcal{C}_{e,g}) = \frac{\exp(\tilde{s}_{e,g}(q, u))}{\sum_{v \in \mathcal{C}_{e,g}(q)} \exp(\tilde{s}_{e,g}(q, v))}.$$

This distribution is defined only over the retrieved candidate set. It should not be interpreted as a calibrated probability over the full corpus. Instead, it represents how expert e at granularity g concentrates its evidence on candidate u relative to its other candidates.

This distribution provides a query-specific reliability signal. If an expert–granularity pair assigns most probability mass to a small number of candidates, it has a clear retrieval preference for the current query. If the distribution is nearly uniform, the expert–granularity pair cannot clearly distinguish among its candidates. We quantify this ambiguity using normalized entropy:

$$H_{e,g}(q) = -\frac{\sum_{u \in \mathcal{C}_{e,g}(q)} p_{e,g}(u|q, \mathcal{C}_{e,g}) \log p_{e,g}(u|q, \mathcal{C}_{e,g})}{\log |\mathcal{C}_{e,g}(q)|}.$$

The entropy value lies in $[0, 1]$. Low entropy indicates concentrated evidence and therefore low retrieval uncertainty. High entropy indicates diffuse evidence and therefore high retrieval uncertainty. This uncertainty is defined relative to the retrieved candidate list and should not be interpreted as a calibrated probability of retrieval correctness.

We convert this uncertainty into confidence:

$$c_{e,g}(q) = 1 - H_{e,g}(q).$$

The confidence values are then normalized across all experts and granularities:

$$w_{e,g}(q) = \frac{c_{e,g}(q)}{\sum_{e' \in \mathcal{E}} \sum_{g' \in \mathcal{G}} c_{e',g'}(q) + \epsilon}.$$

This weighting makes the retrieval process uncertainty-aware by assigning larger weights to expert–granularity pairs with lower entropy and smaller weights to pairs with higher entropy.

The final evidence probability of a chunk is defined as the confidence-weighted mixture of candidate distributions:

$$P(u | q) = \sum_{e \in \mathcal{E}} \sum_{g \in \mathcal{G}} w_{e,g}(q) p_{e,g}(u | q, \mathcal{C}_{e,g}) \mathbb{I}[u \in \mathcal{C}_{e,g}(q)].$$

Here, $P(u | q)$ measures the evidence mass assigned to chunk u after accounting for both retriever confidence and granularity confidence.

Because the generator should receive compact context, we rank chunks by evidence utility:

$$R(u | q) = \frac{P(u | q)}{\sqrt{\ell(u)}}.$$

Here, $\ell(u)$ is the approximate token length of u . The square-root penalty mildly favors compact

chunks while still allowing longer chunks when they receive strong evidence support.

The final retrieval context is constructed by selecting the top K chunks according to $R(u | q)$:

$$\mathcal{S}_K(q) = \text{TopK}_{u \in \mathcal{U}_{\text{cand}}} R(u | q),$$

$\mathcal{U}_{\text{cand}} = \bigcup_{e \in \mathcal{E}} \bigcup_{g \in \mathcal{G}} \mathcal{C}_{e,g}(q)$ is the fused candidate pool. The selected chunks in $\mathcal{S}_K(q)$ are concatenated and used as the retrieval context for the generator. Following common top- K RAG evaluation practice, we set $K = 5$ in the main experiments and use the same value across all retrieval methods for a fair comparison.

3.3 Parent Promotion with Overlap-Aware Deduplication

We further extend UMG-RAG with parent promotion, denoted UMGP-RAG. The goal is to use fine-grained chunks as precise retrieval signals while returning a broader local unit to the generator.

Let $\pi_8(u)$ denote the parent chunk of u at granularity 8. In our implementation, retrieved chunks from granularity 2 or 4 are promoted to their corresponding parent chunk:

$$\rho(u) = \begin{cases} \pi_8(u), & u \in \mathcal{U}_2 \cup \mathcal{U}_4, \\ u, & \text{otherwise,} \end{cases}$$

where $\rho(u)$ is the unit returned after parent promotion. Thus, fine-grained chunks serve as retrieval locators, while the returned parent chunks provide the neighboring sentences needed for local coherence.

After applying $\rho(\cdot)$, multiple fine-grained candidates may map to the same parent chunk. We merge such duplicates using the same bounded evidence aggregation used in the implementation. Let $\mathcal{M}(v) = \{u \in \mathcal{U}_{\text{cand}} : \rho(u) = v\}$ be the set of candidates mapped to promoted chunk v . We define

$$A(v | q) = 1 - \prod_{u \in \mathcal{M}(v)} (1 - P(u | q)).$$

This aggregation increases the evidence score of a parent chunk when multiple retrieved candidates support it, while keeping the score bounded in $[0, 1]$. Promoted candidates are then ranked by

$$R_P(v | q) = \frac{A(v | q)}{\sqrt{\ell(v)}}.$$

To avoid returning redundant contexts, we remove promoted candidates that substantially overlap with

already selected chunks. Let $\text{span}(v)$ denote the sentence span covered by candidate chunk v . For a candidate v and a previously selected chunk s , we define span overlap as

$$\text{overlap}(v, s) = \frac{|\text{span}(v) \cap \text{span}(s)|}{\min(|\text{span}(v)|, |\text{span}(s)|)}.$$

A candidate is skipped if at least 75% of the shorter span overlaps with any previously selected chunk. This rule removes near-duplicate or nested contexts while allowing partially overlapping chunks to remain when they may provide complementary evidence.

The final context is constructed by selecting the top K non-redundant promoted chunks:

$$\mathcal{S}_K^P(q) = \text{TopKDedup}_{v \in \{\rho(u): u \in \mathcal{U}_{\text{cand}}\}} R_P(v | q),$$

where $\mathcal{U}_{\text{cand}}$ is the fused candidate pool produced by UMG-RAG. The selected promoted chunks are concatenated and used as the final retrieval context for the generator. The detailed algorithm is described in Algorithm 1 in Appendix A.

4 Implementation Details

4.1 Models and Datasets

We evaluate our RAG pipeline on Natural Questions (NQ) (Kwiatkowski et al., 2019) and HotPotQA (Yang et al., 2018), following prior long-document RAG evaluation settings (Li et al., 2024; Petroni et al., 2021). NQ evaluates factual question answering over external knowledge, while HotPotQA requires more compositional evidence retrieval and provides a complementary testbed for multi-hop reasoning.

For each dataset, we construct a task-specific retrieval corpus from the KILT (Petroni et al., 2021) validation split. The corpus contains all Wikipedia documents appearing in the gold provenance annotations, plus 10,000 randomly sampled distractor Wikipedia documents that do not overlap with the positive set. This results in 14,274 documents for NQ, consisting of 4,274 positive documents and 10,000 distractors, and 20,519 documents for HotPotQA, consisting of 10,519 positive documents and 10,000 distractors. We use the same corpus for all retrieval methods to ensure that performance differences come from the retrieval strategy rather than the candidate document pool.

For generators, we use Qwen2.5-3B-Instruct (Yang et al., 2024) and Llama-3.2-3B-Instruct

(Grattafiori et al., 2024). For dense retrievers, we evaluate BERT (Devlin et al., 2019), BGE-M3 (Chen et al., 2024), and Qwen3-Embedding-4B (Zhang et al., 2025a). For sparse retrieval, we use SPLADEv3 (Lassance et al., 2024). All methods are evaluated without task-specific training or fine-tuning, so performance differences reflect retrieval and context construction rather than updates to the retriever or generator parameters.

4.2 Evaluation Metrics

We evaluate both retrieval quality and generation quality. For retrieval, we report answer recall at five retrieved chunks (AR@5). A retrieval is counted as successful if the normalized gold answer appears in any of the top five retrieved contexts.

For generation, we report token-level F1 and answer recall (AR). F1 measures token overlap between the generated answer and the gold answer after normalization. Generation AR measures whether the normalized gold answer appears in the generated response. We follow standard question answering normalization by lowercasing text, removing punctuation, and removing articles. Higher values indicate better performance for all metrics.

For readability, the main tables report mean performance. We additionally report standard deviations estimated from 1,000 bootstrap resamples in Appendix G.

4.3 Baselines

We compare our methods against standard and long context RAG baselines. The standard RAG baseline retrieves from a fixed retrieval unit and passes the retrieved text directly to the generator. Summarized RAG compresses retrieved documents using the Falcon summarization model (Falcon, 2023). REPLUG (Shi et al., 2024) represents a retrieval-augmented black-box language modeling baseline that combines retrieved documents without updating the generator. LongLLM-Lingua (Jiang et al., 2024a) represents prompt-compression-based context selection, while MS-PoE (Zhang et al., 2024) and PINE (Wang et al., 2025) represent long-context reader-side interventions for mitigating position bias. LongRAG (Jiang et al., 2024b) uses larger retrieval units to preserve broader context.

We also include a simple hybrid retrieval baseline. Following common hybrid retrieval practice in RAG (Kalra et al., 2024), we combine dense and sparse retrieval results using Reciprocal Rank

Retriever	Method	Retrieval AR@5 ↑	Qwen2.5-3B-Instruct		Llama-3.2-3B-Instruct	
			F1 ↑	AR ↑	F1 ↑	AR ↑
SPLADEv3	RAG	0.7994	0.4745	0.4529	0.4923	0.4741
BERT	RAG	0.4477	0.2270	0.2013	0.2414	0.2178
	Summarized RAG		0.1704	0.1491	0.1972	0.1703
	LongLLMLingua		0.2304	0.2080	0.2573	0.2429
	MS-PoE		0.2239	0.1974	0.2435	0.2192
	REPLUG		0.1373	0.1174	0.2254	0.1896
	PINE		0.1915	0.1843	0.2084	0.2097
	LongRAG	0.4441	0.2093	0.1851	0.2318	0.2094
	Hybrid	<u>0.7011</u>	0.4265	0.4018	0.4504	0.4293
	UMG-RAG (Ours)	0.6852	<u>0.4350</u>	<u>0.4110</u>	<u>0.4514</u>	<u>0.4304</u>
	UMGP-RAG (Ours)	0.7797	0.4572	0.4307	0.4811	0.4533
BGE-M3	RAG	0.8562	0.4407	0.3994	0.4941	0.4593
	Summarized RAG		0.3243	0.2880	0.3520	0.3127
	LongLLMLingua		0.4581	0.4184	0.4631	0.4462
	MS-PoE		0.4432	0.4015	0.5005	0.4709
	REPLUG		0.4701	0.4216	0.4962	0.4459
	PINE		0.4547	0.4420	0.4452	0.4603
	LongRAG	0.9101	0.4598	0.4219	<u>0.5113</u>	0.4910
	Hybrid	0.8241	<u>0.4927</u>	<u>0.4727</u>	0.4986	0.4766
	UMG-RAG (Ours)	0.8023	0.4809	0.4593	0.4873	0.4716
	UMGP-RAG (Ours)	<u>0.8759</u>	0.5052	0.4794	0.5128	<u>0.4878</u>
Qwen3- Embedding-4B	RAG	0.8601	0.4427	0.3990	0.4939	0.4621
	Summarized RAG		0.3333	0.2915	0.3537	0.3162
	LongLLMLingua		0.4459	0.4057	0.4561	0.4346
	MS-PoE		0.4481	0.4064	0.5046	0.4720
	REPLUG		0.4796	0.4367	0.5120	0.4642
	PINE		0.4628	0.4466	0.4291	0.4438
	LongRAG	0.9253	0.4744	0.4329	<u>0.5145</u>	0.4949
	Hybrid	0.8114	0.4803	<u>0.4621</u>	0.4916	0.4741
	UMG-RAG (Ours)	0.7949	<u>0.4807</u>	0.4582	0.4974	0.4790
	UMGP-RAG (Ours)	<u>0.8745</u>	0.4961	0.4748	0.5220	0.4984

Table 1: **Results on the Natural Questions dataset.** The best retrieval and generation scores within each retriever block are highlighted in **bold**, and the second best scores are underlined.

Fusion (RRF) (Cormack et al., 2009), denoted Hybrid. This baseline tests whether fixed dense-sparse fusion is sufficient. In contrast, our UMG-RAG estimates query-specific confidence for each expert and granularity before fusion. UMGP-RAG further promotes fine-grained hits to parent chunks and removes redundant overlaps. All comparison methods are used in a plug-and-play setting without additional training, to isolate the effect of retrieval strategy.

5 Results and Analysis

Tables 1 and 2 report retrieval AR@5 and generation performance with two instruction-tuned generators. Across most retriever-generator combinations, UMGP-RAG achieves the strongest or highly competitive generation performance. The BERT variants remain below the strong SPLADE-

only retriever but substantially improve over BERT-based RAG baselines, while the results with BGE-M3 and Qwen3-Embedding-4B show that the framework also benefits from stronger dense experts.

Retrieval recall and generation quality are not perfectly aligned: standard dense retrieval and LongRAG can obtain high AR@5 when an answer appears somewhere in a retrieved passage, even if the surrounding context is difficult for the generator to use. UMGP-RAG instead uses fine-grained hits as evidence locators and returns coherent parent contexts. Comparing UMG-RAG with UMGP-RAG further isolates context construction: the former performs uncertainty-aware fusion across retrievers and granularities, while the latter adds parent promotion and overlap-aware deduplication. The consistent improvement from UMG-RAG to UMGP-

Retriever	Method	Retrieval AR@5 ↑	Qwen2.5-3B-Instruct		Llama-3.2-3B-Instruct	
			F1 ↑	AR ↑	F1 ↑	AR ↑
SPLADEv3	RAG	0.6043	0.4613	0.3850	0.4798	0.3966
BERT	RAG	0.3638	0.2599	0.2087	0.2803	0.2279
	Summarized RAG		0.2275	0.1864	0.2469	0.2025
	LongLLMLingua		0.2833	0.2271	0.3021	0.2362
	MS-PoE		0.2612	0.2112	0.2860	0.2339
	REPLUG		0.1798	0.1466	0.2560	0.1984
	PINE		0.2424	0.2039	0.2557	0.2180
	LongRAG	0.3230	0.2299	0.1861	0.2625	0.2104
	Hybrid	<u>0.5207</u>	0.4081	0.3311	<u>0.4245</u>	<u>0.3479</u>
	UMG-RAG (Ours)	0.5134	<u>0.4163</u>	<u>0.3416</u>	0.4223	0.3470
	UMGP-RAG (Ours)	0.5830	0.4453	0.3645	0.4460	0.3655
BGE-M3	RAG	0.7811	0.4613	0.3798	0.4913	0.4048
	Summarized RAG		0.3856	0.3161	0.4109	0.3366
	LongLLMLingua		0.4706	0.3807	0.4668	0.3771
	MS-PoE		0.4622	0.3805	<u>0.5001</u>	0.4123
	REPLUG		0.3941	0.3150	0.4287	0.3362
	PINE		0.4442	0.3787	0.4409	0.3941
	LongRAG	<u>0.7457</u>	0.4658	0.3845	0.4799	0.3984
	Hybrid	0.6471	<u>0.4775</u>	<u>0.3975</u>	0.4986	<u>0.4175</u>
	UMG-RAG (Ours)	0.6273	0.4749	0.3957	0.4827	0.4018
	UMGP-RAG (Ours)	0.7002	0.5034	0.4216	0.5133	0.4275
Qwen3- Embedding-4B	RAG	0.7971	0.4883	0.4041	0.4991	0.4132
	Summarized RAG		0.3976	0.3261	0.4315	0.3546
	LongLLMLingua		0.4603	0.3693	0.4644	0.3738
	MS-PoE		0.4948	0.4073	0.5172	0.4295
	REPLUG		0.4075	0.3252	0.4343	0.3411
	PINE		0.4621	0.3980	0.4463	0.4027
	LongRAG	0.7280	0.4524	0.3748	0.4765	0.3961
	Hybrid	0.6091	0.4626	0.3837	0.4709	0.3904
	UMG-RAG (Ours)	0.5845	0.4622	0.3820	0.4752	0.3929
	UMGP-RAG (Ours)	0.6761	<u>0.4919</u>	<u>0.4071</u>	<u>0.5068</u>	<u>0.4200</u>

Table 2: **Results on the HotPotQA dataset.** The best retrieval and generation scores within each retriever block are highlighted in **bold**, and the second best scores are underlined.

RAG shows that converting localized evidence into coherent, non-redundant parent context is a major source of the overall gain.

Table 3 evaluates whether uncertainty-derived confidence is informative about retrieval correctness. Correct retrieval pairs receive higher confidence on average, and the highest-confidence quartile has higher Hit@5 than the lowest-confidence quartile for both dense and sparse experts. These trends support its use as a query-relative fusion signal, while the modest correlations show that it is not a calibrated correctness estimate. Appendix D provides representative dense, sparse, and overconfident-failure cases, and Appendix C isolates fusion and parent promotion under controlled inputs.

Table 4 motivates the fixed promotion target without treating it as universally optimal. The

eight-sentence parent gives the strongest retrieval for both dense retrievers and the strongest BGE-M3 generation result, while a larger parent can help a particular retriever-generator pair. Table 5 shows that the retrieval gains extend beyond the two Wikipedia QA evaluations. A controlled comparison with LLM reranking is provided in Appendix C.

Computational cost. Table 6(a) decomposes on-line latency into retrieval and context construction, generation, and total latency. UMG-RAG and UMGP-RAG incur higher retrieval-side latency because they search multiple dense and sparse indexes across granularities and fuse the resulting candidates. Their similar retrieval costs indicate that parent promotion and overlap-aware deduplication add relatively little overhead beyond multi-granularity fusion.

Scope	Pairs	Conf. hit	Conf. miss	Pearson r	Low-Q Hit@5	High-Q Hit@5	Δ
All	1,000	0.6212	0.5203	0.1327	0.7560	0.8840	+0.1280
Dense	500	0.6821	0.5747	0.1514	0.7120	0.8880	+0.1760
Sparse	500	0.5612	0.4624	0.1266	0.7520	0.9200	+0.1680

Table 3: **Confidence–correctness diagnostic.** A query–retriever pair is a hit when its top-5 candidates contain the gold answer. Uncertainty-derived confidence is higher for hits than misses, while the modest correlations show that it is informative but not calibrated. Low-Q and High-Q denote the lowest- and highest-confidence quartiles.

Target	BERT		BGE-M3	
	AR@5	LLaMA F1	AR@5	LLaMA F1
4	0.7441	0.4726	0.8470	0.5019
8	0.7797	0.4750	0.8759	0.5136
16	0.7638	0.4838	0.8604	0.5109
32	0.7106	0.4605	0.8142	0.4889

Table 4: **NQ parent-target summary.** Eight sentences provide the most stable retrieval–generation trade-off across retrievers and generators, although a 16-sentence parent is strongest for BERT–LLaMA F1. Full Qwen and LLaMA results and window-length statistics are provided in Appendix E.

Method	SciFact R@5	FiQA R@5
Standard	0.1633	0.0432
CombSUM	0.6900	0.3920
CombMNZ	0.6867	0.3920
UMG-RAG	0.6967	0.3827
UMGP-RAG	0.7400	0.4290

Table 5: **Additional retrieval benchmarks.** Retrieval-only R@5 results with BERT on SciFact and FiQA.

Table 6(b) shows that the compact contexts produced by the proposed methods reduce generation-stage memory relative to Standard RAG and several long-context baselines. The corresponding reduction in generation time does not offset the additional retrieval cost in total single-query latency. The proposed methods therefore trade retrieval-side computation and index storage for improved retrieval quality and more efficient context consumption by the generator. Offline segmentation and index construction are performed once per corpus, whereas multi-index search remains an online per-query cost.

6 Conclusion

We introduced UMG-RAG, a training-free uncertainty-aware hybrid retriever that estimates query-relative retrieval reliability, and UMGP-RAG, which returns coherent parent contexts from

(a) Online latency (s/question)			
Method	Ret./ctx.	Generation	Total
Standard RAG	0.15	0.71	0.86
Summ. RAG	0.15	0.28	0.43
LongLLMLingua	0.15	4.70	4.85
MS-PoE	0.15	0.71	0.86
REPLUG	0.15	1.01	1.16
PINE	0.15	1.03	1.18
LongRAG	0.15	0.71	0.86
Hybrid	1.59	0.27	1.86
UMG-RAG	5.16	0.30	5.46
UMGP-RAG	5.36	0.33	5.69

(b) Peak memory (MiB)		
Method	Ret./ctx.	Generation
Standard RAG	7954	7558
Summ. RAG	7954	7594
LongLLMLingua	7954	24398
MS-PoE	7954	7554
REPLUG	7954	9452
PINE	7954	7558
LongRAG	7954	7618
Hybrid	8396	6908
UMG-RAG	8396	6762
UMGP-RAG	8396	6716

Table 6: **Stage-wise online computational cost on NQ.** All methods use Qwen3-Embedding-4B and Llama-3.2-3B-Instruct. (a) Retrieval/context-construction, generation, and total online latency. (b) Peak memory measured separately for retrieval/context construction and generation. Retrieval/context construction includes fusion, parent promotion, and deduplication where applicable. Offline segmentation and index construction are excluded.

fine-grained evidence. The complete pipeline achieves strong generation performance across retrievers and generators without model updates. Additional analyses show that uncertainty-derived confidence is informative about retrieval correctness and that parent promotion contributes across fusion rules.

Limitations

Retrieval preprocessing cost. UMG-RAG and UMGP-RAG construct and search dense and sparse retrieval units at multiple granularities. This increases retrieval-side computation and index storage compared with single-granularity RAG. The corpus-side cost can be amortized when an index is reused, but the benefit depends on corpus stability and query volume. UMGP-RAG returns a compact context that reduces generation time and memory; it is not cheaper at every stage. Table 6 separates retrieval/context construction from generation.

Relative rather than calibrated uncertainty. Normalized entropy measures ambiguity within each expert–granularity candidate list rather than a calibrated probability of retrieval correctness. A concentrated distribution may still correspond to an overconfident distractor, near-duplicate candidates, or a score-scale artifact. Consequently, the derived confidence is used only as a query-relative fusion weight.

Dependence on existing retrieval experts. The proposed framework does not update the dense retriever, sparse retriever, or generator. Its performance therefore depends on the underlying experts. If both retrievers fail to score answer-bearing chunks highly, uncertainty-aware fusion cannot recover evidence absent from their candidate lists. UMG-RAG complements rather than replaces strong retrieval models.

Evaluation scope. The main evaluation uses NQ and HotPotQA corpora constructed from KILT provenance documents and sampled Wikipedia distractors. Additional retrieval-only results cover SciFact (Wadden et al., 2020) and FiQA (Maia et al., 2018), but the experiments do not fully represent web-scale, enterprise, multilingual, domain-specific, or rapidly updated collections. The fixed eight-sentence parent is also a robust empirical compromise rather than an optimum for every query or generator.

Ethical Considerations

This work aims to improve retrieval and context construction for RAG systems. The proposed method does not introduce new training data, does not finetune the retriever or generator, and does not require collecting user information. However, it relies on external corpora and pretrained retrieval

and generation models, which may contain factual errors, outdated information, or social biases. Improving retrieval quality may increase the likelihood that a model uses retrieved evidence, but it does not guarantee that the final generated answer is correct, unbiased, or safe. In real-world deployments, corpus curation, privacy protection, and output monitoring remain necessary.

Acknowledgements

This work was partially supported by Purdue’s Elmore ECE Emerging Frontiers Center, and NSF IIS #2146091, IIS #2345235.

References

- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. [Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). *Preprint*, arXiv:2402.03216.
- Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.
- Falconsai. 2023. Text summarization model. https://huggingface.co/Falconsai/text_summarization.
- Edward A. Fox and Joseph A. Shaw. 1993. [Combination of multiple searches](#). In *Text Retrieval Conference*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. [LightRAG: Simple and fast retrieval-augmented generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10746–10761, Suzhou, China. Association for Computational Linguistics.

- Cheng-Yu Hsieh, Yung-Sung Chuang, Chun-Liang Li, Zifeng Wang, Long Le, Abhishek Kumar, James Glass, Alexander Ratner, Chen-Yu Lee, Ranjay Krishna, and Tomas Pfister. 2024. [Found in the middle: Calibrating positional attention bias improves long context utilization](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14982–14995, Bangkok, Thailand. Association for Computational Linguistics.
- Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024a. 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)*, pages 1658–1677.
- Ziyan Jiang, Xueguang Ma, and Wenhui Chen. 2024b. Longrag: Enhancing retrieval-augmented generation with long-context llms. *arXiv preprint arXiv:2406.15319*.
- Hoin Jung and Xiaoqian Wang. 2026. The cost of context: Mitigating textual bias in multimodal retrieval-augmented generation. *arXiv preprint arXiv:2605.05594*.
- Rishi Kalra, Zekun Wu, Ayesha Gulley, Airlie Hilliard, Xin Guan, Adriano Koshiyama, and Philip Colin Treleaven. 2024. Hypa-rag: A hybrid parameter adaptive retrieval-augmented generation system for ai legal and policy applications. In *Proceedings of the 1st workshop on customizable nlp: Progress and challenges in customizing nlp for a domain, application, group, or individual (customnlp4u)*, pages 237–256.
- Tom Kwiakowski, 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. 2019. [Natural questions: A benchmark for question answering research](#). *Transactions of the Association for Computational Linguistics*, 7:452–466.
- Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. 2024. Splade-v3: New baselines for splade. *arXiv preprint arXiv:2403.06789*.
- 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. 2020. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.
- Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Retrieval augmented generation or long-context llms? a comprehensive study and hybrid approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 881–893.
- Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Wwv’18 open challenge: Financial opinion mining and question answering. pages 1941–1942.
- Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. [KILT: a benchmark for knowledge intensive language tasks](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.
- Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. Replug: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384.
- Wenyu Tao, Xiaofen Xing, Zeliang Li, and Xiangmin Xu. 2025. [SAKI-RAG: Mitigating context fragmentation in long-document RAG via sentence-level attention knowledge integration](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1195–1213, Suzhou, China. Association for Computational Linguistics.
- David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. [Fact or fiction: Verifying scientific claims](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.
- Ziqi Wang, Hanlin Zhang, Xiner Li, Kuan-Hao Huang, Chi Han, Shuiwang Ji, Sham M. Kakade, Hao Peng, and Heng Ji. 2025. [Eliminating position bias of language models: A mechanistic approach](#). In *The Thirteenth International Conference on Learning Representations*.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. [Qwen2 technical report](#). *Preprint*, arXiv:2407.10671.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

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. 2025a. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Zhengxuan Zhang, Yin Wu, Yuyu Luo, and Nan Tang. 2025b. Fine-grained retrieval-augmented generation for visual question answering. *arXiv preprint arXiv:2502.20964*.

Zhenyu Zhang, Runjin Chen, Shiwei Liu, Zhewei Yao, Olatunji Ruwase, Beidi Chen, Xiaoxia Wu, and Zhangyang Wang. 2024. Found in the middle: How language models use long contexts better via plug-and-play positional encoding. *Advances in Neural Information Processing Systems*, 37:60755–60775.

Zheng Zheng, Xinyi Ni, and Pengyu Hong. 2025. Multiple abstraction level retrieve augment generation. *arXiv preprint arXiv:2501.16952*.

A Detailed Algorithm of UMGP-RAG

B Reproducibility Details

Table 7 consolidates the retrieval and context-construction settings used in the main experiments. The same corpus, final top- K context count, retrieved-context prompt, and decoding configuration are used across methods; no method-specific prompt or decoding tuning is applied.

Setting	Value
Sentence granularities	$\{2, 4, 8, 16, 32\}$
Chunk overlap / stride	$50\% / g/2$
Candidates per source	$M = 100$
Final contexts	$K = 5$
Dense maximum length	512 tokens
SPLADE maximum length	256 tokens
Hybrid RRF constant	$k = 60$
Score normalization	Candidate-list z -score
Fine-grained locators	2 and 4 sentences
Promotion target	Start-aligned 8-sentence parent
Unpromoted levels	8, 16, and 32 sentences
Duplicate aggregation	Bounded evidence aggregation
Span-overlap cutoff	0.75 of the shorter span

Table 7: Consolidated retrieval and context-construction settings.

Dense and sparse scores are normalized separately within each expert–granularity top- M list

Algorithm 1 UMGP-RAG

Require: Query q , experts $\mathcal{E}$, granularities $\mathcal{G}$, $M = 100$, $K = 5$, overlap cutoff 0.75

Ensure: Selected context chunks $\mathcal{S}$

- 1: $\mathcal{U}_{\text{cand}} \leftarrow \emptyset$
- 2: **for all** $(e, g) \in \mathcal{E} \times \mathcal{G}$ **do**
- 3: $\mathcal{C}_{e,g}(q) \leftarrow \text{TopM}_{u \in \mathcal{U}_g} s_{e,g}(q, u)$
- 4: Compute $p_{e,g}(u | q, \mathcal{C}_{e,g})$ from normalized scores
- 5: Compute $H_{e,g}(q)$ and $c_{e,g}(q) = 1 - H_{e,g}(q)$
- 6: $\mathcal{U}_{\text{cand}} \leftarrow \mathcal{U}_{\text{cand}} \cup \mathcal{C}_{e,g}(q)$
- 7: **end for**
- 8: Normalize $\{c_{e,g}(q)\}$ to obtain $w_{e,g}(q)$
- 9: Compute $P(u | q)$ and $R(u | q) = P(u | q) / \sqrt{\ell(u)}$ for all $u \in \mathcal{U}_{\text{cand}}$
- 10: Sort $\mathcal{U}_{\text{cand}}$ by $R(u | q)$ in descending order
- 11: $\mathcal{U}_{\text{promoted}} \leftarrow \emptyset$, $A(v | q) \leftarrow 0$
- 12: **for all** u in ranked $\mathcal{U}_{\text{cand}}$ **do**
- 13: $v \leftarrow \pi_8(u)$ if $u \in \mathcal{U}_2 \cup \mathcal{U}_4$; otherwise $v \leftarrow u$
- 14: $\mathcal{U}_{\text{promoted}} \leftarrow \mathcal{U}_{\text{promoted}} \cup \{v\}$
- 15: $A(v | q) \leftarrow 1 - (1 - A(v | q))(1 - P(u | q))$
- 16: **end for**
- 17: Compute $R_P(v | q) = A(v | q) / \sqrt{\ell(v)}$ for all $v \in \mathcal{U}_{\text{promoted}}$
- 18: Sort $\mathcal{U}_{\text{promoted}}$ by $R_P(v | q)$ in descending order
- 19: $\mathcal{S} \leftarrow \emptyset$
- 20: **for all** v in ranked $\mathcal{U}_{\text{promoted}}$ **do**
- 21: **if** $|\mathcal{S}| = K$ **then**
- 22: **break**
- 23: **end if**
- 24: **if** $\mathcal{S} = \emptyset$ or $\max_{s \in \mathcal{S}} \text{overlap}(v, s) < 0.75$ **then**
- 25: $\mathcal{S} \leftarrow \mathcal{S} \cup \{v\}$
- 26: **end if**
- 27: **end for**
- 28: **return** $\mathcal{S}$

before the softmax evidence distribution is computed. Fine-grained chunks are deterministically mapped to the enclosing start-aligned parent window. Multiple children mapped to the same parent are combined by $1 - \prod_i (1 - P_i)$, and a candidate is discarded when at least 75% of its shorter sentence span overlaps a previously selected context.

C Controlled Fusion, Promotion, and Reranking Analyses

Table 8(a) compares equal-weight multi-granularity pooling, CombSUM and CombMNZ (Fox and Shaw, 1993), margin-based confidence, entropy fusion without the square-root length penalty, and the proposed configuration. Retrieval AR@5 is evaluated on the full NQ evaluation set, whereas LLaMA F1 is evaluated on a fixed 200-question subset shared across all methods. The 200 questions used for generation were randomly sampled and fixed before evaluating the compared methods.

The comparison supports three conclusions. Multi-granularity evidence is useful even with simple fusion; parent promotion is a major contributor across fusion rules; and the fusion stage remains

(a) Fusion and parent-promotion ablation				
Method	BERT AR@5	BERT-LLaMA F1	BGE-M3 AR@5	BGE-M3-LLaMA F1
Standard	0.4477	0.2241	0.8562	0.4847
Equal MG	0.6754	0.4462	0.8047	0.4921
CombSUM	0.6754	0.4520	0.8047	0.4931
CombMNZ	0.6983	0.4497	0.8058	0.4924
Margin confidence	0.6792	0.4543	0.8058	0.4959
Entropy, no length penalty	0.7515	0.4765	0.8421	0.5067
UMG-RAG	0.6852	0.4504	0.8023	0.4919
Equal MG + parent	0.7896	0.4880	0.8766	0.5162
CombSUM + parent	0.7913	0.4902	0.8773	0.5141
CombMNZ + parent	0.7949	0.4855	0.8763	0.5178
Margin confidence + parent	0.7649	0.4736	0.8731	0.5097
Entropy, no length penalty + parent	0.7938	0.4897	0.8689	0.5124
UMGP-RAG	0.7797	0.4824	0.8759	0.5105

(b) Comparison with an LLM reranker				
Retriever	Method	AR@5	LLaMA F1	
BERT	Standard RAG	0.4477	0.2241	
BERT	LLM reranker	0.7350	0.3954	
BERT	UMGP-RAG	0.7797	0.4824	
BGE-M3	Standard RAG	0.8562	0.4847	
BGE-M3	LLM reranker	0.7700	0.4566	
BGE-M3	UMGP-RAG	0.8759	0.5105	

Table 8: **Controlled fusion, promotion, and reranking analyses on NQ.** In both analyses, retrieval AR@5 is evaluated on the full NQ evaluation set, while LLaMA F1 is evaluated on a fixed 200-question subset. **(a)** Multi-granularity fusion substantially improves the BERT pipeline, while parent promotion provides a further gain across fusion rules. **(b)** UMGP-RAG is compared with an additional LLM reranking pass.

consequential because it changes which evidence is promoted. Several conventional fusion rules with promotion are competitive with or stronger than UMGP-RAG in this controlled comparison, so the results do not establish a unique empirical advantage for entropy weighting. The no-length-penalty configurations are also competitive, showing that the square-root penalty is an optional compactness heuristic rather than an essential source of improvement.

Table 8(b) compares UMGP-RAG with an LLM reranker. Retrieval AR@5 is evaluated on the full NQ evaluation set, while LLaMA F1 is evaluated on the same fixed 200-question subset used in Table 8(a). Reranking improves the BERT pipeline over Standard RAG but remains below UMGP-RAG in both retrieval and generation. With BGE-M3, it does not improve over the standard retriever. This comparison indicates that an additional LLM pass is not consistently beneficial across retrievers.

D Confidence–Correctness Cases

Table 9 complements the aggregate diagnostic in Table 3. The first two cases show the intended adaptive behavior, while the third shows that a concentrated score distribution can still be wrong.

E Parent Granularity and Document Length

The parent target controls the trade-off between local coherence and context expansion. Table 10(a) shows that the eight-sentence level provides substantially more context than the fine-grained locators while remaining much shorter than 16- and 32-sentence windows.

Table 11 supports eight sentences as a stable operating point rather than a universal optimum. It gives the strongest retrieval and Qwen F1 for both retrievers, whereas a 16-sentence parent is slightly stronger for BERT-LLaMA generation.

Table 10(b) shows that both corpora contain a substantial long tail: the median document is moderate, but the 90th percentile exceeds 3,000 tokens and the maximum exceeds 26,000 tokens. In contrast, the generator receives only the five selected contexts.

F Paired Significance Tests

We compute paired bootstrap 95% confidence intervals and paired randomization/sign-flip tests from per-example LLaMA outputs. Table 12 reports comparisons against Standard RAG and Hybrid

Type	Query	Answer	Dense / sparse conf.	Outcome
Dense success	first jnanpith award was an author of which language	Malayalam	0.848 / 0.424	Dense hits; sparse misses
Sparse success	who gave the famous objective correlative theory	T. S. Eliot	0.169 / 0.766	Sparse hits; dense misses
Failure	who is the leading african producer of wool	South Africa	0.873 / 0.636	Dense misses; sparse hits

Table 9: Representative uncertainty-weighting cases. The failure illustrates an overconfident distractor and motivates treating entropy-derived confidence as a query-relative signal rather than calibrated correctness.

(a) Sentence-window token lengths							
Granularity	NQ		HotPotQA		Median	P90	P95
	Median	P90	Median	P90			
2	38	65	41	71			
4	78	122	83	131			
8	162	231	164	237			
16	332	443	331	445			
32	671	867	657	846			

(b) Corpus document lengths							
Dataset	Docs.	Mean	Median	P75	P90	P95	Max
NQ	14,274	1,212.8	333	1,130	3,379	5,761	33,243
HotPotQA	20,519	1,175.7	424	1,245	3,136	5,200	26,730

Table 10: **Length statistics in whitespace tokens.** (a) Token lengths of the sentence-window granularities used for retrieval. (b) Document lengths in the cached retrieval corpora.

Target	BERT					BGE-M3				
	AR@5	Qwen EM	Qwen F1	LLaMA EM	LLaMA F1	AR@5	Qwen EM	Qwen F1	LLaMA EM	LLaMA F1
4	0.7441	0.3405	0.4514	0.3588	0.4726	0.8470	0.3676	0.4859	0.3761	0.5019
8	0.7797	0.3394	0.4590	0.3585	0.4750	0.8759	0.3846	0.5048	0.3951	0.5136
16	0.7638	0.3394	0.4562	0.3750	0.4838	0.8604	0.3655	0.4861	0.3994	0.5109
32	0.7106	0.3299	0.4378	0.3479	0.4605	0.8142	0.3595	0.4750	0.3726	0.4889

Table 11: NQ ablation over fixed promotion targets under the same top-5 context setting. Retrieval is generator-independent. Target 8 provides the most stable trade-off across retrievers and generators, while target 16 is strongest for BERT-LLaMA generation.

RRF for the BERT and BGE-M3 retrievers.

G Statistical Robustness with Bootstrap Resampling

The main results report mean retrieval and generation performance for readability. To assess statistical robustness, we additionally estimate standard deviations using 1,000 bootstrap resamples over the evaluation examples. Each bootstrap sample is drawn with replacement from the original evaluation set, and all metrics are recomputed on the resampled set. Tables 13 and 14 report the bootstrap results on Natural Questions with Qwen2.5-3B-Instruct and Llama-3.2-3B-Instruct, respectively. Tables 15 and 16 report the corresponding results on HotPotQA.

Overall, the bootstrap results show that the main trends are stable. UMGP-RAG consistently improves over UMG-RAG, supporting the benefit

of parent promotion and overlap-aware deduplication. The results also show that the strongest retrieval AR@5 does not always correspond to the strongest generation performance. For example, long-context retrieval methods can achieve high answer coverage, but the resulting context may still include distractors or place answer-bearing evidence in less favorable positions for the generator. In contrast, UMGP-RAG tends to produce stronger generation performance by constructing compact and locally coherent contexts.

The standard deviations are small relative to most observed differences. Together with the paired tests in Appendix F, these results support the full UMGP-RAG comparisons while leaving the contribution of individual fusion rules to the controlled analysis in Table 8.

Dataset	Retriever	Comparison	Base F1	UMGP F1	Δ (95% CI)	p
NQ	BERT	Standard RAG	0.2414	0.4811	+0.2397 [0.2212, 0.2585]	0.0005
NQ	BERT	Hybrid RRF	0.4504	0.4811	+0.0306 [0.0167, 0.0449]	0.0005
HotPotQA	BERT	Standard RAG	0.2803	0.4460	+0.1657 [0.1529, 0.1783]	0.0005
HotPotQA	BERT	Hybrid RRF	0.4245	0.4460	+0.0215 [0.0106, 0.0318]	0.0005
NQ	BGE-M3	Standard RAG	0.4941	0.5128	+0.0187 [0.0030, 0.0347]	0.0265
NQ	BGE-M3	Hybrid RRF	0.4986	0.5128	+0.0142 [0.0014, 0.0279]	0.0435
HotPotQA	BGE-M3	Standard RAG	0.4913	0.5133	+0.0220 [0.0109, 0.0331]	0.0010
HotPotQA	BGE-M3	Hybrid RRF	0.4986	0.5133	+0.0147 [0.0052, 0.0244]	0.0035

Table 12: Paired LLaMA F1 comparisons. All confidence intervals exclude zero.

H Computational Infrastructure and Budget

All experiments were conducted on a single NVIDIA RTX A5000 GPU with 24GB memory. Because the proposed methods are training-free and do not update retriever or generator parameters, the computational budget is reported as retrieval/context-construction time, generation time, and peak memory usage per question rather than training GPU hours.

I AI Assistant Acknowledgement

AI assistance was used to support code debugging and improve the clarity of the manuscript. The authors carefully reviewed, edited, and verified all AI-assisted content and take full responsibility for the final version of this publication.

Retriever	Method	Retrieval AR@5 $\uparrow$	Qwen2.5-3B-Instruct	
			F1 $\uparrow$	AR $\uparrow$
SPLADEv3	RAG	0.7994 $\pm$ 0.0074	0.4745 $\pm$ 0.0083	0.4529 $\pm$ 0.0094
BERT	RAG	0.4477 $\pm$ 0.0090	0.2270 $\pm$ 0.0069	0.2013 $\pm$ 0.0072
	Summarized RAG		0.1704 $\pm$ 0.0064	0.1491 $\pm$ 0.0068
	LongLLMLingua		0.2304 $\pm$ 0.0071	0.2080 $\pm$ 0.0078
	MS-PoE		0.2239 $\pm$ 0.0070	0.1974 $\pm$ 0.0074
	REPLUG		0.1373 $\pm$ 0.0059	0.1174 $\pm$ 0.0059
	PINE		0.1915 $\pm$ 0.0066	0.1843 $\pm$ 0.0072
	LongRAG	0.4441 $\pm$ 0.0093	0.2093 $\pm$ 0.0070	0.1851 $\pm$ 0.0073
	Hybrid	0.7011 $\pm$ 0.0086	0.4265 $\pm$ 0.0090	0.4018 $\pm$ 0.0093
	UMG-RAG (Ours)	0.6852 $\pm$ 0.0088	0.4350 $\pm$ 0.0084	0.4110 $\pm$ 0.0092
	UMGP-RAG (Ours)	0.7797$\pm$0.0078	0.4572$\pm$0.0087	0.4307$\pm$0.0091
BGE-M3	RAG	0.8562 $\pm$ 0.0068	0.4407 $\pm$ 0.0085	0.3994 $\pm$ 0.0089
	Summarized RAG		0.3243 $\pm$ 0.0080	0.2880 $\pm$ 0.0085
	LongLLMLingua		0.4581 $\pm$ 0.0083	0.4184 $\pm$ 0.0089
	MS-PoE		0.4432 $\pm$ 0.0083	0.4015 $\pm$ 0.0092
	REPLUG		0.4701 $\pm$ 0.0082	0.4216 $\pm$ 0.0093
	PINE		0.4547 $\pm$ 0.0084	0.4420 $\pm$ 0.0092
	LongRAG	0.9101$\pm$0.0055	0.4598 $\pm$ 0.0083	0.4219 $\pm$ 0.0093
	Hybrid	0.8241 $\pm$ 0.0071	0.4927 $\pm$ 0.0085	0.4727 $\pm$ 0.0092
	UMG-RAG (Ours)	0.8023 $\pm$ 0.0074	0.4809 $\pm$ 0.0081	0.4593 $\pm$ 0.0093
	UMGP-RAG (Ours)	0.8759 $\pm$ 0.0063	0.5052$\pm$0.0086	0.4794$\pm$0.0092
Qwen3- Embedding-4B	RAG	0.8601 $\pm$ 0.0067	0.4427 $\pm$ 0.0084	0.3990 $\pm$ 0.0092
	Summarized RAG		0.3333 $\pm$ 0.0084	0.2915 $\pm$ 0.0086
	LongLLMLingua		0.4459 $\pm$ 0.0085	0.4057 $\pm$ 0.0089
	MS-PoE		0.4481 $\pm$ 0.0083	0.4064 $\pm$ 0.0090
	REPLUG		0.4796 $\pm$ 0.0084	0.4367 $\pm$ 0.0094
	PINE		0.4628 $\pm$ 0.0084	0.4466 $\pm$ 0.0096
	LongRAG	0.9253$\pm$0.0050	0.4744 $\pm$ 0.0084	0.4329 $\pm$ 0.0093
	Hybrid	0.8114 $\pm$ 0.0071	0.4803 $\pm$ 0.0088	0.4621 $\pm$ 0.0092
	UMG-RAG (Ours)	0.7949 $\pm$ 0.0077	0.4807 $\pm$ 0.0082	0.4582 $\pm$ 0.0093
	UMGP-RAG (Ours)	0.8745 $\pm$ 0.0064	0.4961$\pm$0.0085	0.4748$\pm$0.0091

Table 13: **Results on the Natural Questions dataset (Qwen2.5-3B-Instruct).** The best retrieval and generation scores within each retriever block are highlighted in **bold**, and the second best scores are underlined.

Retriever	Method	Retrieval AR@5 $\uparrow$	Llama-3.2-3B-Instruct	
			F1 $\uparrow$	AR $\uparrow$
SPLADEv3	RAG	0.7994 $\pm$ 0.0074	0.4923 $\pm$ 0.0081	0.4741 $\pm$ 0.0097
BERT	RAG	0.4477 $\pm$ 0.0090	0.2414 $\pm$ 0.0074	0.2178 $\pm$ 0.0076
	Summarized RAG		0.1972 $\pm$ 0.0068	0.1703 $\pm$ 0.0071
	LongLLMLingua		0.2573 $\pm$ 0.0079	0.2429 $\pm$ 0.0081
	MS-PoE		0.2435 $\pm$ 0.0073	0.2192 $\pm$ 0.0076
	REPLUG		0.2254 $\pm$ 0.0071	0.1896 $\pm$ 0.0071
	PINE		0.2084 $\pm$ 0.0067	0.2097 $\pm$ 0.0074
	LongRAG	0.4441 $\pm$ 0.0093	0.2318 $\pm$ 0.0073	0.2094 $\pm$ 0.0077
	Hybrid	0.7011 $\pm$ 0.0086	0.4504 $\pm$ 0.0083	0.4293 $\pm$ 0.0092
	UMG-RAG (Ours)	0.6852 $\pm$ 0.0088	0.4514 $\pm$ 0.0080	0.4304 $\pm$ 0.0097
	UMGP-RAG (Ours)	0.7797$\pm$0.0078	0.4811$\pm$0.0086	0.4533$\pm$0.0093
BGE-M3	RAG	0.8562 $\pm$ 0.0068	0.4941 $\pm$ 0.0082	0.4593 $\pm$ 0.0094
	Summarized RAG		0.3520 $\pm$ 0.0081	0.3127 $\pm$ 0.0086
	LongLLMLingua		0.4631 $\pm$ 0.0083	0.4462 $\pm$ 0.0096
	MS-PoE		0.5005 $\pm$ 0.0086	0.4709 $\pm$ 0.0091
	REPLUG		0.4962 $\pm$ 0.0085	0.4459 $\pm$ 0.0089
	PINE		0.4452 $\pm$ 0.0080	0.4603 $\pm$ 0.0093
	LongRAG	0.9101$\pm$0.0055	0.5113 $\pm$ 0.0081	0.4910$\pm$0.0095
	Hybrid	0.8241 $\pm$ 0.0071	0.4986 $\pm$ 0.0083	0.4766 $\pm$ 0.0097
	UMG-RAG (Ours)	0.8023 $\pm$ 0.0074	0.4873 $\pm$ 0.0078	0.4716 $\pm$ 0.0095
	UMGP-RAG (Ours)	0.8759 $\pm$ 0.0063	0.5128$\pm$0.0084	0.4878 $\pm$ 0.0094
Qwen3- Embedding-4B	RAG	0.8601 $\pm$ 0.0067	0.4939 $\pm$ 0.0084	0.4621 $\pm$ 0.0093
	Summarized RAG		0.3537 $\pm$ 0.0083	0.3162 $\pm$ 0.0087
	LongLLMLingua		0.4561 $\pm$ 0.0083	0.4346 $\pm$ 0.0098
	MS-PoE		0.5046 $\pm$ 0.0083	0.4720 $\pm$ 0.0094
	REPLUG		0.5120 $\pm$ 0.0084	0.4642 $\pm$ 0.0088
	PINE		0.4291 $\pm$ 0.0082	0.4438 $\pm$ 0.0095
	LongRAG	0.9253$\pm$0.0050	0.5145 $\pm$ 0.0082	0.4949 $\pm$ 0.0096
	Hybrid	0.8114 $\pm$ 0.0071	0.4916 $\pm$ 0.0083	0.4741 $\pm$ 0.0096
	UMG-RAG (Ours)	0.7949 $\pm$ 0.0077	0.4974 $\pm$ 0.0080	0.4790 $\pm$ 0.0093
	UMGP-RAG (Ours)	0.8745 $\pm$ 0.0064	0.5220$\pm$0.0084	0.4984$\pm$0.0094

Table 14: **Results on the Natural Questions dataset (Llama-3.2-3B-Instruct).** The best retrieval and generation scores within each retriever block are highlighted in **bold**, and the second best scores are underlined.

Retriever	Method	Retrieval AR@5 ↑	Qwen2.5-3B-Instruct	
			F1 ↑	AR ↑
SPLADEv3	RAG	0.6043±0.0065	0.4613±0.0061	0.3850±0.0063
BERT	RAG	0.3638±0.0067	0.2599±0.0055	0.2087±0.0053
	Summarized RAG		0.2275±0.0053	0.1864±0.0054
	LongLLMLingua		0.2833±0.0055	0.2271±0.0056
	MS-PoE		0.2612±0.0054	0.2112±0.0054
	REPLUG		0.1798±0.0049	0.1466±0.0047
	PINE		0.2424±0.0052	0.2039±0.0053
	LongRAG	0.3230±0.0062	0.2299±0.0051	0.1861±0.0053
	Hybrid	0.5207±0.0066	0.4081±0.0062	0.3311±0.0060
	UMG-RAG (Ours)	0.5134±0.0068	0.4163±0.0062	0.3416±0.0064
	UMGP-RAG (Ours)	0.5830±0.0067	0.4453±0.0061	0.3645±0.0065
BGE-M3	RAG	0.7811±0.0057	0.4613±0.0064	0.3798±0.0065
	Summarized RAG		0.3856±0.0058	0.3161±0.0064
	LongLLMLingua		0.4706±0.0061	0.3807±0.0065
	MS-PoE		0.4622±0.0059	0.3805±0.0066
	REPLUG		0.3941±0.0060	0.3150±0.0061
	PINE		0.4442±0.0062	0.3787±0.0065
	LongRAG	0.7457±0.0059	0.4658±0.0061	0.3845±0.0065
	Hybrid	0.6471±0.0063	0.4775±0.0063	0.3975±0.0063
	UMG-RAG (Ours)	0.6273±0.0064	0.4749±0.0062	0.3957±0.0066
	UMGP-RAG (Ours)	0.7002±0.0061	0.5034±0.0062	0.4216±0.0068
Qwen3- Embedding-4B	RAG	0.7971±0.0053	0.4883±0.0060	0.4041±0.0065
	Summarized RAG		0.3976±0.0058	0.3261±0.0063
	LongLLMLingua		0.4603±0.0061	0.3693±0.0063
	MS-PoE		0.4948±0.0060	0.4073±0.0067
	REPLUG		0.4075±0.0060	0.3252±0.0063
	PINE		0.4621±0.0061	0.3980±0.0065
	LongRAG	0.7280±0.0062	0.4524±0.0061	0.3748±0.0066
	Hybrid	0.6091±0.0063	0.4626±0.0060	0.3837±0.0062
	UMG-RAG (Ours)	0.5845±0.0068	0.4622±0.0060	0.3820±0.0064
	UMGP-RAG (Ours)	0.6761±0.0064	0.4919±0.0060	0.4071±0.0066

Table 15: **Results on the HotPotQA dataset (Qwen2.5-3B-Instruct).** The best retrieval and generation scores within each retriever block are highlighted in **bold**, and the second best scores are underlined.

Retriever	Method	Retrieval AR@5 ↑	Llama-3.2-3B-Instruct	
			F1 ↑	AR ↑
SPLADEv3	RAG	0.6043±0.0065	0.4798±0.0062	0.3966±0.0069
BERT	RAG	0.3638±0.0067	0.2803±0.0055	0.2279±0.0057
	Summarized RAG		0.2469±0.0055	0.2025±0.0053
	LongLLMLingua		0.3021±0.0057	0.2362±0.0057
	MS-PoE		0.2860±0.0057	0.2339±0.0056
	REPLUG		0.2560±0.0053	0.1984±0.0052
	PINE		0.2557±0.0052	0.2180±0.0055
	LongRAG	0.3230±0.0062	0.2625±0.0056	0.2104±0.0056
	Hybrid	0.5207±0.0066	0.4245±0.0059	0.3479±0.0064
	UMG-RAG (Ours)	0.5134±0.0068	0.4223±0.0060	0.3470±0.0067
	UMGP-RAG (Ours)	0.5830±0.0067	0.4460±0.0059	0.3655±0.0063
BGE-M3	RAG	0.7811±0.0057	0.4913±0.0061	0.4048±0.0064
	Summarized RAG		0.4109±0.0060	0.3366±0.0061
	LongLLMLingua		0.4668±0.0062	0.3771±0.0062
	MS-PoE		0.5001±0.0062	0.4123±0.0068
	REPLUG		0.4287±0.0060	0.3362±0.0065
	PINE		0.4409±0.0060	0.3941±0.0065
	LongRAG	0.7457±0.0059	0.4799±0.0059	0.3984±0.0066
	Hybrid	0.6471±0.0063	0.4986±0.0060	0.4175±0.0069
	UMG-RAG (Ours)	0.6273±0.0064	0.4827±0.0059	0.4018±0.0068
	UMGP-RAG (Ours)	0.7002±0.0061	0.5133±0.0060	0.4275±0.0066
Qwen3- Embedding-4B	RAG	0.7971±0.0053	0.4991±0.0060	0.4132±0.0065
	Summarized RAG		0.4315±0.0060	0.3546±0.0061
	LongLLMLingua		0.4644±0.0060	0.3738±0.0063
	MS-PoE		0.5172±0.0059	0.4295±0.0065
	REPLUG		0.4343±0.0061	0.3411±0.0064
	PINE		0.4463±0.0059	0.4027±0.0067
	LongRAG	0.7280±0.0062	0.4765±0.0061	0.3961±0.0063
	Hybrid	0.6091±0.0063	0.4709±0.0060	0.3904±0.0066
	UMG-RAG (Ours)	0.5845±0.0068	0.4752±0.0060	0.3929±0.0068
	UMGP-RAG (Ours)	0.6761±0.0064	<u>0.5068±0.0059</u>	<u>0.4200±0.0064</u>

Table 16: **Results on the HotPotQA dataset (Llama-3.2-3B-Instruct).** The best retrieval and generation scores within each retriever block are highlighted in **bold**, and the second best scores are underlined.