Practical Advances in Artificial Intelligence and Computing
Frontier Research Publishing • https://www.frpubs.org
Vol. 4, No. 4, 2025

Budget Constrained Evidence Selection for Retrieval Augmented Generation

Brecken Sloane[1]*
[1]Rice University, Houston, USA
*Corresponding author: brckn.s63@outlook.com

Abstract: Retrieval augmented generation must allocate limited context space among passages that jointly support an answer. Ranking passages by relevance alone may repeatedly cover a few requirements while omitting others. This study formulates evidence selection as budget constrained weighted coverage over predefined atomic evidence requirements. The objective is monotone and submodular. A heuristic combines marginal coverage per unit cost with comparison against the best feasible single passage. A separate coverage criterion checks both overall coverage and mandatory requirements before permitting a complete response. Numerical comparisons use 200 generated instances, each containing 12 candidate passages and six requirements, under balanced and concentrated relevance scores. At a budget of 600 nominal tokens in the concentrated setting, mean weighted coverage is 0.95160 for the coverage density heuristic, compared with 0.79445 for relevance first selection, 0.83620 for relevance per cost selection, and 0.97395 for exhaustive optimization. Mean selected passage cost is 442.0 nominal tokens for the heuristic and 593.5 for relevance first selection. The comparisons quantify weighted evidence allocation under supplied support labels.

Keywords: large language models; retrieval augmented generation; evidence coverage; submodular optimization; context budget

1. Introduction

Retrieval augmented generation (RAG) supplements the parametric knowledge of a large language model with retrieved documents [1]. Its practical attraction is that a question can be answered using information held outside the model. However, successful retrieval does not by itself establish that the context contains everything needed for a complete response. A technical question may require operating conditions, implementation steps, and failure boundaries. A collection dominated by descriptions of implementation can remain incomplete even when every passage is relevant to the general topic. Evidence allocation therefore concerns the joint value of selected passages, as well as their individual relevance.

Dense passage retrieval provides an effective mechanism for obtaining candidates [2]. A retrieval score nevertheless need not represent the additional information supplied by a passage after other passages have already been selected. Two high scoring candidates may support the same requirement, whereas a lower scoring candidate may supply a missing qualification. When passages have different lengths, selecting them by relevance also creates a resource allocation problem: an early choice can consume space needed for a complementary combination. The appropriate unit of analysis is consequently the selected set under its available context budget.

A larger context window does not remove the need to make this allocation explicit. Liu et al. found that relevant information was used differently depending on its position in the tested long context tasks [3]. ALCE distinguishes answer correctness from the quality of citation support [4]. These findings motivate separate

treatment of evidence availability, coverage of answer requirements, and faithful generation. This paper evaluates evidence coverage under supplied support labels.

The study addresses two questions. Under a common candidate pool and budget, how does explicit requirement coverage change the selected evidence? How does that behavior vary when relevance scores favor a small subset of requirements? We apply an established weighted coverage objective, evaluate a lightweight selection heuristic against exhaustive optimization, and connect the selected set to a separate criterion for complete responses. A relevance per cost baseline helps distinguish simple cost awareness from the broader effect of coverage based selection. The contribution is a reproducible analysis of this configuration problem, including its failure cases and operational boundaries.

2. Related Work

The original RAG formulation combines a generative model with nonparametric retrieval memory for knowledge intensive tasks [1]. Dense passage retrieval learns question and passage representations that support candidate search through vector similarity [2]. Self-RAG extends control into the generation process through learned retrieval and reflection behavior [5]. The present approach occupies a different location in the pipeline: an explicit selection layer after candidate retrieval and support assessment. It takes passage costs and supplied support labels as inputs.

Maximal marginal relevance (MMR) reduces redundancy by balancing relevance against similarity to previously selected material [6]. Weighted coverage instead values the requirements that a candidate can newly support. These approaches are related through their treatment of set complementarity, but document dissimilarity and additional requirement coverage are different quantities. A passage can differ in wording while repeating the same requirement, or share terminology while adding a necessary boundary condition. In the numerical comparison, MMR uses Jaccard similarity over known support sets so that its information input is explicitly defined.

Monotone submodular optimization supplies the mathematical background for coverage objectives [7]. Its guarantees depend on the constraint and algorithm. In particular, results for a cardinality constraint cannot be transferred without qualification to unequal passage costs. Sviridenko established the classical approximation guarantee for a knapsack constraint using an enumeration enhanced procedure [8]. We use a simpler heuristic and claim no equivalent guarantee. The broader evaluation literature also separates context quality, answer relevance, and faithfulness [9]. Accordingly, the analysis below evaluates the selection objective directly and leaves downstream answer quality as a distinct validation task.

Keyword-constrained retrieval specifies lexical conditions for candidate construction [10], while hierarchical encoding and selective attention organize long documents [11]. These choices affect which qualifications and complementary passages reach the selection stage. Context embedding methods compress multiple retrieved contexts with adjustable compression rates [12]. PISCO uses sequence-level distillation to learn a soft compressor [13]. Compression changes the representation of evidence; the coverage model instead makes the value of the selected passage set explicit.

Semantic matching with learned similarity treats correspondence as a representation problem [14]. Knowledge graph fusion for long-text classification and instruction-driven cross-attention alignment introduce relational and task-specific structure into text representations [15], [16]. Signal-aligned memory compression addresses which information is retained in a long context [17], while low-rank and gated sparse updates address adaptation of the underlying model [18]. These choices can affect the candidates presented to an evidence selector. Here, selection remains an explicit operation over supplied passage costs and support labels.

Structured semantic representations offer further ways to distinguish repeated wording from repeated information. Knowledge graph enhancement for sentiment classification and graph-based semantic alignment for entity recognition organize textual meaning around entities and relations [19], [20]. Frequent subgraph mining with structural prompting similarly places recurring relational patterns within a language-model adaptation workflow [21]. Cross-modal graph recommendation combines temporal behavior and social information, extending the representation question to heterogeneous signals [22]. These studies motivate examining whether evidence units preserve identifiable relations and information sources rather than relying only on surface similarity. In the present model, that distinction is expressed through passage-to-requirement support labels. Learning those labels from structured representations remains a separate problem; graph structure alone does not establish that a passage supports a particular answer requirement.

Agent planning introduces a related distinction between state description and action selection. Structured state representations with hierarchical scheduling and hierarchical attention with policy regularization address the organization of information used by a policy [23], [24]. Task-graph construction makes dependencies between planned operations explicit [25]. Evolutionary self-optimization and joint exploration with goal discovery consider adaptation when the environment or objective is not fixed [26], [27]. For evidence allocation, these directions suggest treating uncovered requirements as an observable state that changes after each selection. The present analysis retains a fixed candidate pool and a deterministic marginal-gain rule, allowing the effects of that state update to be measured directly. Adaptive retrieval or learned policies would require additional evaluation of how new observations change the requirement specification and available evidence.

3. Evidence Selection Model and Coverage Criteria

3.1 Problem Formulation

Let D contain n candidate passages and let U contain m atomic evidence requirements. A requirement specifies information needed for a complete response, such as an admissible input, a required operation, or a failure condition. The requirements and their nonnegative weights w_j are fixed before selection, and the weights sum to one. Support strength a_{ij} in [0,1] describes how passage i supports requirement j. It is a label or score, not a probability. The numerical study uses binary support. Repeated support for the same requirement is aggregated by the strongest selected support:

$$b_j(S) = max_{i \in S} a_{ij}, \quad f(S) = \sum_{j=1}^{m} w_j b_j(S) \tag{1}$$

The empty set has $b_j(\varnothing) = 0$. The weighted objective f therefore lies between zero and one. A positive passage cost c_i includes its text, title, citation marker, and separators. The available budget B is the part of the context remaining after the question, instructions, and reserved output space are accounted for. Evidence selection is formulated as

$$max_{S \subseteq D} f(S), \quad s.t. \sum_{i \in S} c_i \leq B \tag{2}$$

The maximum operation prevents repeated evidence for one requirement from accumulating credit without limit. A requirement already supported at strength one receives no further benefit from another equivalent passage. This captures a particular notion of sufficiency: each atomic requirement can be independently supported by at least one passage. It does not represent a conclusion that becomes supported only when two individually insufficient passages are combined. Such tasks require a representation of combinations or reasoning dependencies before the same interpretation can be used.

A deployed budget can vary with operating conditions. Attention-based traffic prediction and multiscale backend load modeling address anticipated demand [28], [29], while cross-scale temporal learning

distinguishes changing energy generation regimes [30]. Spatio-temporal graph forecasting treats multiple cloud resources jointly [31], and predictive GPU memory modeling addresses capacity in containerized applications [32]. For evidence selection, such forecasts could inform the budget supplied to Equation (2). The present experiment uses fixed costs and budgets for each selection decision.

Relational evidence may require a richer specification than independent requirements. Temporal risk-chain modeling and causal discovery with knowledge graphs represent connections between observations [33], [34]. Temporal evidence aggregation for failure diagnosis adds an ordering dimension [35]. Dynamic dependency graphs and contrastive modeling of microservice call graphs address changing system relationships [36], [37], while joint graph representation and causal inference target root cause localization [38]. Extending the coverage model to such dependencies would require defining which combinations of passages jointly support a requirement before evaluating the selection objective.

Requirement weights express task importance rather than normalized retrieval scores. This separation prevents one relevance bias from defining both the ordering and the evaluation criterion. Support assessment must also distinguish direct evidence from shared vocabulary. A passage mentioning a failure mode without explaining its conditions does not automatically support a requirement to identify those conditions. Contradictory sources and incompatible document versions must be resolved before selection, since the monotone objective cannot penalize a conflict introduced by an additional passage. The formulation assumes that these support and consistency decisions have already been made.

Latent state recursion and enterprise-network graph learning represent observations through changing states and relations [39], [40]. Dynamic graph modeling and temporal diffusion graph methods address corresponding representation problems in cloud monitoring [41], [42]. Topology-aware fault identification makes node relationships explicit [43], while distributed representation alignment and consistency-aware multisource modeling address reconciliation across observations [44], [45]. An automated support pipeline would need to retain source identity and dependencies before reducing evidence to a passage-to-requirement score. The current formulation takes those scores as inputs and leaves support inference outside the numerical analysis.

Automating support assessment also requires a choice of representation. Variational autoencoding for query-plan anomalies and diffusion-prior reconstruction for backend anomalies formulate different ways to characterize unexpected observations [46], [47]. Contrastive masked autoencoding learns anomaly representations from distributed tasks [48], while adaptive temporal state modeling and structural self-supervision retain changing states and dependencies [49], [50]. A support estimator built on such representations would still need validation against passage-level judgments. Its reconstruction or anomaly score would not by itself establish that an answer requirement is supported.

Collecting evidence across systems adds a reconciliation problem. Federated temporal modeling concerns collaborative detection across distributed tasks [51]. Heterogeneity-aware optimization with secure aggregation and secure parameter aggregation across cloud networks address information combined under distributed constraints [52], [53]. Before selection, source-specific outputs would need a common interpretation of support and a record of their provenance. The current matrix assumes that this reconciliation has already occurred.

3.2 Submodularity and Heuristic Selection

Proposition 1. For fixed support strengths and nonnegative weights, f is normalized, monotone, and submodular. Normalization follows from the empty set convention. Monotonicity follows because adding a passage cannot reduce the maximum support for any requirement. To establish diminishing returns, consider $A \subseteq T$ and $k \notin T$. The marginal gain satisfies

$$\Delta(k|A) = \sum_{j=1}^{m} w_j \max\{0, a_{kj} - b_j(A)\} \geq \Delta(k|T) \quad (3)$$

For every requirement, $b_j(A) \leq b_j(T)$, which establishes the inequality and completes the proof. Thus a passage has no greater additional value when a larger evidence set is already available. This is the established diminishing returns behavior of weighted coverage [7]. Its practical role here is to make the treatment of duplicate evidence explicit and to motivate recomputing marginal gains after each selection.

The coverage density heuristic starts from the empty set. At each iteration, it examines all remaining passages that fit within the unused budget and selects the one with the largest positive marginal gain per unit cost:

$$i^* = \arg\max_{i \notin S,\, c_i \leq B - c(S)} \frac{f(S \cup \{i\}) - f(S)}{c_i} \quad (4)$$

Here c(S) is the total cost of the selected set. The procedure stops when no feasible passage has a positive gain. Its result is then compared with the best feasible single passage. The set with higher coverage is retained; coverage ties favor lower cost and then the lexicographically smaller sorted sequence of passage identifiers. During the greedy iterations, equal density favors the smaller identifier. Maintaining the current support vector permits each candidate gain to be computed in O(m) time. At most n selection rounds then require $O(n^2m)$ time. This bound concerns the selection layer, excluding retrieval and support assessment.

Resource scheduling provides a broader context for sequential allocation. Adaptive throughput scheduling, hierarchical microservice orchestration, and scheduling under dynamic competition address decisions with limited resources [54], [55], [56]. Critical-task-prioritized workflow scheduling highlights essential work [57], while adaptive replica placement and reinforcement learning with meta-learning consider changing allocation conditions [58], [59]. Here, residual budget and current requirement support define the allocation state. Recomputing marginal gains updates the value of each passage as that state changes. The finite-instance comparison measures this allocation rule.

Allocation models also distinguish anticipated demand from enforceable constraints. Workforce forecasting with stochastic programming joins demand estimation to cost planning [60]. Risk-constrained elastic scaling and energy-budgeted edge-cloud scheduling explicitly incorporate operating restrictions [61], [62]. In the present selector, the passage budget is enforced at every step, while mandatory evidence coverage is checked afterward. Keeping these roles separate makes it possible to identify whether a failure concerns resource feasibility or the completeness of the selected evidence.

Comparison with a single passage does not eliminate local selection failures. Consider three requirements with weights (0.50, 0.25, 0.25), passage costs (1, 2, 2), and respective support sets {1}, {1,2}, and {1,3}, under budget 4. The first passage has the highest initial density. Adding the second yields coverage 0.75 and leaves insufficient budget for the third. The best single passage also covers 0.75, so the final comparison cannot recover complete coverage. Selecting the second and third passages would instead cover every requirement. This counterexample explains why an exact reference is useful and why empirical closeness to the optimum cannot substitute for a general approximation guarantee.

3.3 Coverage Criteria for Complete Responses

High average coverage can coexist with omission of a low weight requirement that is indispensable to the task. We therefore specify a mandatory subset $M \subseteq U$ before selection, along with thresholds η_j for its elements and an overall threshold τ. All thresholds lie in [0,1]. A selected set passes the coverage criterion when

$$Accept(S) = 1\{f(S) \geq \tau\} \cdot \prod_{j \in M} 1\{b_j(S) \geq \eta_j\} \quad (5)$$

The symbol $1\{\cdot\}$ denotes an indicator. If the criterion is not met, the system can retrieve additional evidence or decline a complete response. A partial response remains possible if its missing scope is made explicit, but it is not counted as complete. After a set passes the check, a generator may organize the supported information into an answer, followed by a separate inspection of citation support. Figure 1 locates these decisions within the workflow. Passing the criterion establishes coverage of the selected evidence under the supplied support labels.

For example, suppose a failure condition has weight 0.10 and every other requirement is fully supported. Overall coverage can reach 0.90 without that condition. An overall threshold of 0.90 alone permits this omission, whereas making the condition mandatory and requiring binary coverage of one prevents it. The criterion is a postselection check, not an additional constraint in Equation (2). The heuristic may therefore return a failing set even when another feasible combination could pass. Its rejection should be interpreted as a decision about the selected evidence, rather than proof that the candidate pool contains no adequate answer.

Joren et al. distinguish context sufficiency from the ability of a generator to use that context successfully [63]. This distinction supports evaluating the evidence preparation decision separately from the generated answer. Our coverage criterion makes the former decision inspectable under supplied support labels, while the latter requires an independent assessment of answer and citation quality.

Reliability-aware planning with conservative action gating separates permission to act from the choice of action [64]. Risk-aware language-model fine-tuning with tail-robust objectives also directs attention to failures that an average objective may underrepresent [65]. Mandatory requirements address a corresponding concern during evidence preparation, since high weighted coverage can conceal an indispensable omission. The thresholds remain task-defined conditions rather than calibrated risk estimates. Kahardipraja et al. use attention-head attribution to distinguish contextual information use from parametric knowledge [66]. Their analysis reinforces the need to examine how a generator uses evidence after the coverage criterion has passed.

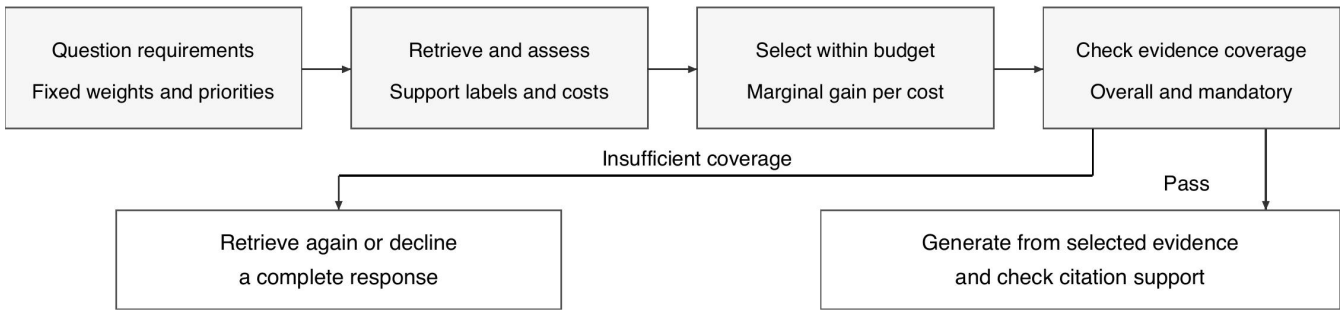

Figure 1. Budget constrained evidence selection and coverage checking.

4. Reproducible Numerical Analysis

4.1 Instance Generation and Comparison Protocol

The numerical study isolates redundancy and budget allocation using 200 finite instances. Each instance contains 12 candidate passages and six requirements. A single NumPy default_rng generator is initialized with seed 20251121. Each entry of the binary support matrix is independently one with probability 0.30. If any requirement has no supporting passage, the entire matrix is discarded and resampled. Conditional on an accepted matrix, integer passage costs are sampled uniformly from {1,2,3,4}, followed by a shared vector of uniform random values u_i in [0,1). These operations are repeated in that order for each instance.

The requirement weights are fixed at (0.25, 0.20, 0.18, 0.15, 0.12, 0.10). One abstract cost unit is defined as 100 nominal tokens. Budgets are 400, 600, 800, and 1000 nominal tokens. Every method receives the same generated support matrix, weights, and costs.

Each instance is evaluated under two relevance score settings. Balanced scores are $r_i = \Sigma_j w_j a_{ij} + 0.10u_i$. Concentrated scores are $r_i = 0.65a_{i1} + 0.25a_{i2} + 0.10u_i$, giving most of the score to support for the first two requirements. The settings share the same noise values as well as the same support and costs. They therefore form paired conditions on 200 instances, rather than 400 independent observations. Both settings are retained to show how the comparison depends on the relevance structure.

Table 1 specifies five selection methods. Relevance first and relevance per cost scan a fixed descending order and skip passages that no longer fit. MMR repeatedly maximizes $0.7r_i$ minus 0.3 times the largest Jaccard similarity to an already selected support set. Empty unions have similarity zero, and the first iteration has no redundancy penalty. MMR continues until no remaining passage fits, even when the current score is negative. Score ties favor the smaller identifier. Exhaustive optimization evaluates all 4096 subsets, retaining maximum coverage, then minimum cost, then the lexicographically smallest sorted identifier sequence. The coverage heuristic uses the rules in Section 3.2.

Table 1. Candidate selection methods and decision rules.

Method	Selection rule	Stopping or reference rule
Relevance first	Descending relevance score	Skip infeasible passages; continue until none fits
Relevance per cost	Descending relevance / cost	Skip infeasible passages; continue until none fits
MMR	0.7 × relevance − 0.3 × maximum similarity	Recompute among feasible passages at each step
Coverage density	Marginal weighted coverage / cost	Stop at zero gain; compare best feasible singleton
Exact	Enumerate all 4096 subsets	Maximize coverage, then minimize selected cost

For each setting, budget, and method, we report mean coverage, the sample standard deviation with denominator N − 1, and mean selected passage cost. The mean absolute optimality gap is

$$Gap = \frac{1}{N}\sum_{t=1}^{N}[f_t^*(B) - f_t(S_t)] \quad (6)$$

Here N is the number of instances, S_t is the selected set for instance t, and $f_t^*(B)$ is its exact optimal coverage under budget B. The gap is measured on the zero to one coverage scale and is not a relative error rate. All 8000 selection records are checked for budget feasibility and domination by the exhaustive optimum. An independent combinations based enumeration additionally checks objective values, costs, and tie decisions for the first ten instances at all four budgets.

4.2 Coverage under Different Budgets

Table 2. Coverage and selected cost at a budget of 600 nominal tokens (N = 200 per setting).

Score setting	Method	Coverage mean ± SD	Mean gap	Nominal tokens
Balanced	Relevance first	0.882 ± 0.122	0.092	597.5
Balanced	Relevance per cost	0.874 ± 0.135	0.100	581.5
Balanced	MMR	0.922 ± 0.100	0.052	597.5
Balanced	Coverage density	0.952 ± 0.083	0.022	442.0
Balanced	Exact	0.974 ± 0.058	0.000	451.5
Concentrated	Relevance first	0.794 ± 0.161	0.180	593.5
Concentrated	Relevance per cost	0.836 ± 0.148	0.138	588.5
Concentrated	MMR	0.813 ± 0.148	0.161	593.5
Concentrated	Coverage density	0.952 ± 0.083	0.022	442.0
Concentrated	Exact	0.974 ± 0.058	0.000	451.5

Table 2 reports the 600 nominal token condition. Under balanced scores, relevance first selection reaches mean coverage 0.88185, while coverage density reaches 0.95160. Under concentrated scores, relevance first falls to 0.79445 and coverage density remains at 0.95160, a difference of approximately 15.7 percentage points. The latter result is unchanged because the heuristic does not use r_i. Exhaustive optimization is likewise identical across settings, with mean coverage 0.97395. The identical coverage-based results follow from the common support matrix and method definitions.

MMR reaches 0.81285 under concentrated scores, compared with 0.92200 under balanced scores. Its redundancy penalty can redirect selection, but overlap among support sets is not identical to the weighted value of uncovered requirements. Relevance per cost reaches 0.83620 in the concentrated setting, improving on relevance first by incorporating passage cost. Coverage density still exceeds this cost aware baseline by 11.54 percentage points. The difference is consistent with an advantage from explicit requirement coverage in this instance family. It does not separately identify the effects of dynamic gain updates, the stopping rule, or comparison with the best single passage.

Coverage density has a mean absolute optimality gap of 0.02235 at this budget. It therefore remains suboptimal on some instances, in agreement with the counterexample. Its mean selected passage cost is 442.0 nominal tokens, compared with 593.5 for relevance first and 588.5 for relevance per cost under concentrated scores. The greedy stage stops when no feasible passage has positive marginal gain; the subsequent singleton comparison can return a set that admits a positive gain extension. The observed cost difference therefore includes both the stopping rule and the final comparison.

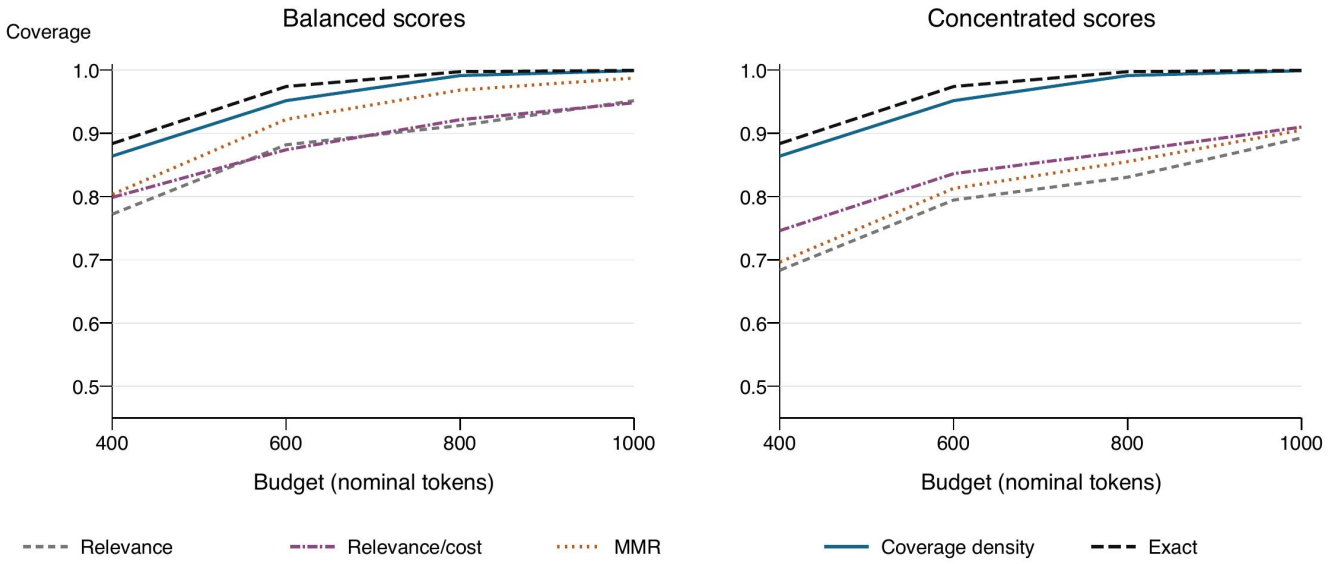

Figure 2. Mean weighted evidence coverage under different budgets.

Figure 2 shows mean coverage across the four budgets. The exact optimum is nondecreasing with budget because a larger budget retains every previously feasible subset. A greedy heuristic can follow a different selection path when the budget changes and has no corresponding monotonicity guarantee for an individual instance. Within the displayed averages, the coverage heuristic approaches saturation as the budget increases, while the concentrated relevance scores continue to favor repeated support for the first two requirements. They support examining incremental coverage when configuring context space, rather than assuming that filling all available space is itself beneficial.

5. Discussion and Limitations

The numerical results characterize a finite selection objective with known inputs. Coverage optimization depends on requirement quality and support labeling; downstream answer correctness and citation completeness require separate evaluation [9]. If a requirement specification omits a necessary condition, or a support label credits irrelevant evidence, even exact optimization can produce an inadequate context. Better optimization cannot repair a misspecified evidence representation.

The mtRAG benchmark evaluates retrieval and generation on human-generated multi-turn conversations, including questions that depend on earlier turns and questions that cannot be answered from the available evidence [67]. Such settings extend the evaluation beyond a fixed support matrix and require explicit checks of answerability and generated responses.

Evaluation over time requires checking whether the representation used for support assessment remains valid. Continual learning for cloud anomaly detection addresses ongoing model updates [68]. Semantic log modeling and behavior representation for inference-backend monitoring treat changing operational observations as inputs to analysis [69], [70]. A deployed evidence workflow should similarly record when support labels were assessed and whether source changes invalidate them. Reusing a previously selected context requires checking that its supporting information still applies.

The instance generator deliberately separates selection behavior from natural language uncertainty. Concentrated scores express one controlled bias and do not estimate the error distribution of a real retriever. The study uses small binary support matrices, fixed weights, and a fixed MMR coefficient. It does not cover multiple passage reasoning, conflicting evidence, or measured passage length distributions. Reported

standard deviations describe variation under this generator rather than uncertainty about a population of user queries. A larger evaluation would need independent questions and a fixed requirement annotation protocol before comparing downstream answers.

Practical deployment also introduces work outside the selection algorithm. Requirement decomposition and support checking may cost more than the input space saved by choosing fewer passages. Stable domain tasks could reuse requirement types or cache versioned support labels, while open questions may require fresh assessment. A review subset should record disagreements about whether a passage supports a requirement and why those disagreements occur. End to end evaluation should include this effort, retrieval, selection, generation, and citation checking. Only measurements of the complete process can establish whether the nominal context savings translate into an overall efficiency gain.

Joint corporate budget-revenue and labor expenditure planning treats allocation across coupled resource categories [71]. An operational RAG evaluation should likewise record retrieval, support verification, and generation costs separately, then compare their combined cost at a specified coverage target. This accounting would clarify whether context savings compensate for additional preparation effort.

The coverage criterion creates a further evaluation obligation. Mandatory requirements protect against omissions that a weighted average can conceal, but their designation reflects a task policy that must be fixed before inspecting the selected result. The threshold is not calibrated to an error probability. Its practical usefulness depends on whether it rejects incomplete contexts while retaining adequate ones. A future evaluation should therefore measure the correctness of complete response decisions alongside answer and citation quality. It should also distinguish a failure of the heuristic to find an adequate subset from the absence of adequate evidence anywhere in the candidate pool.

Distribution-aware risk prediction addresses changing operating conditions [72], while adaptive multi-agent risk modeling considers evolving transmission relationships [73]. Such changes may require revising requirements and support labels before repeating selection. Interpretable feedback introduces a separate interface objective, as illustrated by vision-language support systems [74]. When a complete response is withheld, the interface should identify missing requirements and the scope that remains supported. Evaluating that feedback separately would keep user-facing explanations distinguishable from the numerical coverage criterion.

The temporal scope of an observation can affect its interpretation. Structure-aware report summarization explicitly considers information fidelity [75], while cross-period anomaly learning and microstructure-aware sequence modeling represent observations at different temporal resolutions [76], [77]. Semi-supervised robust optimization addresses learning from limited noisy data [78]. These perspectives suggest testing support assessment across document periods and annotation quality levels. Such tests should measure whether a passage retains the conditions attached to its source, rather than awarding coverage for a superficially similar statement.

6. Conclusions

This paper connects atomic evidence requirements, passage costs, and coverage checks in a budget constrained RAG selection workflow. Weighted maximum coverage is monotone and submodular, and marginal coverage density supplies an explicit cost sensitive heuristic. Exhaustive optimization provides a reference that exposes its remaining losses. Across 200 reproducible instances, coverage based selection alleviates omissions associated with concentrated relevance scores and often uses less than the available nominal context budget. The results are limited to evidence allocation under supplied support labels. The practical value of the workflow depends on requirement specification, support verification, and faithful use of selected evidence, each of which requires separate evaluation.

References

[1] P. Lewis, E. Perez, A. Piktus, et al., Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, Advances in Neural Information Processing Systems, vol. 33, pp. 9459-9474, 2020.

[2] V. Karpukhin, B. Oguz, S. Min, et al., Dense Passage Retrieval for Open-Domain Question Answering, Proceedings of EMNLP, pp. 6769-6781, 2020. doi: 10.18653/v1/2020.emnlp-main.550.

[3] N. F. Liu, K. Lin, J. Hewitt, et al., Lost in the Middle: How Language Models Use Long Contexts, Transactions of the Association for Computational Linguistics, vol. 12, pp. 157-173, 2024. doi: 10.1162/tacl_a_00638.

[4] T. Gao, H. Yen, J. Yu, et al., Enabling Large Language Models to Generate Text with Citations, Proceedings of EMNLP, pp. 6465-6488, 2023. doi: 10.18653/v1/2023.emnlp-main.398.

[5] A. Asai, Z. Wu, Y. Wang, et al., Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection, The Twelfth International Conference on Learning Representations, 2024. arXiv: 2310.11511.

[6] J. Carbonell and J. Goldstein, The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries, Proceedings of the 21st Annual International ACM SIGIR Conference, pp. 335-336, 1998. doi: 10.1145/290941.291025.

[7] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, An Analysis of Approximations for Maximizing Submodular Set Functions—I, Mathematical Programming, vol. 14, pp. 265-294, 1978. doi: 10.1007/BF01588971.

[8] M. Sviridenko, A Note on Maximizing a Submodular Set Function Subject to a Knapsack Constraint, Operations Research Letters, vol. 32, no. 1, pp. 41-43, 2004. doi: 10.1016/S0167-6377(03)00062-2.

[9] S. Es, J. James, L. Espinosa Anke, et al., RAGAs: Automated Evaluation of Retrieval Augmented Generation, Proceedings of EACL: System Demonstrations, pp. 150-158, 2024. doi: 10.18653/v1/2024.eacl-demo.16.

[10] W. Ma and W. Zhong, Keyword-Constrained Retrieval-Augmented Generation for Large Language Models, vol. 4, no. 8, 2025.

[11] Z. Wang, A Hierarchical Encoding and Selective Attention Enhancement Framework Based on Large Language Models for Long-Document Text Classification, vol. 4, no. 11, 2025.

[12] D. Rau, S. Wang, H. Déjean, S. Clinchant, and J. Kamps, Context Embeddings for Efficient Answer Generation in Retrieval-Augmented Generation, Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining, pp. 493-502, 2025. doi: 10.1145/3701551.3703527.

[13] M. Louis, H. Déjean, and S. Clinchant, PISCO: Pretty Simple Compression for Retrieval-Augmented Generation, Findings of the Association for Computational Linguistics: ACL 2025, pp. 15506-15521, 2025. doi: 10.18653/v1/2025.findings-acl.800.

[14] R. Hu, Unified Semantic Matching with Logistic Similarity Learning for Cold-Start Advertising Recommendation, vol. 4, no. 2, 2024.

[15] Z. Zhu, Knowledge Graph-Augmented Semantic Fusion for Large Language Model-Based Long-Text Classification, vol. 4, no. 10, 2024.

[16] Y. Jiang and Y. L. Peng, Instruction-Driven Language Model for Text Classification via Cross-Attention Fusion and Semantic Alignment, vol. 4, no. 2, 2024.

[17] J. Zhang, Learning What to Remember: Key Signal-Aligned Memory Compression for Long-Context LLMs, vol. 3, no. 6, 2024.

[18] B. Chen, Adaptive Fusion of Low-Rank and Soft-Gated Sparse Updates for Parameter-Efficient Instruction Tuning, vol. 4, no. 6, 2024.

[19] Z. Gu and L. Nie, Knowledge Graph-Enhanced Large Language Models for Opinion Target Sentiment Classification, vol. 4, no. 7, 2025.

[20] J. Zhou, Graph-Based Knowledge Representation and Semantic Alignment for Large Language Model-Driven Entity Recognition, vol. 5, no. 6, 2025.

[21] Z. Xiao, C. Moretti, and Q. Li, Frequent Subgraph Mining and Structural Prompt Injection for Large Language Model Fine-Tuning, vol. 5, no. 7, 2025.

[22] Z. Ke, MCA-TAN: Cross-Modal Graph Neural Recommendation with Temporal Behavior and Social Network Fusion, vol. 5, no. 10, 2025.

[23] S. Liu, Structured State Representation and Hierarchical Policy Scheduling for Reinforcement Learning-Based Task Planning, vol. 5, no. 11, 2025.

[24] Q. Guo and L. Ma, Hierarchical Attention and Policy Regularization for Context-Aware Agent Decision-Making, vol. 5, no. 10, 2025.

[25] Q. Yang and K. Y. Lin, Autonomous Planning and Execution Method Based on Task Graph Construction for Large Language Model Agents, vol. 4, no. 7, 2025.

[26] Y. Liu, Evolutionary Strategy-Driven Self-Optimization for Adaptive Intelligent Agents in Dynamic Environments, vol. 4, no. 2, 2025.

[27] Y. Chen, S. Liu, and C. Qian, Unified Modeling of Autonomous Exploration and Goal Discovery for Open-Environment Agents, vol. 5, no. 9, 2025.

[28] Z. Qing and F. Huang, Attention-Enhanced Network Traffic Prediction for Intelligent Load Balancing Optimization, vol. 5, no. 8, 2025.

[29] N. Awate and Z. Chen, Research on Multi-scale Dynamic Evolution Time Series Modeling Methods for Predicting Backend System Load Change Trends, vol. 4, no. 10, 2025.

[30] H. Lu, Cross-Scale Temporal Representation Learning for Short-Term Wind-Solar Energy Generation Regime Classification, vol. 5, no. 3, 2025.

[31] F. Wang, D. Verma, and Y. Chen, Multi-Resource Workload Forecasting in Cloud Data Centers via Spatio-Temporal Graph Transformers, vol. 5, no. 8, 2025.

[32] X. Liu and K. Gong, Predictive GPU Memory Modeling in Containerized Cloud Environments Using Self-Supervised Multi-Scale Learning, vol. 5, no. 4, 2025.

[33] Y. Duan, Uncovering Hidden Financial Risk Chains through Temporal and Relational Knowledge Graph Learning, vol. 4, no. 2, 2024.

[34] L. Chen and S. Liu, Explainable Cross-Market Causal Discovery via Financial Knowledge Graphs, vol. 3, no. 1, 2024.

[35] Z. Hu, Temporal Evidence Aggregation with Causal-Contrastive Transformer Learning for Intelligent Cloud Failure Diagnosis, vol. 3, no. 8, 2024.

[36] F. Chen, Heterogeneous Dynamic Dependency Graph Learning for Anomaly Detection in Large-Scale Backend Systems, vol. 4, no. 7, 2024.

[37] Y. Gong, A. Mehta, and Z. Zhou, Research on a Graph Contrastive Learning-Based Anomaly Detection Method for Backend Microservice Call Graph Structure Change Awareness, vol. 4, no. 2, 2024.

[38] L. Wang and C. Y. Chang, Structure-Aware Root Cause Localization for Microservice Backends via Joint Graph Representation Learning and Causal Inference, vol. 3, no. 1, 2024.

[39] J. Chen and A. Tursun, Deep Latent Representation and State Recursion for Enterprise Audit Risk Detection, vol. 5, no. 2, 2025.

[40] X. Meng, Deep Graph Learning for Concealed Accounting Fraud Detection in Enterprise Networks, vol. 4, no. 9, 2025.

[41] G. Feng, Dynamic Graph Structure Modeling for Anomaly Detection in Complex Cloud Platforms, vol. 5, no. 7, 2025.

[42] B. Su, A Cloud Computing Anomaly Detection Method Integrating Diffusion Graph Neural Networks with Long- and Short-Term Temporal Dependency Modeling, vol. 5, no. 8, 2025.

[43] F. Xue, K. Wu, and A. Abdulla, Topology-Aware Graph Neural Networks for Distributed Node Fault Identification, vol. 5, no. 3, 2025.

[44] K. Yang and Z. Liu, Distributed Data Anomaly Detection Algorithm Integrating Federated Learning and Dynamic Representation Alignment Mechanism, vol. 4, no. 5, 2025.

[45] Y. Yang, M. Nagel, and A. Li, Consistency-Aware Unified Modeling for Multi-Source System Anomaly Detection, vol. 4, no. 4, 2025.

[46] L. Ren, S. Li, and Z. Liu, Variational Autoencoder-Based Query Plan Anomaly Detection and Cost Deviation Warning for Database Systems, vol. 4, no. 2, 2024.

[47] J. Chen, Discriminative Reconstruction Method for Backend Anomaly Detection Based on Diffusion-Generated Prior Enhancement, vol. 4, no. 2, 2024.

[48] J. Zhou, Contrastive Masked Autoencoding for Self-Supervised Anomaly Representation Learning in Distributed Task Systems, vol. 4, no. 2, 2024.

[49] Y. Zhan, Adaptive Representation Learning and Temporal State Modeling for Robust Anomaly Detection in Distributed Systems, vol. 3, no. 6, 2024.

[50] T. Ying and C. Ding, Self-Supervised Unified Representation Learning with Structural Dependency Modeling for Early Risk Detection in Distributed Systems, vol. 4, no. 2, 2024.

[51] K. Yang, Federated Learning and Self-Supervised Temporal Modeling for Collaborative Anomaly Detection in Distributed Tasks, vol. 4, no. 10, 2024.

[52] Y. Ke, Heterogeneity-Aware Federated Optimization with Secure Aggregation for Distributed Data Mining, vol. 4, no. 4, 2024.

[53] S. Zhang, Secure Parameter Aggregation and Distributed Anomaly Detection in Cross-Cloud Data Center Networks, vol. 3, no. 4, 2024.

[54] Y. Wang, Reinforcement Learning Based Adaptive Scheduling Method for Throughput Improvement in Distributed Computing Environments, vol. 5, no. 7, 2025.

[55] C. C. Peng, Joint Scheduling Method for Resource Orchestration Optimization in Distributed Microservices via Hierarchical Reinforcement Learning, vol. 4, no. 9, 2025.

[56] S. Shu, Adaptive Resource Scheduling in Distributed Systems via Reinforcement Learning under Dynamic Competition, vol. 5, no. 7, 2025.

[57] C. He and K. Y. Chen, Value-Iteration Reinforcement Learning for Distributed Cloud Workflow Scheduling: A Critical-Task-Prioritized Execution Optimization Framework, vol. 5, no. 2, 2025.

[58] B. Zhang and I. Abdurahman, Research on a Deep Reinforcement Learning-Based Adaptive Replica Allocation Method for Data Replica Placement Optimization in Distributed Storage Systems, vol. 4, no. 8, 2025.

[59] F. Xue, G. Zhu, and J. Taha, Intelligent Backend Service Scheduling through the Integration of Reinforcement Learning and Meta-Learning, vol. 4, no. 6, 2025.

[60] Y. Lin, Attrition-Aware Spatio-Temporal Graph Learning and Stochastic Programming for Multi-Horizon Workforce Demand Forecasting and Labor Cost Planning, vol. 4, no. 1, 2024.

[61] P. Hu, G. Uziel, and X. Ma, Risk-Constrained Reinforcement Learning for SLO-Driven Cloud-Native Elastic Scaling, vol. 3, no. 1, 2024.

[62] J. Sun, Reinforcement Learning-Based Task Scheduling Under Energy Budget Constraints in Edge-Cloud Systems, vol. 4, no. 4, 2024.

[63] H. Joren, J. Zhang, C.-S. Ferng, D.-C. Juan, A. Taly, and C. Rashtchian, Sufficient Context: A New Lens on Retrieval Augmented Generation Systems, The Thirteenth International Conference on Learning Representations, 2025.

[64] Z. Luo, Y. W. Hsu, and R. Qeyum, Safe Tool Use in LLM Agents via Reliability-Aware Planning and Conservative Action Gating, vol. 4, no. 7, 2025.

[65] Y. Wang, Risk-Aware Large Language Model Fine-Tuning with Tail-Robust Objectives and Feasible Gradient Updates, vol. 4, no. 3, 2025.

[66] P. Kahardipraja, R. Achtibat, T. Wiegand, W. Samek, and S. Lapuschkin, The Atlas of In-Context Learning: How Attention Heads Shape In-Context Retrieval Augmentation, Advances in Neural Information Processing Systems, vol. 38, 2025. doi: 10.52202/085713-3940.

[67] Y. Katsis, S. Rosenthal, K. Fadnis, et al., mtRAG: A Multi-Turn Conversational Benchmark for Evaluating Retrieval-Augmented Generation Systems, Transactions of the Association for Computational Linguistics, vol. 13, pp. 784-808, 2025. doi: 10.1162/tacl.a.19.

[68] S. Wang, Continual Learning Algorithm for Online Cloud Anomaly Detection, vol. 4, no. 7, 2024.

[69] Z. Yang, Transformer-Based Semantic Log Modeling for Robust and Explainable Cloud Security Analytics, vol. 4, no. 5, 2024.

[70] T. Ding and Y. Yuan, Behavior Representation Learning for Reliable Monitoring of AI Inference Backends, vol. 4, no. 1, 2024.

[71] N. Zhao, Y. Lin, and N. Abulikemu, Deep Reinforcement Learning Framework for Joint Corporate Budget-Revenue Allocation and Labor Cost Expenditure Planning, vol. 4, no. 11, 2025.

[72] L. Zheng, Enterprise Risk Prediction Under Dynamic Distribution Shift Using Robust Adaptive Learning, vol. 5, no. 10, 2025.

[73] S. Hu, Modeling Dynamic Financial Risk Transmission with Adaptive Multi-Agent Reinforcement Learning, vol. 5, no. 2, 2025.

[74] J. Y. Su, Vision-Language Model-Based Emotional Understanding and Interpretable Feedback for Art Therapy Support, vol. 5, no. 7, 2025.

[75] M. Wang, P. L. Peng, and J. Chen, Financial Report Summarization via Structure-Aware Modeling and Information Fidelity Constraints, vol. 4, no. 7, 2024.

[76] Z. Peng, Cross-Period Financial Statement Anomaly Detection via Contrastive Representation Learning, vol. 3, no. 8, 2024.

[77] Z. Su, Microstructure-Aware Risk Detection in High-Frequency Trading Using Embedding and Sequence Modeling, vol. 4, no. 3, 2024.

[78] Y. Mei, Unified Semi-Supervised and Robust Optimization Framework for Low-Resource Noisy Data Mining, vol. 3, no. 7, 2024.