SUBMODULAR EVIDENCE SELECTION FOR RETRIEVAL-AUGMENTED GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented generation (RAG) conditions a frozen generator on a k-passage subset of a larger candidate pool. Relevance-based selection suffers two failure modes that a larger budget relieves only inefficiently: facet starvation, where the top ranks over-serve one semantic dimension of the query while its remaining facets scatter at lower ranks, and higher-order redundancy, where pairwise-dissimilar passages jointly span a low-dimensional subspace and crowd out multi-hop evidence—in both cases, extra budget buys redundant copies rather than the missing evidence. We propose **Submodular Evidence Selection (SES)**, which maximizes a single monotone submodular objective combining relevance-gated coverage—a frozen language model decomposes the query into sub-queries, and direct relevance is fused multiplicatively into per-facet coverage credit, so coverage is earned only by passages that themselves answer the query—with geometric diversity, the log-determinant volume spanned by the selected embeddings, which penalizes redundancy of any order. Both normalizers are constants fixed before optimization, so greedy maximization inherits the classical $(1 - 1/e)$ guarantee exactly, with no surrogate step. Across four multi-hop QA benchmarks and three frozen generators, SES consistently improves exact match over relevance-based and diversity-aware baselines, composes cleanly with cross-encoder reranking, and adds only millisecond-scale CPU selection beyond one cacheable decomposition call per query.

1 INTRODUCTION

Retrieval-augmented generation (RAG) has become the dominant paradigm for open-domain question answering and is widely deployed in industrial systems (Wang et al., 2022). The pipeline is minimal: a retriever fetches candidate passages, and a frozen generator conditions on the top-k to produce an answer. Yet one decision inside it shapes the answer: which of the retrieved passages enter the context window.

Q: *Who won the Nobel Prize earlier, Curie or Einstein?*

Rank	Snippet	Entities	Key?
1	**Nobel Prize** laureates include **Curie** and **Einstein**... both ...	Both (Nobel)	✗
2	**Marie Curie** ... pioneering research on radioactivity ... awarded the **Nobel Prize** for ...	Curie (Nobel)	✗
3	**1903**: **Marie Curie** received the **Nobel Prize** in Physics ...	Curie (Nobel, 1903)	✓
4	**Albert Einstein** ... theoretical physicist ... won the **Nobel Prize** for ...	Einstein (Nobel)	✗
5	**1921**: **Albert Einstein** awarded the **Nobel Prize** in Physics ...	Einstein (Nobel, 1921)	✓

Figure 1: An example of facet starvation and redundancy in top-k retrieval. High-scoring passages overlap on generic Nobel Prize descriptions, while the complementary award years (1903 vs. 1921) are scattered at lower ranks.

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 illustrates why this choice matters. For the query "Who won the Nobel Prize earlier, Curie or Einstein?", the top-ranked passages are all directly relevant—they mention Nobel Prizes, Curie, and Einstein—so relevance ranking has done its job. They nevertheless fail twice over. First, a coverage failure: ranks 1 and 4 both serve the same semantic facet—generic Nobel achievements—and rank 2 again covers Curie without her award year, while the two facets the question actually demands, the award years 1903 and 1921, appear only at ranks 3 and 5; we term this facet starvation. Second, a redundancy failure: the top passages not only cover overlapping facets, they say nearly the same thing about them, so the scarce context budget is spent twice on one piece of information.

Both failure modes are, at their core, problems of marginal returns: enlarging the budget does eventually admit the missing complements, but most of the additional slots are spent on further copies of evidence already present, while the extra low-relevance passages only dilute the context window. The pattern is systematic in multi-hop reasoning, where passages about the same entity or event cluster in the top ranks while complementary facts scatter at lower ranks (Tian et al., 2025; LeVine & Varjavand, 2025); generators respond to such incompleteness by hallucinating intermediate facts or abstaining entirely (Shi et al., 2025; Sharifymoghaddam & Lin, 2026). As generators scale, the error floor of RAG is increasingly set by which evidence is presented, not by parametric knowledge.

We contend that practical evidence selection requires three properties: (i) a joint objective over the three selection factors—direct relevance, facet coverage, and redundancy control—with provable optimization guarantees; (ii) an explicit organization of these factors by the type of relation they capture, rather than their entanglement in a single similarity score or an opaque learned ranker; and (iii) training-free, deterministic, millisecond-scale selection on top of a fixed retriever–generator pipeline. The factors in (i) are of two types: direct relevance and facet coverage both characterize how a passage relates to the query, whereas redundancy control characterizes how the selected passages relate to each other. Existing methods satisfy these properties only in part.

Relevance-oriented selection—top-k truncation and cross-encoder reranking (Guo et al., 2024; Xiao et al., 2024)—scores each passage in isolation and is blind to coverage and redundancy. MMR (Carbonell & Goldstein, 1998) adds a redundancy penalty that sees one selected passage at a time and therefore misses higher-order redundancy; its greedy update can be cast as the greedy optimization of a *non-monotone* submodular objective (Lin & Bilmes, 2011), so it is not covered by the classical monotone $(1 - 1/e)$ greedy analysis of Nemhauser et al. (1978), and constant-factor approximation for non-monotone submodular maximization is a separate algorithmic problem. Determinantal selection (Kulesza et al., 2012; Li et al., 2024) exploits higher-order information through the determinant, but entangles relevance with diversity inside the kernel, and its sampling-based instantiation (k-DPPs) sacrifices determinism. GeoRAG (Zhang et al., 2026) makes semantic coverage a first-class submodular objective, yet controls redundancy only incidentally, and its deployed optimizer substitutes a Sinkhorn-based surrogate for the exact marginal gain, so the deployed algorithm falls outside its own $(1 - 1/e)$ analysis.

The factors are not symmetric: two describe query–passage relations that pull in the same direction, and one describes passage–passage relations that must be held in check. This suggests a two-term design: fuse the two query-side factors into a single *relevance-gated coverage* term, and oppose it with a *geometric diversity* term. No existing method, however, brings first-class semantic coverage together with explicit, geometry-based collective redundancy control in a single normalized submodular objective for *evidence selection under a fixed retriever–generator pipeline*. In particular, no deployed optimizer enjoys a $(1 - 1/e)$ guarantee of its own. The same three primitives—relevance, submodular coverage, and log-determinant diversity—have been combined outside this setting (Qureshi et al., 2026), but without a relevance gate that decides *which* passages may carry coverage credit, and with coverage and diversity entangled rather than separately normalized.

Submodular Evidence Selection (SES) is such a post-hoc selector atop a fixed retriever–generator pipeline (Figure 2). Its coverage term prompts a frozen language model to decompose the query into 3–5 complementary sub-queries and accumulates a saturating log-sum of per-facet coverage, where each passage's contribution is gated multiplicatively by its direct relevance to the original query (Equation (3)): a passage that matches a sub-query but is off-topic with respect to the original question contributes little, while one that both answers the query and serves an uncovered facet is doubly rewarded. Its diversity term measures the log-determinant volume spanned by the selected embeddings—a collective criterion that detects redundancy of any order, including higher-order degeneracy invisible to pairwise penalties—and is relevance-agnostic by design: it rewards geometric

novelty regardless of topicality. The balance between answering the query and avoiding collective redundancy is therefore governed by a single weight λ, with two safeguards preventing the diversity term from admitting off-topic passages.

This is not a juxtaposition of two heuristics. Each term is individually monotone submodular, so their non-negative combination is monotone submodular, and greedy maximization inherits the classical $(1 - 1/e)$ approximation guarantee (Nemhauser et al., 1978). Because the normalizers are fixed constants computed before optimization, the guarantee applies to the deployed greedy algorithm exactly, with no surrogate step; and because relevance is fused into coverage rather than appended as a third term, the three factors are optimized jointly rather than sequentially—at every greedy step, a passage that is redundant with the current set, or that serves only already-served facets, is passed over in favor of the candidate with the largest joint marginal gain. Closed-form marginal gains and incremental Cholesky updates keep the loop at $\mathcal{O}(nkm + nk^3)$.

Our contributions are threefold:

- We propose **Submodular Evidence Selection (SES)**, which maximizes a single normalized monotone submodular objective fusing relevance-gated coverage with log-determinant geometric diversity, and inherits the classical $(1 - 1/e)$ greedy guarantee (Theorem 1).
- SES admits exact, closed-form marginal gains, computable incrementally via coverage updates and Cholesky factorization; with normalizers fixed before optimization, the guarantee applies to the deployed greedy algorithm itself, with no surrogate step.
- On four multi-hop QA benchmarks and three frozen generators, SES attains the best average exact match, composes with cross-encoder reranking, and adds only millisecond-scale CPU selection beyond one cacheable decomposition call.

2 Related Work

We study evidence selection under fixed deployment: the retriever and generator are fixed, no task-specific training is permitted, and added latency must remain negligible. We organize existing methods along the three selection factors of Section 1 and highlight the gap each leaves open.

Relevance-oriented selection. The dominant practice scores each passage in isolation against the query: top-k truncation by first-stage retriever scores, and cross-encoder rerankers (Guo et al., 2024) that score query–passage pairs with a heavier encoder. Both are blind to the composition of the selected set. Query expansion and multi-query aggregation rewrite the query into several variants to improve recall, but they operate at the retrieval stage and do not change which passages enter the context, nor in what combination. SES instead uses sub-queries to reweight *selection*, not to expand retrieval.

Diversity-aware selection. Maximal Marginal Relevance (Carbonell & Goldstein, 1998) interpolates relevance with a penalty on the maximum pairwise similarity to the selected set; being purely pairwise, it cannot detect higher-order redundancy (Section 1). Determinantal approaches share our volume-based view of diversity: k-DPPs (Kulesza et al., 2012) sample subsets with probability proportional to the squared volume they span, and SMART-RAG (Li et al., 2024) greedily grows the selected set by increasing its determinantal volume. Because relevance enters through the kernel, quality and diversity are entangled—their balance shifts with the score distribution and is not separately steerable—and sampling-based variants sacrifice determinism. The $(1 - 1/e)$ guarantee applies to greedy MAP inference on $\log\det(L_S)$; k-DPP sampling carries no such guarantee, and SMART-RAG tunes the quality–diversity balance through a weight outside the objective.

Coverage-oriented selection. GeoRAG (Zhang et al., 2026) is the closest prior method: it decomposes the query into sub-queries with a frozen LLM and selects passages by greedy maximization of a demand-weighted facility-location objective. Redundancy enters only through facility location's reassignment mechanism—a passage that duplicates an already-selected one *at the same demand point* earns a discounted gain—but the objective never inspects the geometry of the selected set as a whole, so redundancy among pairwise-dissimilar passages goes undetected (Section 1). Its deployed optimizer also replaces the exact marginal gain with a Sinkhorn-based surrogate that falls outside the objective's $(1 - 1/e)$ analysis.

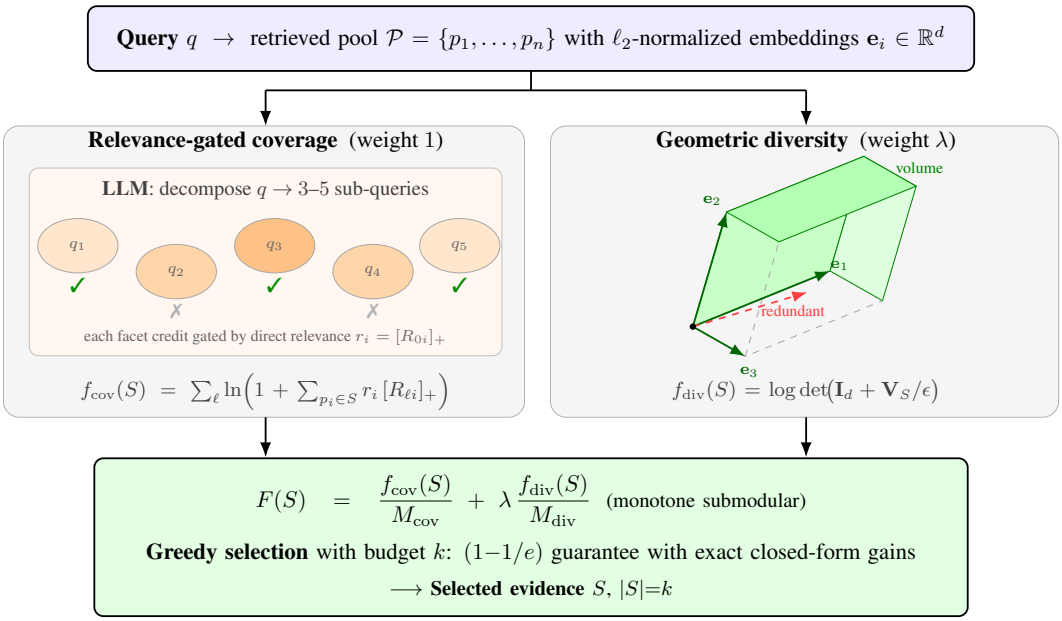


Figure 2: The SES pipeline. Relevance-gated facet coverage (f_{cov}) and geometric diversity (f_{div}) are combined and maximized greedily, inheriting the $(1 - 1/e)$ guarantee.

Submodular selection in NLP. Submodularity has been exploited for extractive summarization (Lin & Bilmes, 2011), training-data pruning (Wei et al., 2015), and conversation subset selection (Gillick et al., 2009), mostly through coverage-style objectives; log-determinant volume criteria have a parallel lineage in sensor placement and experimental design (Krause & Golovin, 2014). Closest in spirit, Qureshi et al. (2026) combine relevance, coverage, and log-determinant diversity for budgeted selection, but outside the fixed-deployment RAG setting: there is no relevance gate deciding *which* passages may carry coverage credit, and the two terms are entangled rather than separately normalized. SES unites the two lines in a single normalized objective whose classical guarantee applies to the deployed greedy algorithm exactly.

Outside the fixed-deployment setting. Trained selectors (Fan et al., 2026; Sharifymoghaddam & Lin, 2026) require task-specific training data or inference budgets incompatible with fixed deployment, and zero-shot multifacet rankers (Wan et al., 2025; LeVine & Varjavand, 2025) score-and-truncate, remaining blind to set composition; we therefore do not compare against them.

3 Method

Figure 2 gives an overview: SES greedily maximizes a normalized submodular objective over the candidate pool, adding only a one-shot, cacheable query-decomposition call per query.

3.1 Problem Statement

Let q be a query and $\mathcal{P} = \{p_1, \ldots, p_n\}$ the candidate pool returned by a first-stage retriever, where n typically ranges from 100 to 1000. Each passage p_i is represented by an ℓ_2-normalized dense embedding $\mathbf{e}_i \in \mathbb{R}^d$ from a bi-encoder; normalization is used throughout and is what makes the first-pick tie and the bounds $\rho(j \mid S) \in [0, 1]$ in Section 3.3 valid. Given a selection budget $k \ll n$, we seek a subset $S \subseteq \mathcal{P}$ with $|S| = k$ that jointly optimizes two criteria. Relevance-gated information coverage requires the selected passages to cover the different semantic facets of q, with each unit of facet coverage contributed by passages that are themselves directly relevant to the original query. Low redundancy requires the selected passages to collectively span a large geometric volume, so that redundancy of any order is penalized.

Query decomposition for coverage. To operationalize coverage, we prompt a frozen instruction-tuned language model to decompose q into 3–5 sub-queries $\mathcal{Q} = \{q_1, \ldots, q_m\}$ ($m \leq 5$), each targeting a distinct semantic dimension of q; the exact prompt is given in Section E. The decomposi-

tion is issued once per query and returns only tens of tokens, so its cost is negligible next to retrieval and generation (Section 4). Together with the original query $q_0 \equiv q$, we obtain the augmented query set $\{q_0, q_1, \ldots, q_m\}$.

Let $\mathbf{g}_\ell \in \mathbb{R}^d$ denote the ℓ_2-normalized embedding of q_ℓ. Since all embeddings are normalized, we define the relevance matrix $\mathbf{R} \in \mathbb{R}^{(m+1)\times n}$ with entries

$$R_{\ell i} = \mathrm{sim}(\mathbf{g}_\ell, \mathbf{e}_i) = \mathbf{g}_\ell^\top \mathbf{e}_i, \tag{1}$$

computed in a single batched matmul. We write $r_i \coloneqq [R_{0i}]_+$ for the direct relevance of p_i, where $[x]_+ \coloneqq \max(x, 0)$. In our experiments the retriever is itself a cosine-scoring bi-encoder, so r_i recovers the first-stage retriever's own score up to clipping; in the reranking experiments of Section 4.2, r_i is instead replaced by the reranker's score, min–max normalized per query to $[0, 1]$ to keep the gate on the same numerical scale as the cosine-based default (with the degenerate guard $r_i = 1$ when all reranker scores are identical, which neutralizes the gate uniformly rather than zeroing out coverage), while $[R_{\ell i}]_+$ and the diversity geometry remain computed with the bi-encoder embeddings.

3.2 Submodular Objective for Evidence Selection

We formulate passage selection as the maximization of a set function $F : 2^{\mathcal{P}} \to \mathbb{R}$ combining coverage and diversity with a trade-off weight $\lambda \geq 0$:

$$F(S) = \frac{f_{\mathrm{cov}}(S)}{M_{\mathrm{cov}}} + \lambda \cdot \frac{f_{\mathrm{div}}(S)}{M_{\mathrm{div}}}, \tag{2}$$

where M_{cov} and M_{div} are normalization constants defined below so that both terms are upper-bounded by 1 and λ is interpretable independently of encoder, pool size, and regularization.

Relevance-gated coverage. We measure the coverage of facet ℓ by the *effective coverage* accumulated by S, where each passage contributes in proportion to both its match against the sub-query and its direct relevance to the original query:

$$\tilde{R}_{\ell i} \coloneqq r_i \cdot [R_{\ell i}]_+. \tag{3}$$

The multiplicative form is the key design choice: $[R_{\ell i}]_+$ ensures the passage addresses facet ℓ, while r_i acts as a *relevance gate* ensuring that the coverage credit is carried by passages that also answer the original query. The clipping at zero is not merely cosmetic: it makes every contribution non-negative, so that the marginal gain of any passage is non-negative—a necessary condition for the monotonicity proved in Theorem 1. Facet coverage is then defined by a saturating log-sum:

$$f_{\mathrm{cov}}(S) = \sum_{\ell=1}^{m} \ln\Big(1 + \sum_{p_i \in S} \tilde{R}_{\ell i}\Big). \tag{4}$$

Equation (4) is a concave, non-decreasing composition of a non-negative modular function, giving the non-negativity and diminishing returns on which the proof of Theorem 1 relies: adding a passage yields a non-negative gain on every facet it serves, but the gain is discounted by the coverage already accumulated, so repeatedly selecting passages that confirm the same facet yields diminishing returns. The logarithm *saturates*, penalizing duplication smoothly while still allowing a passage that partially serves several weakly-covered facets to be preferred over one that perfectly serves a single saturated facet. Note also that no standalone relevance term is needed: through Equation (3), direct relevance shapes *where* each marginal unit of coverage comes from.

Geometric diversity. We measure redundancy as the log-determinant volume spanned by the selected embeddings. Define the scatter matrix

$$\mathbf{V}_S = \sum_{p_i \in S} \mathbf{e}_i \mathbf{e}_i^\top \in \mathbb{R}^{d\times d}, \qquad f_{\mathrm{div}}(S) = \log\det\left(\mathbf{I}_d + \frac{1}{\epsilon}\mathbf{V}_S\right), \tag{5}$$

where $\epsilon > 0$ controls the regularization strength; the identity alone guarantees positive definiteness. Geometrically, $\det(\mathbf{I}_d + \mathbf{V}_S/\epsilon)$ is the squared volume of the regularized parallelotope spanned by the selected vectors. Because the determinant is a *joint* function of all selected vectors, the volume collapses whenever the set degenerates along any collective direction, including higher-order degeneracy among pairwise dissimilar passages that no pairwise criterion can detect.

Global normalization. Both terms are divided by *fixed* upper-bound constants, computed once per query before optimization. For coverage, since every summand in Equation (4) is positive and the logarithm is monotone,

$$M_{\text{cov}} \coloneqq \sum_{\ell=1}^{m} \ln\Big(1 + \sum_{i=1}^{n} \tilde{R}_{\ell i}\Big) = f_{\text{cov}}(\mathcal{P}), \tag{6}$$

the value attained by the full pool, hence a true upper bound on $f_{\text{cov}}(S)$ for any S. In the degenerate case $M_{\text{cov}} = 0$ (every passage scores non-positively on every sub-query), the coverage term is identically zero; we therefore clamp $M_{\text{cov}} \leftarrow \max\{M_{\text{cov}}, 10^{-12}\}$ before optimization, so that the objective remains defined and the diversity term alone drives selection. This clamp is a fixed constant computed before greedy runs, so it does not affect monotonicity, submodularity, or the $(1 - 1/e)$ argument. For diversity, let $\sigma_1^{\downarrow}(\mathbf{G}) \geq \cdots \geq \sigma_n^{\downarrow}(\mathbf{G})$ denote the eigenvalues of the pool Gram matrix $\mathbf{G} = \mathbf{E}\mathbf{E}^{\top}$ in non-increasing order, and set

$$M_{\text{div}} \coloneqq \sum_{i=1}^{\min(k,n)} \ln\Big(1 + \frac{\sigma_i^{\downarrow}(\mathbf{G})}{\epsilon}\Big). \tag{7}$$

To see that this is an upper bound, let $\mathbf{G}_S$ be the principal submatrix of $\mathbf{G}$ indexed by S, with $|S| = k$. By Sylvester's determinant identity,

$$f_{\text{div}}(S) = \log\det\Big(\mathbf{I}_d + \tfrac{1}{\epsilon}\mathbf{E}_S^{\top}\mathbf{E}_S\Big) = \log\det\Big(\mathbf{I}_k + \tfrac{1}{\epsilon}\mathbf{G}_S\Big) = \sum_{i=1}^{k} \ln\Big(1 + \frac{\lambda_i(\mathbf{G}_S)}{\epsilon}\Big), \tag{8}$$

where $\mathbf{E}_S$ stacks the embeddings of S. By Cauchy's interlacing theorem, the eigenvalues of any principal submatrix $\mathbf{G}_S$ with $|S| = k$ are termwise dominated by the top-k eigenvalues of $\mathbf{G}$, and hence $f_{\text{div}}(S) \leq M_{\text{div}}$ for every k-subset S. We thus have $f_{\text{cov}}(S)/M_{\text{cov}} \leq 1$ and $f_{\text{div}}(S)/M_{\text{div}} \leq 1$ for all S, so $F(S) \leq 1 + \lambda$ and λ has the same interpretation for every encoder and pool.

Theorem 1 (Properties of the objective)**.** *The objective $F(S)$ defined in Equation* (2)*–Equation* (7)*, with M_{cov} clamped as described after Equation* (6)*, satisfies, for any $\lambda \geq 0$:*

- ***(i)*** *Empty set: $F(\emptyset) = 0$;*
- ***(ii)*** *Monotonicity: $F(S) \leq F(T)$ for all $S \subseteq T \subseteq \mathcal{P}$;*
- ***(iii)*** *Submodularity: $F(S \cup \{p\}) - F(S) \geq F(T \cup \{p\}) - F(T)$ for all $S \subseteq T \subseteq \mathcal{P}$ and $p \notin T$.*

See Section C for the proof.

The proof combines two standard facts: log-determinant potentials are monotone submodular, and non-negative scaling and addition preserve monotonicity and submodularity; $F(\emptyset) = 0$ is immediate. Theorem 1 matters for a practical reason: unlike per-step re-weighting schemes, the normalization constants are fixed *before* optimization, so the classical $(1 - 1/e)$ constant-factor guarantee for greedy maximization under a cardinality constraint (Nemhauser et al., 1978) applies to F *exactly*. Two structural facts shape the greedy trajectory before any candidate is scored: at $S = \emptyset$ every candidate has the same diversity gain $\log(1 + 1/\epsilon)$ because all embeddings are ℓ_2-normalized, so the first pick is driven entirely by relevance-gated coverage; and because each facet's coverage saturates, later picks are governed by whichever term still carries residual mass.

3.3 Marginal Gains and Efficient Computation

For a candidate $p_j \notin S$, the two marginal gains are:

$$\Delta_{\text{cov}}(j \mid S) = \sum_{\ell=1}^{m} \ln\Big(1 + \frac{\tilde{R}_{\ell j}}{1 + u_\ell(S)}\Big), \qquad u_\ell(S) \coloneqq \sum_{p_i \in S} \tilde{R}_{\ell i}, \tag{9}$$

$$\Delta_{\text{div}}(j \mid S) = \log\left(1 + \frac{\rho(j \mid S)}{\epsilon}\right), \qquad \rho(j \mid S) = \mathbf{e}_j^{\top}\mathbf{A}_S^{-1}\mathbf{e}_j, \tag{10}$$

where $\mathbf{A}_S = \mathbf{I}_d + \mathbf{V}_S/\epsilon$. Since $\tilde{R}_{\ell j} \in [0, 1]$ and $u_\ell(S) \geq 0$, every summand in Equation (9) is non-negative, so $\Delta_{\text{cov}}(j \mid S) \in [0, m]$; the coverage state $\mathbf{u}(S)$ is maintained incrementally in $\mathcal{O}(m)$ per selected passage, so evaluating Equation (9) over all remaining candidates costs $\mathcal{O}(nm)$ per greedy step.

Because the embeddings are ℓ_2-normalized, $\rho(j \mid S) \in [0, 1]$ is evaluated on the small $|S| \times |S|$ Gram matrix rather than the $d \times d$ scatter matrix. Let $\mathbf{E}_S \in \mathbb{R}^{|S| \times d}$ stack the selected embeddings, $\mathbf{G}_S = \mathbf{E}_S \mathbf{E}_S^\top$, $\mathbf{C}_S = \epsilon \mathbf{I}_{|S|} + \mathbf{G}_S$, and $\mathbf{g}_{jS} = \mathbf{E}_S \mathbf{e}_j$ (a subvector of the precomputed pool Gram matrix $\mathbf{G} = \mathbf{E}\mathbf{E}^\top$). The Woodbury identity applied to $\mathbf{A}_S = \mathbf{I}_d + \frac{1}{\epsilon}\mathbf{E}_S^\top \mathbf{E}_S$ gives

$$\mathbf{A}_S^{-1} = \mathbf{I}_d - \mathbf{E}_S^\top \big(\epsilon \mathbf{I}_{|S|} + \mathbf{G}_S\big)^{-1} \mathbf{E}_S \quad \Longrightarrow \quad \rho(j \mid S) = 1 - \mathbf{g}_{jS}^\top \mathbf{C}_S^{-1}\, \mathbf{g}_{jS}, \tag{11}$$

and writing $\mathbf{C}_S = \mathbf{L}_S \mathbf{L}_S^\top$,

$$\Delta_{\text{div}}(j \mid S) = \log\left(1 + \frac{1 - \|\mathbf{L}_S^{-1} \mathbf{g}_{jS}\|_2^2}{\epsilon}\right). \tag{12}$$

The quantity $s(j \mid S) \coloneqq \epsilon + \rho(j \mid S) = \epsilon \exp(\Delta_{\text{div}}(j \mid S))$ is the Schur complement of $\mathbf{C}_{S \cup \{j\}}$, positive by construction; since $\log s(j \mid S) = \Delta_{\text{div}}(j \mid S) + \log \epsilon$, computing $\log s(j \mid S)$ instead of Equation (12) shifts all diversity gains by the candidate-independent constant $\log \epsilon$ and leaves the greedy choice unchanged. Two boundary effects shape the later greedy steps and are analyzed precisely in Section D.

3.4 Greedy Optimization

Since F is monotone submodular, standard greedy selection—iteratively picking the element with the largest joint marginal gain $\Delta_F(j \mid S)$—provides a $(1 - 1/e)$ constant-factor approximation under the cardinality constraint (Nemhauser et al., 1978). The joint gain is the weighted sum $\Delta_F(j \mid S) = \Delta_{\text{cov}}(j \mid S)/M_{\text{cov}} + \lambda\, \Delta_{\text{div}}(j \mid S)/M_{\text{div}}$; the selection state updates incrementally: the coverage accumulator $\mathbf{u}$ in $\mathcal{O}(m)$ and the Cholesky factor of $\mathbf{C}_S$ in $\mathcal{O}(|S|^2)$ per step (Equation (15)). The complete procedure is given in Algorithm 1. Summing the per-step costs over k steps, the greedy loop runs in

$$\mathcal{O}\big(nkm + nk^3\big), \tag{13}$$

linear in the pool size n for fixed k and $m \in [3, 5]$, plus a one-time $\mathcal{O}(n^2 d)$ cost to precompute the pool Gram matrix. At $k \leq 10$ and n up to 1600, the computation reduces to a handful of small matrix operations that complete in milliseconds (Section 4).

4 Experiments

Datasets. We evaluate on four multi-hop QA benchmarks: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (2Wiki) (Ho et al., 2020), StrategyQA (Geva et al., 2021), and Bamboogle (Press et al., 2023). We report exact match (EM, %) on 1,000-instance samples per dataset (Bamboogle: the full 125-question test set), computed after standard normalization against the gold answer string; sampling details appear in Section G.

Pipeline. All methods share one fixed pipeline: E5-base-v2 retrieves the top n=100 passages (Wang et al., 2022; Jin et al., 2024), and one of three frozen instruction-tuned generators (Llama-3.1-8B-Instruct (Dubey et al., 2024), Qwen2.5-7B-Instruct, Qwen2.5-14B-Instruct (Qwen et al., 2025); henceforth Llama-8B, Qwen-7B, and Qwen-14B) answers from the k=4 selected passages at temperature 0 with at most 256 output tokens. We set λ=0.1 and ϵ=10^{-6} throughout, with the fixed normalizers of Equation (6)–Equation (7). Sub-queries are decomposed by Qwen2.5-14B-Instruct in all configurations, so that selection quality is held constant across generator ablations; the answer generator never observes the sub-queries.

Baselines. Top-k selection by retriever score; MMR (Carbonell & Goldstein, 1998) (trade-off 0.7); greedy k-DPP (MAP) with a quality-weighted kernel (Kulesza et al., 2012); SMART-RAG's determinantal selection (Li et al., 2024); and GeoRAG's submodular coverage with LLM-generated sub-queries (Zhang et al., 2026). All methods, including SES, are deterministic. Per-baseline hyperparameters, prompts, and hardware appear in Sections E to G.

4.1 Main Results

Table 1 shows three consistent effects. **First**, SES attains the best average EM under all three generators, improving over top-k by $+1.9$ (Llama-8B), $+2.6$ (Qwen-7B), and $+2.5$ (Qwen-14B)

Table 1: Exact match (EM, %) on four multi-hop QA benchmarks with three frozen generators, $k = 4$. Best result per generator block and per dataset is in **bold**.

Generator	Method	2Wiki	Bamb.	HotpotQA	StratQA	Avg.
Llama-3.1-8B	Top-k	23.4	12.8	31.0	62.5	32.4
	MMR	24.4	13.6	29.5	63.9	32.9
	DPP	23.6	11.2	31.5	**64.4**	32.7
	SMART-RAG	25.1	13.6	30.3	63.4	33.1
	GeoRAG	23.4	14.4	30.5	64.1	33.1
	SES	**25.5**	**16.0**	**32.1**	63.4	**34.3**
Qwen2.5-7B	Top-k	24.9	16.0	30.7	65.4	34.3
	MMR	26.1	16.8	31.4	65.0	34.8
	DPP	26.3	17.6	30.4	65.7	35.0
	SMART-RAG	25.7	20.0	30.2	65.5	35.4
	GeoRAG	26.2	18.4	30.9	65.1	35.2
	SES	**27.0**	**21.6**	**32.7**	**66.2**	**36.9**
Qwen2.5-14B	Top-k	26.6	18.4	32.3	63.9	35.3
	MMR	27.2	18.4	32.9	64.1	35.7
	DPP	28.1	17.6	33.4	65.0	36.0
	SMART-RAG	28.4	20.8	33.1	64.2	36.6
	GeoRAG	27.4	21.6	34.1	64.4	36.9
	SES	**28.8**	**22.4**	**34.5**	**65.5**	**37.8**

points and over the strongest baseline in each block by $+1.2$, $+1.5$, and $+0.9$ points. **Second**, the gains are largest where the diagnosed failure modes are most acute: on Bamboogle, whose hand-crafted questions defeat retrieval shortcuts, SES gains between $+3.2$ and $+5.6$ points over top-k, and on 2Wiki, whose comparison and bridge questions demand complementary facets, SES improves over every diversity-aware and coverage-aware baseline. **Third**, the method ordering is largely preserved from the 7B to the 14B generator—SES attains the best average in every block, and only the relative order of the two mid-tier baselines (SMART-RAG and GeoRAG) swaps—indicating that the bottleneck addressed by selection remains binding as generators scale.

The single exception is StrategyQA under Llama-8B, where k-DPP baseline edges out SES. We attribute this to the generator rather than the selection rule: StrategyQA's implicit commonsense questions stress the model's *ability to exploit* evidence more than the composition of that evidence. Under the two Qwen generators SES leads on StrategyQA as well, confirming that the cell reflects generator-side evidence utilization rather than a deficiency of selection.

4.2 Composition with Cross-Encoder Reranking

Rerankers improve passage quality but still select the top-k by score, leaving composition uncontrolled. Table 2 shows that SES stacks cleanly on both rerankers: it improves MiniLM-CE by $+0.7$ and BGE-CE by $+0.6$ average EM, reaching the best overall configuration. The single flat cell is consistent with our account of the method: a reranker that already places complementary facets at the top of the pool leaves less compositional headroom for selection.

Table 2: Additivity of SES to cross-encoder reranking (Qwen2.5-14B, EM %). SES selects from the reranked top-100 pool, using the reranker score as the relevance gate r_i. **Bold**: adding SES improves over the corresponding reranker-only selection.

Selection	2Wiki	Bamb.	HotpotQA	StratQA	Avg.
E5 + SES	28.8	22.4	34.5	65.5	37.8
MiniLM-CE	29.2	19.2	36.6	64.6	37.4
MiniLM-CE + SES	**29.6**	**20.0**	**37.4**	**65.2**	**38.1**
BGE-CE	32.2	20.8	40.9	62.1	39.0
BGE-CE + SES	32.2	**22.4**	**41.4**	**62.3**	**39.6**

4.3 Encoder Ablation

Table 3 replaces the bi-encoder for everything it feeds *except* the direct relevance gate: the substituted encoder supplies both the embeddings for the log-determinant volume and the passage–sub-

Table 3: Encoder ablation (Qwen2.5-14B, direct relevance fixed to the first-stage retriever score, EM %). Each row replaces the bi-encoder for both the log-determinant volume and the passage–sub-query matching $[R_{\ell i}]_+$; only r_i is held at the original retriever score.

Encoder	2Wiki	Bamb.	HotpotQA	StratQA	Avg.
E5-base-v2	28.8	22.4	34.5	65.5	37.8
all-MiniLM-L6-v2	26.2	16.8	28.3	63.6	33.7
bge-base-en-v1.5	29.6	21.6	34.5	**65.7**	37.9
Qwen3-8B-embed.	**30.1**	**26.4**	**36.0**	65.4	**39.5**

query matching scores $[R_{\ell i}]_+$, while the direct relevance r_i remains the original first-stage retriever score in all rows. Across these four encoders, performance tracks embedding quality rather than encoder family: bge-base-en-v1.5 matches E5, the weaker all-MiniLM-L6-v2 degrades the average to 33.7, and Qwen3-8B-embedding, trained with stronger contrastive objectives, lifts it to 39.5. SES thus inherits the geometry quality of whatever bi-encoder it is paired with, rather than requiring a specific one.

4.4 Sensitivity to the Trade-off Weight

Figure 3 sweeps the single global weight λ on the Qwen2.5-14B generator. Two conclusions follow. **First**, both terms contribute: removing diversity ($\lambda = 0$) costs 0.4 average EM relative to the default $\lambda = 0.1$, confirming that the geometric term captures redundancy the coverage term cannot see. **Second**, the per-dataset optima differ markedly—2Wiki peaks at $\lambda = 0.2$, HotpotQA prefers $\lambda = 0$, and Bamboogle and StrategyQA prefer $\lambda = 0.1$—reflecting how differently the two diagnosed failure modes weigh on each dataset. The cross-dataset average, however, forms a plateau rather than a peak: any $\lambda \in [0, 0.2]$ attains an average within 0.7 EM of the best configuration. The value of composition is likewise budget-dependent: SES improves over top-k at every budget tested, with the gain peaking at k=4 (+2.5 EM) and shrinking as the budget loosens (Section H)

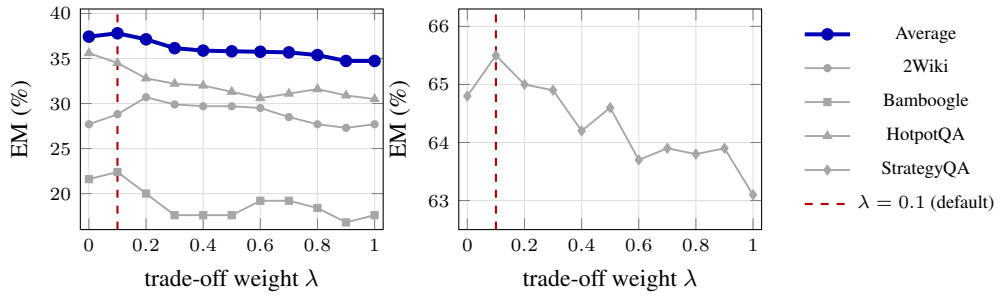


Figure 3: Sensitivity to the trade-off weight λ on Qwen2.5-14B (full results in Table 5). StrategyQA (right) is plotted on a separate scale.

5 Conclusion

We revisited a choice that RAG pipelines usually leave to default: which of the retrieved passages enter the context window, and in what composition. Relevance-based selection fails in two ways that enlarging the budget does not by itself repair—facet starvation and higher-order redundancy—and that pairwise diversity penalties cannot cure. Grouping the three selection factors by relation type, we fused the two query-side factors into a single relevance-gated coverage term and opposed it with a log-determinant geometric diversity term, yielding one monotone submodular objective whose normalizers are fixed before optimization. The classical $(1-1/e)$ guarantee of greedy maximization therefore holds for the deployed algorithm unchanged. Across four multi-hop QA benchmarks and three frozen generators, SES consistently improves average exact match, composes cleanly with cross-encoder reranking, and adds only millisecond-scale CPU selection to a fixed pipeline—evidence that the error floor of RAG can be lowered by *choosing* evidence better, not only by retrieving more.

AI USE STATEMENT

In this work, we used generative AI tools (Kimi) to aid and polish the writing of the manuscript, to verify mathematical derivations and proofs, and to assist with code implementation. We have not used generative AI tools for conceiving the research idea, designing the selection objective or experiments, or interpreting experimental results. All AI-assisted text, derivations, and code were reviewed and verified by the authors. We take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only publicly available datasets (HotpotQA, 2Wiki- MultiHopQA, StrategyQA, Bamboogle), publicly released retrievers, rerankers, and language models. No human subjects are involved, no personal or sensitive data is collected, and no potentially harmful insights or applications are identified.

REPRODUCIBILITY STATEMENT

The selection procedure is fully specified by Algorithm 1 (Section B), with the objective's properties proved in Section C. Scale alignment, saturation behavior, and the exactness of the $(1 - 1/e)$ guarantee are analyzed in Section D. All implementation details needed to reproduce our results—query decomposition and answer generation prompts (Sections E and F), hardware, sampling, hyperparameters, encoder and reranker configurations, and baseline settings (Section G)—are provided in the appendices, together with the full trade-off sweep (Table 5) and selection latency measurements (Section J).

REFERENCES

Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 335–336, 1998.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Yongqi Fan, Yuxiang Chu, Zhentao Xia, Xiaoyang Chen, Jie Liu, Haijin Liang, Jin Ma, Ben He, Yingfei Sun, Dezhi Ye, and Tong Ruan. Rank4gen: Rag-preference-aligned document set selection and ranking, 2026. URL https://arxiv.org/abs/2601.11273.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021.

Dan Gillick, Korbinian Riedhammer, Benoit Favre, and Dilek Hakkani-Tur. A global optimization framework for meeting summarization. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4769–4772. IEEE, 2009.

Jun Guo, Bojian Chen, Zhichao Zhao, Jindong He, Shichun Chen, Donglan Hu, and Hao Pan. Bkrag: A bge reranker rag for similarity analysis of power project requirements. In *Proceedings of the 2024 6th International Conference on Pattern Recognition and Intelligent Systems*, pp. 14–20, 2024.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Jiajie Jin, Yutao Zhu, Xinyu Yang, Chenghao Zhang, and Zhicheng Dou. Flashrag: A modular toolkit for efficient retrieval-augmented generation research. *CoRR*, abs/2405.13576, 2024. URL https://arxiv.org/abs/2405.13576.

Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 3(71-104): 3, 2014.

Alex Kulesza, Ben Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.

Will LeVine and Bijan Varjavand. Relevance isn't all you need: Scaling rag systems with inference-time compute via multi-criteria reranking. *arXiv preprint arXiv:2504.07104*, 2025.

Jiatao Li, Xinyu Hu, and Xiaojun Wan. Smart-rag: Selection using determinantal matrices for augmented retrieval, 2024. URL https://arxiv.org/abs/2409.13992.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pp. 510–520, 2011.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models, 2023. URL https://arxiv.org/abs/2210.03350.

Khizar Qureshi, Geoffrey Martin, and Yifan Peng. Budget-aware routing for long clinical text. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 42583–42598, 2026.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Sahel Sharifymoghaddam and Jimmy Lin. Rerank before you reason: Analyzing reranking tradeoffs through effective token cost in deep search agents, 2026. URL https://arxiv.org/abs/2601.14224.

Zhengliang Shi, Lingyong Yan, Weiwei Sun, Yue Feng, Pengjie Ren, Xinyu Ma, Shuaiqiang Wang, Dawei Yin, Maarten de Rijke, and Zhaochun Ren. Direct retrieval-augmented optimization: Synergizing knowledge selection and language models, 2025. URL https://arxiv.org/abs/2505.03075.

Fangzheng Tian, Debasis Ganguly, and Craig Macdonald. Is relevance propagated from retriever to generator in rag? In *European Conference on Information Retrieval*, pp. 32–48. Springer, 2025.

Bingyu Wan, Fuxi Zhang, Zhongpeng Qi, Jiayi Ding, Jijun Li, Baoshi Fan, Yijia Zhang, and Jun Zhang. Cognitive-aligned document selection for retrieval-augmented generation, 2025. URL https://arxiv.org/abs/2502.11770.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022. URL https://arxiv.org/abs/2212.03533.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International conference on machine learning*, pp. 1954–1963. PMLR, 2015.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 641–649, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. URL https://arxiv.org/abs/1809.09600.

Bingxue Zhang, Jianying Jia, and Feida Zhu. Covering the unseen: Information demand coverage optimization for retrieval-augmented generation, 2026. URL https://arxiv.org/abs/2606.29328.

A Greedy Optimization with Incremental Cholesky Updates

The body of the paper gives closed-form expressions for both marginal gains; this appendix derives the joint gain, the Cholesky update rule, and the full complexity accounting. At each greedy step the joint gain of a candidate $p_j \notin S$ is

$$\Delta_F(j \mid S) = \frac{\Delta_{\text{cov}}(j \mid S)}{M_{\text{cov}}} + \lambda \cdot \frac{\Delta_{\text{div}}(j \mid S)}{M_{\text{div}}}, \tag{14}$$

with Δ_{cov} and Δ_{div} computed as in Equation (9)–Equation (12). After selecting p_{j^*}, the coverage state updates as $u_\ell \leftarrow u_\ell + \tilde{R}_{\ell j^*}$ in $\mathcal{O}(m)$, and the Cholesky factor of $\mathbf{C}_{S'} = \epsilon\mathbf{I} + \mathbf{G}_{S'}$ updates in $\mathcal{O}(|S|^2)$:

$$\mathbf{L}_{S'} = \begin{bmatrix} \mathbf{L}_S & \mathbf{0}_{|S|} \\ \boldsymbol{\ell}^\top & \sqrt{s} \end{bmatrix}, \qquad \boldsymbol{\ell} = \mathbf{L}_S^{-1}\mathbf{g}_{j^*S}, \qquad s = \epsilon + 1 - \|\boldsymbol{\ell}\|_2^2, \tag{15}$$

where the triangular solve for $\boldsymbol{\ell}$ is reused from the evaluation of Equation (12) in the same greedy step. Since $\mathbf{C}_{S'} \succ 0$, $s > 0$ in exact arithmetic; the diagonal is clamped at $s \geq \epsilon$ solely as a round-off guard (see Section B for the guard semantics).

Complexity derivation. The relevance matrix $\mathbf{R}$ and the pool Gram matrix $\mathbf{G}$ are precomputed once per query in $\mathcal{O}\big((m{+}1)nd + n^2d\big)$; the normalizer M_{cov} costs $\mathcal{O}(mn)$, and M_{div} requires the top-$\min(k, n)$ eigenvalues of $\mathbf{G}$, obtainable in $\mathcal{O}(n^2k)$ via an iterative eigensolver (Section G). Each greedy step evaluates Δ_{cov} over all remaining candidates in $\mathcal{O}(nm)$ and Δ_{div} in $\mathcal{O}(n|S|^2)$ via gathers and triangular back-substitutions; the Cholesky update costs $\mathcal{O}(|S|^2)$, giving Equation (13).

B Greedy Selection Procedure

Algorithm 1 SES: Greedy Submodular Evidence Selection

Require: pool $\mathcal{P}$ with ℓ_2-normalized embeddings $\{\mathbf{e}_i\}_{i=1}^n$; query q; budget k; weight λ; sub-queries $\{q_\ell\}_{\ell=1}^m$; regularization ϵ

1: Compute $\mathbf{R}$ (Equation (1)) and pool Gram matrix $\mathbf{G} = \mathbf{E}\mathbf{E}^\top$; set $\tilde{R}_{\ell i} = r_i[R_{\ell i}]_+$ and compute M_{cov}, M_{div} (Equation (6), Equation (7)); $M_{\text{cov}} \leftarrow \max\{M_{\text{cov}}, 10^{-12}\}$ (degenerate-pool guard; fixed before the loop)
2: $S \leftarrow \emptyset$; $\mathbf{u} \leftarrow \mathbf{0}_m$; $\mathbf{L} \leftarrow$ empty
3: **for** $t = 1$ **to** k **do**
4: **for** each $p_j \notin S$ **do**
5: $\Delta_{\text{cov}}(j \mid S)$ via Equation (9); $\boldsymbol{\ell}_j \leftarrow \mathbf{L}^{-1}\mathbf{g}_{jS}$ (empty if $S = \emptyset$)
6: $s(j \mid S) \leftarrow \epsilon + 1 - \|\boldsymbol{\ell}_j\|_2^2$; score candidates by $\Delta_F(j \mid S)$ using $\Delta_{\text{div}}(j \mid S) = \log s(j \mid S) - \log \epsilon$ (constant shift; ranking-equivalent to Equation (12)) via Equation (14)
7: **end for**
8: $j^* \leftarrow \arg\max_{p_j \notin S} \Delta_F(j \mid S)$; $S \leftarrow S \cup \{p_{j^*}\}$
9: $\mathbf{u} \leftarrow \mathbf{u} + \tilde{R}_{:j^*}$; extend $\mathbf{L}$ by Equation (15) with $\boldsymbol{\ell}_{j^*}$ and $s = \max\{\epsilon + 1 - \|\boldsymbol{\ell}_{j^*}\|_2^2,\ \epsilon\}$, where the clamp is a round-off guard only: if s is not within round-off tolerance of $[\epsilon, \epsilon + 1]$, the implementation should recompute s in higher precision rather than silently altering the objective
10: **end for**
11: **return** S

C Proof of Theorem 1

We prove the three properties for each term of

$$F(S) = \frac{f_{\text{cov}}(S)}{M_{\text{cov}}} + \lambda \cdot \frac{f_{\text{div}}(S)}{M_{\text{div}}},$$

where $f_{\text{cov}}(S) = \sum_{\ell=1}^{m} \ln\big(1 + \sum_{p_i \in S} \tilde{R}_{\ell i}\big)$ with $\tilde{R}_{\ell i} = r_i[R_{\ell i}]_+ \geq 0$, and $f_{\text{div}}(S) = \log\det\big(\mathbf{I}_d + \mathbf{V}_S/\epsilon\big)$. Since $\tilde{R}_{\ell i} \geq 0$ by the clipping in Equation (3), every modular sum $u_\ell(S) = \sum_{p_i \in S} \tilde{R}_{\ell i}$ is non-negative, and M_{cov} is clamped to a positive constant as described after Equation (6) before optimization.

(i) Empty set. $f_{\text{cov}}(\emptyset) = \sum_{\ell=1}^{m} \ln(1+0) = 0$ and $f_{\text{div}}(\emptyset) = \log\det(\mathbf{I}_d) = 0$, hence $F(\emptyset) = 0$.

(ii) Monotonicity. For f_{cov}, if $S \subseteq T$ then $u_\ell(S) \leq u_\ell(T)$ for every ℓ, and since $\ln(1+x)$ is non-decreasing on $x \geq 0$, $f_{\text{cov}}(S) \leq f_{\text{cov}}(T)$. For f_{div}, if $S \subseteq T$ then $\mathbf{V}_T = \mathbf{V}_S + \sum_{p_i \in T \setminus S} \mathbf{e}_i \mathbf{e}_i^\top \succeq \mathbf{V}_S$ in Loewner order, and the map $\mathbf{X} \mapsto \log\det(\mathbf{I}_d + \mathbf{X}/\epsilon)$ is monotone on the positive semidefinite cone, so $f_{\text{div}}(S) \leq f_{\text{div}}(T)$. Finally, $M_{\text{cov}} > 0$ (after the fixed clamp) and $M_{\text{div}} > 0$ are positive constants and $\lambda \geq 0$, so monotonicity is preserved under the conic combination.

(iii) Submodularity. For f_{cov}, fix $S \subseteq T$ and $p_j \notin T$. From Equation (9),

$$\Delta_{\text{cov}}(j \mid S) = \sum_{\ell=1}^{m} \ln\Big(1 + \frac{\tilde{R}_{\ell j}}{1 + u_\ell(S)}\Big) \;\geq\; \sum_{\ell=1}^{m} \ln\Big(1 + \frac{\tilde{R}_{\ell j}}{1 + u_\ell(T)}\Big) = \Delta_{\text{cov}}(j \mid T),$$

where the inequality holds because $u_\ell(S) \leq u_\ell(T)$, $\tilde{R}_{\ell j} \geq 0$, and $x \mapsto \ln(1 + c/(1+x))$ is non-increasing in $x \geq 0$ for any $c \geq 0$; this is exactly the diminishing-returns property. For f_{div}, write $\mathbf{M}(S) = \mathbf{I}_d + \mathbf{V}_S/\epsilon \succ 0$. By the matrix determinant lemma,

$$\Delta_{\text{div}}(j \mid S) = \log\det\big(\mathbf{M}(S) + \mathbf{e}_j \mathbf{e}_j^\top/\epsilon\big) - \log\det \mathbf{M}(S) = \log\Big(1 + \tfrac{1}{\epsilon}\, \mathbf{e}_j^\top \mathbf{M}(S)^{-1} \mathbf{e}_j\Big).$$

Since $\mathbf{V}_T \succeq \mathbf{V}_S$, we have $\mathbf{M}(T) \succeq \mathbf{M}(S) \succ 0$, and operator antitonicity of inversion on the positive definite cone gives $0 \prec \mathbf{M}(T)^{-1} \preceq \mathbf{M}(S)^{-1}$, hence $\mathbf{e}_j^\top \mathbf{M}(T)^{-1} \mathbf{e}_j \leq \mathbf{e}_j^\top \mathbf{M}(S)^{-1} \mathbf{e}_j$. Because $\log(1 + x/\epsilon)$ is non-decreasing in $x \geq 0$, $\Delta_{\text{div}}(j \mid S) \geq \Delta_{\text{div}}(j \mid T)$. Submodularity is preserved under non-negative conic combination, proving the claim for F. □

D Scale Alignment and Saturation

The two terms live on heterogeneous scales before normalization: f_{cov} depends on the number of facets and grows at most logarithmically with pool size (since $\tilde{R}_{\ell i} \leq 1$ implies $f_{\text{cov}}(\mathcal{P}) \leq m \ln(1+n)$), whereas $f_{\text{div}} \in [0, k\log(1+1/\epsilon)]$ is bounded (the sum $\sum_{i=1}^{k} \ln(1 + \lambda_i(\mathbf{G}_S)/\epsilon)$ is maximized, by concavity, when the k eigenvalues are equal) and depends on the regularizer ϵ; note that any single-step coverage marginal is also bounded, $\Delta_{\text{cov}}(j|S) \leq m$, because each summand satisfies $\ln(1 + a/(1+u)) \leq a$ for $a \in [0,1]$. We therefore normalize *once*, before optimization, by the fixed upper bounds $M_{\text{cov}} = f_{\text{cov}}(\mathcal{P})$ (clamped as in Equation (6)) and M_{div} of Equation (6)–Equation (7), so that both terms are upper-bounded by 1 and the trade-off weight λ is interpretable independently of encoder, pool, and ϵ. Two boundary cases are handled exactly. *First selection:* at $S = \emptyset$ every candidate has diversity gain $\log(1 + 1/\epsilon)$, since all embeddings are ℓ_2-normalized, so the first pick is driven by the coverage term. *Saturation:* the coverage gain of Equation (9) decays as $\mathcal{O}(1/u_\ell)$ once a facet is well covered, and once the selected set approaches the effective rank of the candidate embeddings, $\rho(j \mid S) \approx 0$ for all remaining candidates and diversity gains vanish; selection then gracefully reduces to accumulating residual coverage on the least-served facets. Because the normalizers are fixed constants, the $(1 - 1/e)$ guarantee of Theorem 1 applies unchanged to the deployed algorithm, which is exact greedy on F with closed-form marginal gains; the clamping $s \geq \epsilon$ in Equation (15) is a round-off guard that does not alter the objective. *Relevance-agnostic volume.* The volume gain is blind to direct relevance by design: in principle a large λ could admit a passage that is geometrically novel yet off-topic. Two safeguards keep this behavior in check in practice: the coverage-driven first pick (all candidates tie on diversity at $S = \emptyset$), and the small default $\lambda = 0.1$, far below the regime in which diversity dominates (Figure 3).

E Query Decomposition Prompt

We decompose each query with the prompt below (system / user messages), using temperature 0. The model is asked to carve the query into 3–5 sub-queries along distinct semantic dimensions and to output only a JSON array.

```
System: You are a query decomposition assistant.
Decompose the input query into 3-5 sub-queries that together
cover its different semantic dimensions, such as:
entities/definitions, attributes/properties,
causes/mechanisms, comparisons, temporal or contextual constraints.
Rules:
1. Each sub-query must be self-contained and answerable independently.
2. Do NOT answer the queries, only decompose.
3. Keep the original language of the input query.
4. Output ONLY a JSON array of strings, e.g. ["q1", "q2", "q3"].
No explanations, no markdown fences.
User: Query: {query}
Sub-queries:
```

F Answer Generation Prompts

All methods share the same answer-generation configuration: temperature 0, maximum 256 output tokens. For HotpotQA, 2Wiki, and Bamboogle we use the prompt below; StrategyQA uses a variant that constrains the answer to Yes/No.

HotpotQA, 2Wiki, and Bamboogle.

```
System: Answer STRICTLY based on the given documents only.
Do NOT use external knowledge.
Provide ONLY the final answer in the exact format.
NO explanations, NO introductions, NO analysis.
Format: 'The answer is: <answer>.'
The following are given documents.
{reference}
User: Question: {question}
```

StrategyQA.

```
System: Answer STRICTLY based on the given documents only.
Do NOT use external knowledge or make assumptions beyond the
provided documents.
The question requires a Yes or No answer. Choose the best answer
based ONLY on the documents.
Provide ONLY the final answer in the exact format.
NO explanations, NO introductions, NO analysis.
Format: 'The answer is: <Yes or No>.'
The following are given documents.
{reference}
User: Question: {question}
```

G Reproducibility Details

- **Hardware.** All experiments run on a Linux server with an AMD EPYC 7742 CPU, 512 GB RAM, and an NVIDIA A800-SXM4-80GB GPU; SES selection latency is measured on a single CPU thread with pre-computed embeddings.
- **Sampling.** We sample 1,000 instances with a fixed random seed (42) from the development splits of HotpotQA and 2Wiki; for StrategyQA we follow prior retrieval-augmented evaluations and sample 1,000 instances from the public labeled split with the same seed; Bamboogle is evaluated in full on its test set (125 questions).
- **Regularization.** $\epsilon = 10^{-6}$ throughout; the Cholesky diagonal is clamped at $s \geq \epsilon$ (Equation (15)) as a round-off guard only.

- **Normalizer computation.** M_{div} uses the top $\min(k, n)$ eigenvalues of the pool Gram matrix $\mathbf{G}$, computed with an iterative eigensolver (Lanczos/LOBPCG; each iteration costs an $\mathcal{O}(n^2)$ matrix–vector product with $\mathbf{G}$), with a full eigendecomposition $\mathcal{O}(n^3)$ as a robust fallback. Storing $\mathbf{R}$ and $\mathbf{G}$ requires $(m{+}1)n + n^2$ floats ($\approx$4 MB at $n = 1000$), cache-resident in practice.
- **Embeddings.** All passage and query embeddings are ℓ_2-normalized before the Gram matrix is formed; queries and passages are prefixed with `query:` and `passage:` before encoding. Direct relevance is $r_i = [R_{0i}]_+$ as defined after Equation (1). In the default E5 configuration no per-query re-scaling of relevance or diversity gains is applied, since the fixed normalizers M_{cov} and M_{div} already align the scales (Section D); the reranking combinations are the explicit exception, in which the gate r_i is the min–max normalized reranker score described in Section 4.2. The full pool Gram matrix $\mathbf{G} = \mathbf{E}\mathbf{E}^\top$ is precomputed once per query pool. The default encoder is E5-base-v2; the encoder ablation (Table 3) replaces the bi-encoder for both the log-determinant volume and the passage–sub-query matching $[R_{\ell i}]_+$, holding the E5 relevance signal r_i fixed.
- **Query decomposition.** We use the prompt of Section E with temperature 0, issued once per query and cached. Decomposition is performed by Qwen2.5-14B-Instruct in all configurations, independent of the answer generator, so that selection quality is constant across generator ablations.
- **Hyperparameters.** Pool size $n = 100$, selection budget $k = 4$, and default $\lambda = 0.1$ for all datasets and generators; no per-dataset tuning is performed for any selection method (Section 4.4).
- **MMR baseline.** The MMR trade-off parameter is fixed at 0.7 for all datasets, i.e., the relevance weight in its greedy update rule is 0.7 and the redundancy-penalty weight is 0.3.
- **DPP baseline.** The kernel is constructed as $L_{ij} = q_i \cdot s(\mathbf{e}_i, \mathbf{e}_j) \cdot q_j$, where $q_i = r_i = [R_{0i}]_+$ is the clipped first-stage retriever score and $s(\mathbf{e}_i, \mathbf{e}_j)$ is the cosine similarity between ℓ_2-normalized E5 embeddings; the cosine Gram matrix is positive semidefinite and the rank-one factor $\mathbf{q}\mathbf{q}^\top$ is positive semidefinite, so their Hadamard product L is a valid DPP kernel. Subsets of size k are selected by greedy MAP inference Kulesza et al. (2012).
- **Cross-encoder reranking.** Reranker baselines use ms-marco-MiniLM-L-6-v2 (MiniLM-CE) and BGE-Reranker-Large (BGE-CE) to score the top-100 pool; scores are min-max normalized per query (with the degenerate guard of Section 3.1) and the top-k passages are selected, with no task-specific adaptation. In the combined configurations, the normalized reranked scores serve as the direct relevance r_i; sub-query matching $[R_{\ell i}]_+$ remains computed with E5 embeddings, so only the relevance gate is replaced. All configurations share the same generation setup.
- **Generation.** Temperature 0, maximum 256 output tokens, identical prompts across methods.

H Sensitivity to the Selection Budget

Table 4: Sensitivity to the selection budget k (Qwen2.5-14B, EM %). **Bold**: better result within each budget block.

k	Method	2Wiki	Bamb.	HotpotQA	StratQA	Avg.
3	Top-k	24.9	**19.2**	30.8	62.9	34.5
	SES	**28.1**	18.4	**33.9**	**64.8**	**36.3**
4	Top-k	26.6	18.4	32.3	63.9	35.3
	SES	**28.8**	**22.4**	**34.5**	**65.5**	**37.8**
5	Top-k	27.5	20.0	34.9	63.6	36.5
	SES	**29.7**	20.0	**35.0**	**64.7**	**37.4**
7	Top-k	29.6	**21.6**	35.5	65.3	38.0
	SES	**31.1**	20.0	**35.6**	**66.7**	**38.4**

Three observations follow. **First**, SES improves over top-k on the cross-dataset average at every budget tested, but the gain is non-monotone in k: about $+1.8$ EM at k=3, $+2.5$ at k=4, $+0.9$ at k=5, and $+0.4$ at k=7 (averages rounded to one decimal). The peak at k=4 rather than at the tightest

budget indicates that the diagnosis of Section 1 is about composition, not budget size per se: at very small budgets there are few slots to trade off, and the relevance gate plus coverage term already dominate. **Second**, the gap shrinks for $k \geq 4$, consistent with the idea that a looser budget eventually admits the scattered complements by itself; selection matters most where context is scarce—the regime in which context-window cost dominates. **Third**, SES trails top-k on Bamboogle at k=3 (18.4 vs. 19.2) and k=7 (20.0 vs. 21.6) and ties at k=5 (20.0 vs. 20.0), while winning the other three datasets at each of these budgets; this exposes the limitation of a single fixed trade-off. At k=7, top-k's additional slots now include the needed complements, whereas the diversity term at $\lambda = 0.1$ can still substitute geometrically novel but answer-peripheral passages; at k=3 the same fixed weight leaves too few slots for a peripheral volume pick to be absorbed. Per-budget or per-dataset tuning of λ would likely remove these cells, but would weaken the single-global-weight story, so we keep $\lambda = 0.1$ fixed throughout.

I Full Trade-off Weight Sweep

Table 5: Full trade-off weight sweep (Qwen2.5-14B, EM %). $\lambda = 0$ recovers pure relevance-gated coverage without the diversity term; see Figure 3 for the corresponding curves; **bold** marks the best cross-dataset average.

λ	2Wiki	Bamboogle	HotpotQA	StrategyQA	Avg.
0.0	27.7	21.6	35.6	64.8	37.4
0.1	28.8	22.4	34.5	65.5	**37.8**
0.2	**30.7**	20.0	32.8	65.0	37.1
0.3	29.9	17.6	32.2	64.9	36.2
0.4	29.7	17.6	32.0	64.2	35.9
0.5	29.7	17.6	31.3	64.6	35.8
0.6	29.5	19.2	30.6	63.7	35.8
0.7	28.5	19.2	31.1	63.9	35.7
0.8	27.7	18.4	31.6	63.8	35.4
0.9	27.3	16.8	30.9	63.9	34.7
1.0	27.7	17.6	30.5	63.1	34.7

J Efficiency and Scaling

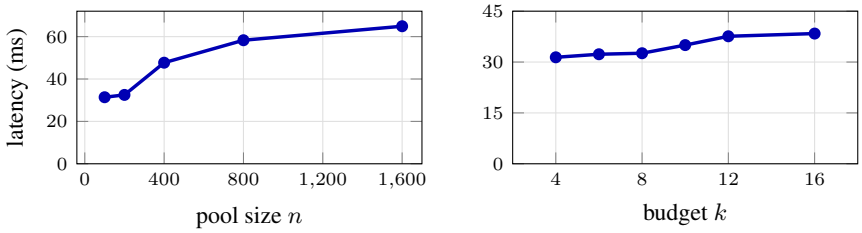


Figure 4: Selection latency of SES (single CPU thread, mean per query, including per-query Gram-matrix and normalizer computation; E5-base-v2, $d = 768$). *Left*: pool size n at $k = 4$—a 16× larger pool raises latency only 2.1×. *Right*: budget k at $n = 100$—quadrupling the budget raises latency only 1.2×. The near-flat curves reflect a constant implementation floor of ≈31 ms, on top of which the asymptotic terms of Equation (13) grow gently.

SES adds two costs to a fixed retriever–generator pipeline, measured separately on StrategyQA with $n = 100$, $k = 4$, $m \leq 5$: the one-shot sub-query decomposition takes 968 ms per query on average (a single call to Qwen2.5-14B-Instruct, GPU), while the selection algorithm itself, measured under the same protocol as Figure 4 (including per-query Gram-matrix and normalizer computation), takes 31.4 ms per query on a single CPU thread at $(n, k) = (100, 4)$. Two remarks put these numbers in perspective. **First**, the millisecond-scale selection validates Equation (13): at practical scales the asymptotic cost is a handful of small matrix operations dwarfed by fixed per-query implementation overhead, so selection latency is effectively flat. **Second**, the decomposition cost is per-query, not per-selection: sub-queries depend only on the query, so they can be cached and reused across rerankers, generators, budgets, and repeated runs, and in a serving deployment they are computed

once and amortized over all downstream selection decisions. If even the uncached latency matters, the decomposition can be shifted to a smaller instruct model, since it only needs to identify the query's facets, not to answer them; we leave this trade-off to future work.

Scaling. Figure 4 reports selection latency as the pool size and the budget vary (E5-base-v2, $d = 768$, single CPU thread, mean per query, including per-query Gram-matrix and normalizer computation). A 16-fold pool growth ($n = 100 \rightarrow 1600$) raises latency by only $2.1\times$ ($31.4 \rightarrow 64.9$ ms), and quadrupling the budget ($k = 4 \rightarrow 16$) raises it by $1.2\times$ ($31.4 \rightarrow 38.4$ ms): a constant implementation floor of $\approx$31 ms dominates. The gap between theory and observation is starkest along k: the $\mathcal{O}(nk^3)$ term grows 64-fold from k=4 to k=16, yet latency grows only $1.2\times$, and even along n the theoretical operation count at $(n, k) = (1600, 4)$ is roughly 16 times that at $(100, 4)$ against an observed $2.1\times$. In all configurations selection remains a sub-100 ms, CPU-only operation that negligibly loads the GPU-bound answer generation.