AdaLoc: Adaptive Granularity and Hierarchical Filtering for Evidence-Aware Academic Search

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) improves factuality for scientific question answering, but scientific queries vary systematically in evidence scope. A single static retrieval pipeline faces a granularity–precision tension: strategies that maximize semantic coverage for *global* synthesis often miss needle-like evidence required for *local* extraction. We propose **AdaLoc**, a *scope-adaptive evidence localization* module. AdaLoc (i) routes queries by predicted scope, (ii) selects scope-specific evidence granularity, and (iii) applies a *hierarchical filter cascade* to balance semantic and lexical matching. We validate AdaLoc on the public QASPER benchmark, where it achieves competitive performance compared to closed-source baselines while using over 50% fewer tokens and purely open-source components. Furthermore, to rigorously probe the granularity trade-off, we introduce SCISCOPE, a diagnostic dataset of 2000 queries explicitly annotated with evidence scope. On SCISCOPE, AdaLoc substantially outperforms strong RAG baselines and long-context methods, particularly on local queries (improving F1 by over 3.7× compared to full-context baselines, 74.37% vs. 20.17%). These results demonstrate that precise evidence localization is more critical than simply extending context volume. Code and data are available at https://anonymous.4open.science/r/AdaLoc-EEC3.

1 Introduction

Large Language Models (LLMs) have revolutionized knowledge interaction (OpenAI, 2023) but often hallucinate (Ji et al., 2023). Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) addresses this by grounding responses in external evidence, improving factuality.

While successful in general domains, applying RAG to scientific question answering (QA) presents challenges due to the complexity of academic documents (Dasigi et al., 2021; Hong et al., 2024). Existing scientific retrieval systems, such as PaSa (He et al., 2025), primarily focus on *document-level* retrieval—identifying *which* papers are relevant. However, retrieving the correct paper is only the first step. The downstream challenge lies in *evidence localization*: identifying precisely *where* within a dense, multi-page paper the answer resides. Standard RAG pipelines often employ a static, "one-size-fits-all" chunking and retrieval strategy, which fails to accommodate the high variance in information density typical of scientific literature (Jiang et al., 2023; Asai et al., 2024).

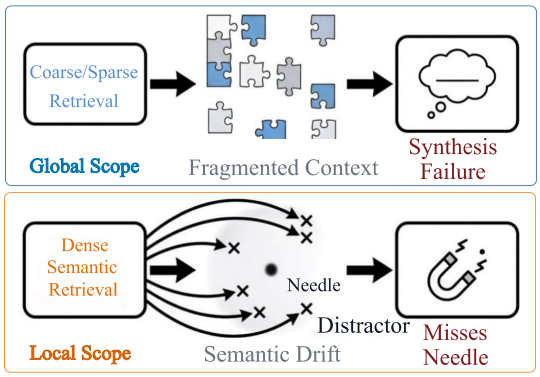

Figure 1: **The Granularity-Precision Trade-off.** (Top) GLOBAL queries fail with sparse retrieval due to context fragmentation; (Bottom) LOCAL queries suffer from semantic drift in dense retrieval. AdaLoc resolves this specific tension *in a principled way*.

Specifically, scientific queries differ systematically in *evidence scope*, creating a fundamental tension in retrieval granularity. We identify two distinct challenges. First, the **Global Synthesis Challenge**: Queries like "What are the main contributions?" require aggregating information dispersed across the introduction, method, and conclusion. Here, sparse lexical retrieval (e.g., BM25) often fragments the context, retrieving scattered keywords while missing the semantic holistic view required for comprehensive summarization.

Second, and conversely, the **Local Extraction Challenge**: Queries like "What is the dropout rate in Table 2?" target a sharply localized fact. Here, dense semantic retrieval often suffers from *semantic drift*, retrieving conceptually related passages (e.g., discussions on regularization) while missing the exact numeric cell. This results in a "needle-in-a-haystack" failure. Crucially, simply increasing context length is not a remedy, as long-context prompting suffers from the "lost-in-the-middle" effect (Liu et al., 2024), where decisive details buried in mid-document tables are overlooked.

To resolve this granularity-precision trade-off, we propose **AdaLoc** (Adaptive Localization), a scope-adaptive retrieval framework that treats evidence localization as a dynamic decision process. Rather than enforcing a uniform workflow, AdaLoc routes queries into distinct pathways based on predicted scope. For GLOBAL queries, it employs broader semantic chunking with LLM-based filtering to ensure comprehensive synthesis. For LOCAL queries, it switches to fine-grained chunking with strict lexical constraints (e.g., in-paper BM25) to pinpoint "needle-in-a-haystack" facts. This design allows AdaLoc to function as a plug-and-play downstream module for any document retrieval agent, bridging the gap between finding a paper and extracting the answer.

We evaluate AdaLoc using a two-stage protocol. First, to establish generalizability, we test on the established **QASPER** benchmark (Dasigi et al., 2021). Results show that AdaLoc achieves performance competitive with state-of-the-art methods while reducing token usage by over 50%, validating its efficiency on standard tasks. To further investigate the granularity-precision trade-off, we introduce SCISCOPE, a specialized benchmark explicitly designed to disentangle global synthesis from local extraction. On SCISCOPE, AdaLoc demonstrates superior capability, outperforming strong baselines (including Full-Context 128k and PaperQA) by a large margin on local queries, proving that disentangling retrieval strategies by scope is crucial for precise scientific QA.

Our contributions are as follows.

- We identify the scope-granularity mismatch in scientific QA and propose **AdaLoc**, a framework that adaptively modulates retrieval granularity and filtering logic.
- We demonstrate AdaLoc's general effectiveness on QASPER, and introduce SCISCOPE, a specialized benchmark to diagnosing the granularity-precision trade-off.
- Extensive experiments show that AdaLoc performs as well as reading the entire paper, proving that precise localization can achieve high accuracy without the heavy cost of processing long documents.

2 Related Work

2.1 Scientific Retrieval and Granularity

Scientific QA requires navigating complex documents. Traditional methods utilize dense retrievers (Beltagy et al., 2019; Karpukhin et al., 2020; Izacard et al., 2022) or agentic frameworks like PaperQA (Lála et al., 2023) and OpenScholar (Zheng et al., 2024) for document discovery. However, these systems often treat retrieved papers as monolithic blocks or apply fixed-size chunking (Pradeep et al., 2022), leading to precision loss in *intra-document* localization. To address granularity, methods like Atlas (Izacard et al., 2023), RAPTOR (Sarthi et al., 2024), and Mix-of-Granularity (Zhong et al., 2024) introduce hierarchical or learned chunking. Unlike these approaches, which require expensive pre-indexing or training, AdaLoc implements a zero-shot, *scope-adaptive* strategy. We treat granularity as a dynamic decision dependent on the query's evidence scope (GLOBAL vs. LOCAL), focusing specifically on the "last-mile" localization problem that document-level agents often overlook *in practice*.

2.2 Adaptive RAG Paradigms

Recent work actively modulates retrieval or reasoning. Agents like ReAct (Yao et al., 2023) interleave thought and action. Self-RAG (Asai et al., 2024) and FLARE (Jiang et al., 2023) decide *when* to retrieve; CRAG (Yan et al., 2024) and Adaptive-RAG (Jeong et al., 2024) route queries to different sources *as needed*. AdaLoc introduces an orthogonal dimension of adaptivity: *retrieval logic adaptation*. Instead of just triggering retrieval, AdaLoc determines *how* to retrieve—switching between coarse semantic synthesis and fine-grained lexical extraction. This allows AdaLoc to function as a specialized retrieval module within broader adaptive frameworks *more generally*.

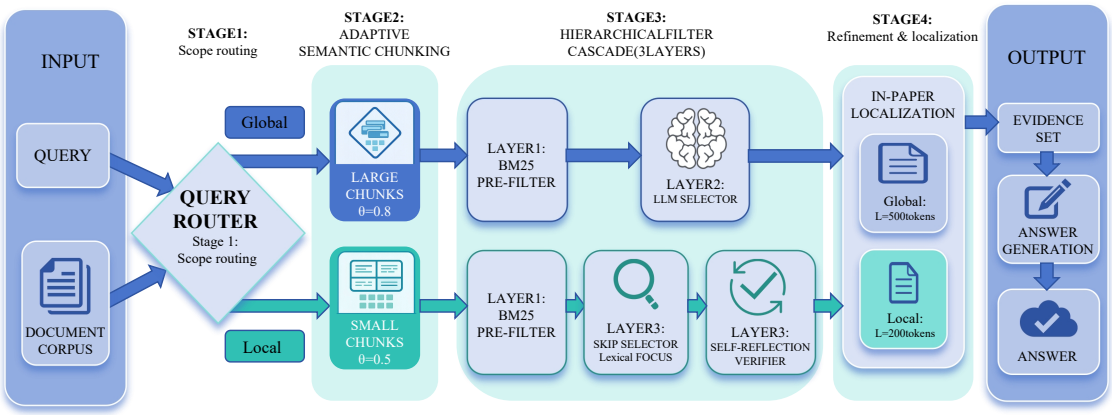

Figure 2: AdaLoc pipeline. The Query Router classifies queries as GLOBAL or LOCAL. Each pathway applies scope-specific chunking and filtering: GLOBAL uses broad semantic chunking with LLM selection, while LOCAL uses fine-grained lexical retrieval with BM25. Stage-2 localizes evidence within selected papers for answer generation.

3 Methodology

3.1 Problem Formulation

We formulate evidence-aware scientific QA as a conditional generation task grounded in retrieved segments. Let $\mathcal{D} = \{d_1, \ldots, d_N\}$ denote a corpus of scientific papers. Given a query q, the objective is to identify a specific evidence set $\mathcal{E}^* \subset d_i$ and generate an answer a maximizing $P(a|\mathcal{E}^*, q)$.

Unlike general open-domain QA, scientific queries exhibit a distinct dichotomy in *evidence scope*, denoted as $s \in \{\text{GLOBAL}, \text{LOCAL}\}$. We formally define these scopes based on the structural properties of the required evidence:

Global Scope ($s = \textbf{GLOBAL}$): Queries requiring synthesis across multiple non-adjacent sections of a document. Let $\mathcal{S}_q = \{s_1, \ldots, s_k\}$ be the set of sentences containing relevant evidence. A query is GLOBAL if $\max_{i,j} |pos(s_i) - pos(s_j)| > \delta_{\text{section}}$, where $pos(\cdot)$ returns the document position. Examples include *"What are the main contributions?"* or *"Summarize the methodology"*.

Local Scope ($s = \textbf{LOCAL}$): Queries targeting a single contiguous span, typically a numeric value, table cell, or equation. Formally, $|\mathcal{S}_q| \leq 3$ and all evidence sentences are within a single paragraph or table context. Examples include *"What is the learning rate?"* or *"What F1 score is reported on dataset X?"*.

A static retrieval granularity g creates a bottleneck: coarse granularity dilutes local signals, while fine granularity fragments global discourse. AdaLoc addresses this by modeling retrieval as a scope-dependent decision process $f(q) \rightarrow (s, g, \phi)$, where g is the chunking granularity (controlled by coherence threshold θ) and ϕ represents the filtering logic optimized for scope s.

3.2 System Overview

AdaLoc operates as a dynamic pipeline (Figure 2) that modulates retrieval behavior at three stages:

1. **Perception (Sec. 3.3):** A Query Router projects the query into a scope probability space to determine the optimal retrieval branch.
2. **Planning (Sec. 3.4):** Based on the predicted scope, the Adaptive Chunker dynamically segments retrieved documents into granularity-optimized units (context-preserving vs. precision-oriented).
3. **Execution (Sec. 3.5):** A Hierarchical Filter applies a cascade of lexical and semantic sieves, branching into distinct scoring mechanisms to reject noise while preserving scope-specific signals.
4. **Refinement (Sec. 3.6):** A final Last-Mile Localization step performs window-level reranking and citation traversal to ensure the generator receives high-density evidence.

This effectively mimics expert reading: skimming for synthesis while scanning experimental lists for precise extraction.

3.3 Step 1: Scope-Based Query Routing

The router serves as the control center, dispatching queries to avoid the "one-size-fits-all" trap (Li and Roth, 2002). We adopt a hybrid architecture that balances inference efficiency with semantic robustness, ensuring precise intent detection.

Signal-based Scoring. We define two lexical signal sets based on analysis of scientific query patterns. The LOCAL signal set $\mathcal{V}_{\text{loc}}$ contains 30+ extraction indicators organized into categories: (i) Numeric triggers: *"how many", "what value", "percentage"*; (ii) Hyperparameters: *"learning rate", "dropout", "batch size", "hidden dimension"*; (iii) Metrics: *"accuracy", "F1 score", "BLEU", "perplexity"*; (iv) Comparison: *"ablation", "compared to", "baseline", "outperform"*.

Conversely, the GLOBAL signal set $\mathcal{V}_{\text{glo}}$ contains 20+ synthesis indicators: *"main contribution", "what method", "describe the", "summarize", "overview", "key novelty"*. Each signal w is assigned a weight α_w (for LOCAL) or β_w (for GLOBAL) empirically tuned on a development set, typically in the range $[0.15, 0.40]$.

The scope probability $S_{\text{router}}(q) \in [0, 1]$ is computed via a weighted indicator function:

$$S_{\text{router}}(q) = \sigma\left(\sum_{w \in \mathcal{V}_{\text{loc}}} \alpha_w \mathbb{I}(w \in q) - \sum_{w \in \mathcal{V}_{\text{glo}}} \beta_w \mathbb{I}(w \in q) + \gamma\right) \tag{1}$$

where $\sigma(\cdot) = \text{clamp}(\cdot, 0, 1)$ bounds the score, $\mathbb{I}(\cdot)$ is the indicator function, and $\gamma = 0.5$ is the base score. This rule-based component handles $\sim$97% of queries with sub-millisecond latency.

Structural Features. Beyond lexical signals, we incorporate structural query features: (i) presence of numbers or percentages in the query (+0.25 weight); (ii) comparison words like *"higher", "better", "versus"* (+0.15); (iii) query length (short queries < 7 words favour GLOBAL at -0.15; long queries > 18 words favour LOCAL at +0.10). These features capture intent patterns not covered by lexical matching.

Semantic Fallback. For queries falling into an ambiguity margin $[\tau_{\text{low}}, \tau_{\text{high}}]$ (empirically $[0.48, 0.52]$), rule-based routing is unreliable. We trigger a lightweight LLM classifier using few-shot prompting with 10 exemplar pairs (5 GLOBAL, 5 LOCAL). The prompt structure includes category definitions and representative examples, enabling the model to discern intent for compositional queries, akin to chain-of-thought decomposition (Wei et al., 2022; Press et al., 2023).

3.4 Step 2: Candidate Retrieval & Adaptive Semantic Chunking

AdaLoc first retrieves the top-K candidate papers (default $K = 5$) using a document-level BM25 index. The core innovation lies in the subsequent lightweight Adaptive Semantic Chunking, which segments papers based on scope-conditioned coherence thresholds.

Algorithm 1 Hybrid Query Routing Strategy

Require: Query q, Signal sets $\mathcal{V}_{\text{loc}}, \mathcal{V}_{\text{glo}}$
Require: Thresholds $\tau_{\text{low}}, \tau_{\text{high}}$
$s_{\text{raw}} \leftarrow 0$
for token w in q **do**
  if $w \in \mathcal{V}_{\text{loc}}$ **then** $s_{\text{raw}} \leftarrow s_{\text{raw}} + \alpha$
  else if $w \in \mathcal{V}_{\text{glo}}$ **then** $s_{\text{raw}} \leftarrow s_{\text{raw}} - \beta$
  end if
end for
$P_{\text{loc}} \leftarrow \sigma(s_{\text{raw}} + \gamma)$
if $P_{\text{loc}} > \tau_{\text{high}}$ **then**
  return LOCAL
else if $P_{\text{loc}} < \tau_{\text{low}}$ **then**
  return GLOBAL
else
  ▷ Ambiguity detected, trigger fallback
  $s_{\text{llm}} \leftarrow \text{LLM\_Classify}(q)$
  return s_{llm}
end if

Dynamic Boundary Detection. Let $U = \{u_1, \ldots, u_m\}$ be the sequence of sentences in a paper. We compute the semantic coherence between adjacent sentences using a pre-trained sentence encoder (Reimers and Gurevych, 2019). Each sentence u_i is encoded into a dense vector $\mathbf{e}_i = \mathbf{E}(u_i) \in \mathbb{R}^{768}$, and the coherence score is:

$$\text{sim}(u_i, u_{i+1}) = \frac{\mathbf{e}_i \cdot \mathbf{e}_{i+1}}{\|\mathbf{e}_i\| \|\mathbf{e}_{i+1}\|} \tag{2}$$

A chunk boundary is triggered at position i when $\text{sim}(u_i, u_{i+1}) < \theta_s$, where θ_s is the scope-dependent threshold.

Token Length Constraints. Raw boundary detection may produce chunks that are too short (fragmenting context) or too long (exceeding model limits). We enforce token constraints through a post-processing step: (i) Merge: If a chunk has fewer than τ_{min} tokens (80 for GLOBAL, 50 for LOCAL), it is merged with the preceding chunk. (ii) Split: If a chunk exceeds τ_{max} tokens (400 for GLOBAL, 250 for LOCAL), it is split at sentence boundaries to stay within the limit.

Dual-Granularity Strategy. Global Pathway ($\theta = 0.8$): We enforce a high similarity threshold, triggering boundaries only at significant topic shifts. This produces Coarse Chunks (avg. 120-400 tokens) that encapsulate complete argumentative units (e.g., an entire methodology paragraph), thereby avoiding context fragmentation and preserving the discourse flow essential for synthesis.

Local Pathway ($\theta = 0.5$): We use a lower threshold to tolerate lower semantic coherence. This is critical for LOCAL queries, as tabular data or experimental lists often exhibit low sentence-to-sentence similarity due to their heterogeneous structure. A lower threshold merges these elements into a single context, preventing the accidental separation of a value from its header.

3.5 Step 3: Hierarchical Evidence Filtering

Retrieving the right chunks is insufficient; we must also rank them correctly. We introduce a *bifurcated filtering logic* that addresses the semantic drift problem inherent in vector search.

Layer 1: Universal Lexical Pre-filter (BM25). To handle large candidate sets efficiently, we first apply a BM25 filter (Robertson and Zaragoza, 2009) over all generated chunks, computing the Okapi BM25 score for each chunk c:

$$S_{\text{BM25}}(c, q) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{tf(t, c) \cdot (k_1 + 1)}{tf(t, c) + k_1 \cdot (1 - b + b \cdot \frac{|c|}{\text{avgdl}})} \tag{3}$$

where $k_1 = 1.5$ and $b = 0.75$ are standard parameters. We retain the top-k candidates per paper (default $k = 5$), pruning chunks missing basic keyword matches early.

Layer 2: Scope-Specific Scoring. This layer implements the key insight that GLOBAL and LOCAL queries require fundamentally different scoring strategies:

Path A: Global (Semantic Focus). GLOBAL queries rely on topical alignment ("aboutness") rather than exact keyword matching. We employ an LLM-based Selector that estimates the relevance probability $P(\text{relevant}|q, c)$. This model identifies chunks that discuss the target topic conceptually, effectively bridging the lexical gap similar to neural expansion (Gao et al., 2022). Chunks scoring below threshold $\tau_{\text{sel}} = 0.5$ are filtered out.

Path B: Local (Lexical Focus). For LOCAL extraction, we bypass the semantic selector to avoid over-smoothing, relying directly on BM25 ($S_{\text{final}} = S_{\text{BM25}}$). This "hard" filtering preserves specific values often suppressed by vector-based ranking.

Layer 3: Self-Reflection (Local Only). For LOCAL queries, precision is paramount. Top candidates from Path B undergo a verification step where an LLM acts as a verifier. The prompt asks: (i) Does this chunk *directly* answer the query? (ii) Is the information specific enough? (iii) Could it cause hallucination? The verifier outputs an adjusted score $S_{\text{ref}} \in [0, 1]$, penalizing chunks that are lexically similar but semantically irrelevant, like baseline results sharing identical metric names. This step reduces false positives by approximately 15% based on our ablation analysis.

3.6 Step 4: Refinement & Localization

The final stage bridges the gap between retrieved chunks and the generation prompt, ensuring the "last mile" of retrieval is accurate.

Precision Expansion (Citation Chasing). Scientific evidence is often non-local (e.g., *"We follow the setup in [12]"*), requiring multi-hop reasoning (Yang et al., 2018). AdaLoc detects citation markers via regex patterns. For LOCAL queries, which often require traversing experimental lineage, we resolve these references against the corpus and recursively expand the candidate pool with chunks from cited papers (Depth=3), enabling multi-hop retrieval without retrieving full documents.

In-Paper Window Localization. To minimize the "lost-in-the-middle" effect, we perform a second-pass localization within selected papers. We slide a dense window of size L (e.g., 200 tokens for LOCAL) across the top papers with a stride of $L/2$ to prevent boundary truncation. These windows are then re-ranked using a lightweight cross-encoder (Nogueira and Cho, 2019), and we retain only the top-N scoring windows (default $N = 3$) as the final context. This mechanism compresses a 10k-token document into a refined context of less than 600 tokens, significantly reducing computational costs while filtering out irrelevant noise that often confuses the generator.

3.7 Answer Generation

The finalized evidence set $\mathcal{E}^*$ is concatenated into a prompt. We use scope-aware instructions: for GLOBAL queries, the model is prompted to synthesize information across chunks, highlighting connections between sections (e.g., method and results). For LOCAL queries, we enforce a strict "extraction" mode that forbids paraphrasing. Crucially, if the evidence is missing, the model is guided via few-shot examples to output a *Not Found* token, preventing hallucination and ensuring reliability.

Method	Overall				Global		Local	
	F1↑	LLM%↑	P-Rec↑	Tokens↓	F1↑	LLM%↑	F1↑	LLM%↑
Direct LLM	18.51	18.10	0.00	**41**	25.35	18.58	12.02	17.64
Naive RAG	29.40	52.25	69.62	1,500	33.43	38.09	25.57	65.69
PaperQA†	58.29	83.50	44.05	3,063	47.24	84.50	68.79	82.55
RAPTOR†	48.77	79.10	41.25	2,989	45.45	94.66	51.92	64.33
GraphRAG†	38.42	66.55	25.75	2,526	47.22	**96.92**	30.07	37.72
Full-Context (128K)†	37.14	91.25	**92.59**	6,984	55.03	88.19	20.17	**94.15**
AdaLoc (Ours)	**65.62**	**92.30**	84.66	738	**56.86**	96.10	**74.37**	88.50

Table 1: Performance on SCISCOPE ($N = 2000$). † denotes methods with *oracle* access to the ground-truth target paper (no paper discovery). AdaLoc outperforms oracle Full-Context on LOCAL queries (+54 F1) while using 89% fewer tokens; Full-Context retains an advantage on GLOBAL synthesis where full document context aids synthesis.

4 Experiments

4.1 Experimental Setup

We introduce SCISCOPE ($N = 2000$), a diagnostic benchmark explicitly designed to test the granularity-precision trade-off by decoupling GLOBAL synthesis from LOCAL extraction (construction details in Appendix K). We compare AdaLoc against three categories of baselines: General RAG (Naive/Hybrid), Scientific Agents (PaSa, PaperQA), Long-Context Methods (Full-Context 128K), and Structure-Aware RAG (RAPTOR, GraphRAG), with full implementation details in Appendix A. Following standard practice, we report **F1**, **LLM-Eval Accuracy**, **Passage Recall**, and **Token Usage** (see Appendix C for definitions).

4.2 Main Results

Table 1 summarizes our findings. AdaLoc achieves the highest overall F1 (**65.62%**) and passage recall (**84.66%**), confirming the effectiveness of adaptive constraints. Crucially, on LOCAL queries, it outperforms the oracle Full-Context baseline by **54 F1 points** (74.37% vs. 20.17%), demonstrating that precise evidence localization is more effective than standard long-context processing for extraction tasks. In terms of efficiency, AdaLoc uses 89% fewer tokens than the 128K baseline (738 vs. 6,984), offering a superior cost-performance ratio. Compared to agentic baselines (PaperQA, RAPTOR, GraphRAG), AdaLoc yields higher overall accuracy without the massive token overhead. This performance gap is statistically significant ($p < 0.01$), underscoring the necessity of adaptive mechanisms over static retrieval pipelines for complex heterogeneous scientific queries.

Method	Overall F1	Global F1	Local F1	Local EM	Ctx Tokens
Hybrid RAG	62.88%	48.23%	83.99%	63.24%	1212
Hierarchical RAG	61.05%	47.23%	80.97%	61.76%	1332
Naive RAG + Stage-2	62.93%	47.98%	84.47%	58.82%	1407

Table 2: Token-matched strong baselines (∼1.5K context tokens per query).

4.3 Token-Matched Strong Baselines

To ensure a fair comparison among retrieval-heavy systems, we evaluate three strong baselines under a matched evidence budget of 1500 context tokens per query: Hybrid RAG, Hierarchical (small-to-big) RAG, and Naive RAG with our Stage-2 in-paper localization (without routing). As shown in Table 2, the three baselines become broadly comparable once the context budget is controlled, indicating that much of the apparent gap among strong retrievers can be attributed to the amount of evidence provided to the generator rather than the retriever alone. We additionally report answer-token overlap@K as a lightweight proxy for evidence quality (Table 10); Hierarchical retrieval yields lower overlap, consistent with its slightly lower local extraction scores under the same budget.

The advantage is most pronounced on LOCAL queries (+54 F1 over 128K baseline), confirming that precise evidence localization outperforms simply extending context length. On GLOBAL queries, AdaLoc achieves competitive performance (56.86% vs. 55.03%), while Full-Context benefits from complete document access. Crucially, AdaLoc uses only 738 tokens per query—89% fewer than the 128K baseline (6,984 tokens).

This confirms that evidential *quality* outweighs *quantity*. While baselines reduce search space, they fail to separate "needle" facts from noise. AdaLoc's explicit constraints prevent the model from being distracted by proximal but irrelevant numbers, mitigating common dense retrieval failures.

4.4 Ablation Study

To disentangle the sources of AdaLoc's performance, we systematically remove key components and observe the degradation in Overall, GLOBAL, and LOCAL F1 scores. The results, summarized in Table 3, reveal that while all modules contribute, in-paper localization is the foundational driver of performance, while adaptive granularity and hierarchical filtering provide critical refinement.

Variant	All F1	Glo. F1	Loc. F1
AdaLoc (Full)	**65.62**	**56.86**	**74.37**
w/o Router	54.85	–	54.85
w/o Adapt. Chunk	61.43	52.93	69.92
w/o Ctx. Retrieval†	37.23	56.19	18.27

Table 3: Ablation results on SCISCOPE. †Removing context retrieval causes the largest drop, with LOCAL F1 collapsing from 74.37% to 18.27%.

Impact of In-Paper Localization (Critical). Removing the Stage-2 localization step ("w/o Ctx. Retrieval") yields the most catastrophic performance drop: Overall F1 plummets by 28.4 points (65.62% $\rightarrow$ 37.23%). This impact is highly asymmetric: LOCAL F1 collapses from 74.37% to 18.27%, while GLOBAL F1 remains relatively stable (56.86% $\rightarrow$ 56.19%). **Analysis:** This clearly confirms our core hypothesis that document retrieval alone is insufficient for fact-extraction queries. Without the fine-grained BM25 localization window ($L = 200$), the generator is fed broader, noisier chunks where specific values (e.g., hyperparameters in tables) are either truncated or diluted by surrounding text, leading to hallucination *in practice*.

Impact of Adaptive Granularity. Replacing our scope-conditioned chunking with a fixed strategy ("w/o Adapt. Chunk", fixed at 400 tokens) reduces performance by 4.2 points (65.62% $\rightarrow$ 61.43%). **Analysis:** This further validates the granularity-precision trade-off. Fixed chunks are a suboptimal compromise: they are often too coarse for LOCAL precision (introducing noise) and too fragmented for GLOBAL synthesis (breaking discourse flow). AdaLoc's adaptive boundary threshold ($\theta = 0.8$ vs. 0.5) successfully decouples these conflicting requirements *in a robust manner*.

Impact of Query Routing. Disabling the router and treating all queries as the default LOCAL class ("w/o Router") leads to a 10.8-point drop (65.62% $\rightarrow$ 54.85%). **Analysis:** While LOCAL processing is robust, applying it to GLOBAL queries forces the model to extract specific spans rather than synthesizing broad answers, degrading GLOBAL F1. The high accuracy of our Router ensures that each query is served by its specialized optimal pathway.

In summary, AdaLoc is not merely a sum of parts; its performance relies on the synergy between identifying the evidence scope (Router), preparing the right data shape (Chunker), and applying the correct selection logic (Filter).

4.5 Diagnostic Analysis & Ablation

To disentangle the sources of AdaLoc's performance and validate our design choices, we conduct a comprehensive ablation study coupled with mechanism analysis. Table 3 summarizes the contribution of each module.

Impact of In-Paper Localization (The "Needle" Hypothesis). Removing Stage-2 localization ("w/o Ctx. Retrieval") causes the most catastrophic drop: Overall F1 plummets by 28.4 points (65.62% $\rightarrow$ 37.23%), with LOCAL F1 collapsing from 74.37% to 18.27%. **Mechanism:** This confirms that LOCAL queries suffer from the "lost-in-the-middle" effect (Liu et al., 2024). Without the fine-grained BM25 localization window ($L = 200$), the generator is fed broader chunks where specific values (e.g., hyperparameters) are diluted by surrounding text. AdaLoc's targeted retrieval concentrates attention on the "needle," making localization more valuable than context volume.

Impact of Hierarchical Filtering (Why Skip Selector?). Enabling the semantic Selector for LOCAL queries actually *hurts* performance (Table 4). **Mechanism:** The Selector, trained for topical relevance, tends to over-filter chunks containing isolated numerical values that appear less "topically relevant" but are factual answers. By bypassing the Selector and relying on lexical constraints for LOCAL queries, AdaLoc prevents this semantic drift, ensuring that precision-critical evidence is preserved *in the final context*.

Method	F1	LLM%	P-Rec
LOCAL w/ Selector	57.58	64.06	64.34
LOCAL w/o Selector	**74.37**	**88.50**	**86.11**
Δ	**+16.79**	**+24.44**	**+21.77**

Table 4: Selector ablation on LOCAL queries . Enabling the Selector hurts F1 by 18.5 points, confirming that semantic filtering over-prunes needle facts.

Impact of Adaptive Granularity. Replacing adaptive chunking with a fixed strategy ("w/o Adapt. Chunk") reduces performance by 4.2 points (65.62% $\rightarrow$ 61.43%). **Mechanism:** Fixed chunks enforce a suboptimal compromise: too coarse for LOCAL precision (introducing noise) and too fragmented for GLOBAL synthesis (breaking discourse flow). AdaLoc's adaptive threshold ($\theta = 0.8$ vs. 0.5) decouples these conflicting requirements.

Router Reliability. Disabling the router ("w/o Router") degrades performance substantially by 10.8 points (65.62% $\rightarrow$ 54.85%). Our rule-based router achieves 97.1% accuracy with sub-millisecond latency. Error analysis (Appendix B) reveals that misclassifying a LOCAL query as GLOBAL is significantly more harmful than the reverse, validating our use of a neutral base score ($S_{\text{base}} = 0.48$) to avoid systematic bias.

In summary, AdaLoc's performance relies on the synergy between identifying evidence scope (Router), preparing the right data shape (Chunker), and applying correct selection logic (Filter).

4.6 Efficiency & Robustness

Pareto Efficiency. AdaLoc achieves a Pareto improvement over baselines, dominating the trade-off curve between accuracy and cost. Compared to Full-Context (128K), AdaLoc uses 89% fewer API tokens (738 vs. 6,984) while determining the correct answer more often (+28.5 F1 points). This efficiency gain transforms the economics of scientific QA: processing 2,000 queries with Full-Context requires approximately 14M tokens (costing over $70 at standard API rates), whereas AdaLoc reduces this to $\sim$1.5M tokens ($7.50). This 10$\times$ cost reduction makes high-precision automated literature review accessible for individual researchers and large-scale meta-analyses.

Robustness to Hyperparameters. Sensitivity analysis (Appendix G) confirms robustness. Adaptive chunking outperforms fixed baselines by over 4 F1 points, while varying retrieval depth and decision thresholds yields minimal fluctuation (< 1.5 F1), proving performance relies on the adaptive design rather than intricate tuning.

Error Analysis: Case Studies. We examine representative failure modes to understand *why* baselines fail where AdaLoc succeeds. **Case A (Local Extraction):** For the query "*What is the learning rate used in training?*", the oracle Full-Context model is overwhelmed by the document's length. It attends to a generic statement about "*standard learning rate schedules*" in the Related Work section, resulting in a plausible but hallucinated paraphrase. In contrast, AdaLoc's LOCAL pathway utilizes strict lexical anchoring to pinpoint the exact string "*lr = 3e-4*" buried within the tabular "Hyperparameters" subsection, ignoring semantically related but factually irrelevant text. **Case B (Global Synthesis):** For the query "*What are the main contributions?*", Naive RAG retrieves isolated sentences scattered across the Abstract and Conclusion due to keyword matching. This fragmentation prevents the generator from forming a complete picture. AdaLoc's GLOBAL pathway, employing coarse semantic chunking ($\theta = 0.8$), successfully encapsulates the entire "Contributions" list from the Introduction as a single coherent unit. This preservation of local discourse structure enables the generator to synthesize a summary that accurately reflects the paper's logical flow.

4.7 External Validity on Public Benchmarks

To verify generalizability, we evaluate AdaLoc on **QASPER** (Dasigi et al., 2021) (preferring it over short-answer sets like PubMedQA (Jin et al., 2019)), mapping its *Extractive* queries to our LOCAL pipeline and *Abstractive* to GLOBAL. As detailed in Appendix D (Table 9), results mirror our SCISCOPE findings: AdaLoc outperforms baselines on Extractive queries (33.40 vs. 32.77 F1) and remains competitive overall (29.94 vs. 30.52 F1) despite strictly lower token usage (1.3k vs. 2.7k). While Abstractive performance trails the full-context oracle, this confirms AdaLoc's efficiency as a robust localization module for fact-seeking queries in the wild.

5 Conclusion

We presented **AdaLoc**, a scope-adaptive localization module. By dynamically routing queries to semantic or lexical pathways, it resolves the granularity-precision tension. Experiments on SCISCOPE show that accurate localization is essential for needle-like queries, outperforming long-context baselines by 54 F1 points while saving 89% tokens. External validation on QASPER confirms its robustness on extractive tasks, though synthesis capabilities can be further improved. Future work includes improving robustness to noisy PDF parsing and strengthening synthesis grounding.

6 Limitations

AdaLoc inherits limitations of pipeline RAG systems: performance depends on upstream PDF-to-text quality (e.g., flattened tables/formulas), and scope is modeled as a binary GLOBAL/LOCAL decision for fast routing. More compositional queries and multimodal evidence (e.g., figures) are not fully addressed and remain for future work.

7 Ethical Considerations

Scientific QA may produce plausible but incorrect outputs; AdaLoc reduces this risk by enforcing evidence grounding and allowing "Not found" when support is absent. All benchmarks use open-access arXiv papers (CC-BY/CC0), and outputs should be verified against the source text, especially in high-stakes domains.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *Preprint*, arXiv:1903.10676.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, and Weinan E. 2025. Pasa: An llm agent for comprehensive academic paper search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11663–11679.

Yanning Hong, Liangming Pan, and Zhijing Lin. 2024. Scholarqa: Answering scientific questions with evidence from multiple papers. *arXiv preprint arXiv:2405.00106*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *TMLR*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, and 1 others. 2023. Few-shot learning with retrieval augmented language models. *JMLR*, 24:1–43.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqiu Sun, Qian Liu, Jane Dwyer, Daniel Andor, Yizhong He, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *EMNLP*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Jakub Lála, Odhran O'Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G Rodriques, and Andrew D White. 2023. Paperqa: Retrieval-augmented generative agent for scientific research. In *arXiv preprint arXiv:2312.07559*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, pages 1–7.

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

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2022. Squeezing water from a stone: A bag of tricks for intra-document retrieval. *arXiv preprint arXiv:2205.02242*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *EMNLP*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3982–3992.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. In *Foundations and Trends in Information Retrieval*, volume 3, pages 333–389.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *ICLR*.

Y Zheng, R Zhang, J Zhang, Y Ye, Z Luo, Z Ma, and 1 others. 2024. Openscholar: Synthesizing scientific literature with retrieval-augmented lms. *arXiv preprint arXiv:2411.14199*.

Zijie Zhong, Hanwen Liu, Xiaoya Cui, Xiaofan Zhang, and Zengchang Qin. 2024. Mix-of-granularity: Optimize the chunking granularity for retrieval-augmented generation. *arXiv preprint arXiv:2406.00456*.

A Implementation Details

Table 5 lists the full hyperparameters for AdaLoc, showing the scope-specific configuration for Stage-1 chunking/filtering and Stage-2 localization.

Parameter	Global	Local
Stage-1: Semantic Chunking (for scoring)		
Similarity threshold θ	0.8	0.5
Token bounds $[\ell_{\min}, \ell_{\max}]$	[20, 120]	[120, 400]
Stage-1: Hierarchical Filtering		
BM25 top-k (Layer 1)	5	3
LLM Selector (Layer 2)	✓	–
Self-Reflection (Layer 3)	–	✓
Evidence frontier size M	3	3
Precision Expansion		
Traversal depth	1	3
Max citations per chunk	3	3
Stage-2: Answer Localization (for generation)		
Window size L	500	200
Windows per paper	2	3

Table 5: Hyperparameters for AdaLoc.

Scope	Routing Signals
LOCAL	"how many", "what value", "dataset size" "hyperparameter", "learning rate", "Table" "exact", "specific", "number"
GLOBAL	"contribution", "propose", "methodology" "summary", "overview", "main idea" "difference", "compare", "approach"

Table 6: Representative routing signals (Appendix). Base score is 0.55.

B Router Analysis (Extended)

Learned Router Baseline (Paper-level Generalization). To assess whether routing generalizes across papers, we run 5-fold cross-validation on SCISCOPE ($N = 2000$). We compare our fixed rule router to a lightweight learned baseline (TF-IDF features + logistic regression). As shown in Table 7, the learned router achieves near-perfect accuracy (99.1% vs. 97.1%), confirming that routing is a well-defined task that can be almost entirely solved with minimal supervision. Since AdaLoc is router-agnostic, this learned router can directly replace our rules when labeled data is available; we retain rule-based routing to preserve our zero-shot deployment.

Router	Acc (mean±std)	Macro-F1 (mean±std)
Rule-based (ours)	97.10	–
Learned (TFIDF+LR)	**99.10±0.46**	**99.10±0.46**

Table 7: Router comparison on SCISCOPE using 5-fold cross-validation ($N = 2000$).

Router Error Impact on End-to-End F1. We group queries by router correctness and report average F1 (Table 8). Correctly routed queries achieve 28.5 F1 points higher than incorrectly routed ones, confirming that routing errors are highly detrimental.

Router Correctness	N	Avg F1
Correct (97.1%)	1942	66.5%
Wrong (2.9%)	58	38.0%
Gap	–	**28.5 pts**

Table 8: End-to-end F1 grouped by router correctness.

C Metrics and Reproducibility

Metrics Definition. We employ three primary metrics:

1. **F1 Score:** Measures the token-level overlap between the predicted answer and the ground truth. We use the standard SQuAD-style normalization (removing punctuation, articles, etc.).

2. **LLM-Eval Accuracy:** A judge model (`DeepSeek-Chat`) assesses whether the semantic meaning of the prediction matches the ground truth, outputting a binary CORRECT/INCORRECT label.

3. **Passage Recall (P-Rec):** A binary metric indicating whether the top retrieved chunks contain the answer string.

We additionally track **API Tokens**, summing the input tokens sent to the generator.

Reproducibility. All experiments are conducted on a single NVIDIA A100 (80GB) GPU.

- **LLM:** We use `deepseek-chat` (DeepSeek-V2) via API for generation, reasoning, and judgment.
- **Embeddings:** We use `BAAI/bge-base-en-v1.5` (768d) for semantic scoring.
- **Tokenization:** Token counts are calculated using the `cl100k_base` tokenizer.

Method	Overall F1	Breakdown (F1) Extractive	Abstractive	Avg Tokens
Naive RAG	19.67	21.46	12.30	424
Full-Context†	**30.52**	32.77	**21.21**	2,730
AdaLoc (Ours)	29.94	**33.40**	15.66	**1,303**

Table 9: External validation on QASPER. AdaLoc matches the oracle Full-Context baseline in overall performance while dominating on Extractive questions, validating the generalizability of our scope-adaptive strategy.

Method	Overlap@5	Overlap@10	Overlap@20
Hybrid RAG	89.41%	91.32%	92.55%
Hierarchical RAG	80.70%	85.42%	88.45%
Naive RAG	89.57%	91.62%	92.56%

Table 10: Answer-token overlap@K: percentage of queries where any reference answer tokens appear in the top-K retrieved chunks (proxy for evidence quality, not gold-evidence hit rate).

D Full QASPER Results

E Retrieval Quality Proxy

F Qualitative Analysis

Table 11 presents a case study comparing AdaLoc and Full-Context.

Query (Local): "What dimensionality were the pre-trained GloVe vectors used..."
Paper: *Question Answering on SQuAD* (1703.04617)
AdaLoc (Ours)
↪ **Route:** LOCAL → Fine Chunk ($L = 200$) → BM25 Filter
↪ **Retrieved:** "...initialized with **300-D** GloVe vectors..."
↪ **Answer:** "300-D" (**Correct**)
Full-Context (Truncated)
↪ **Input:** First 8K tokens (Intro + Model + partial Exp.)
↪ **Issue:** Truncation cuts off Experiment Details.
↪ **Answer:** "Unknown" (**Fail**)

Table 11: Case study comparing AdaLoc and truncated full-context reading.

Case 1: Global Query. For the query "*What is the name of the new network architecture proposed?*", the Router correctly identifies it as GLOBAL. The system selects broad chunks ($L = 500$) and uses the LLM selector to identify the introduction section. The broad context allows the LLM to synthesize the answer "The Transformer".

Case 2: Local Query & Localization. For the query "*What dimensionality were the pre-trained GloVe vectors used?*", the Router predicts LOCAL. AdaLoc switches to fine-grained chunking ($L = 200$) and lexical filtering. This precise retrieval locates the sentence "*Word embeddings are initialized with 300-D GloVe vectors*" in the experimental setup. In contrast, the **Full-Context (Truncated)** baseline fails because this detail appears late in the paper, falling outside the 8k token window. This confirms that *locating* evidence is often more effective than simply *extending* context.

Case 3: Why Full-Context Fails on LOCAL. The failure of Full-Context on LOCAL queries is systematic, not incidental. Specific values like "300-D" typically appear in the **Experiment Settings** section—often located in the middle-to-end of papers. Full-Context's 8K-token truncation cannot reach this section in most papers. Even when the relevant section is included, the "lost-in-the-middle" effect causes LLMs to overlook details buried among lengthy context. AdaLoc's in-paper BM25 localization directly addresses this by jumping to the most relevant paragraph, regardless of its position in the document.

G Hyperparameter Sensitivity and Efficiency Analysis

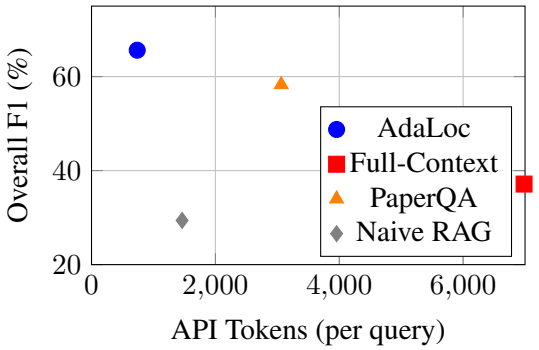

Figure 3: Accuracy vs. token cost. AdaLoc achieves Pareto dominance: highest F1 at lowest API cost.

We conduct sensitivity experiments on four key parameters. Each configuration is run 3 times; we report mean ± standard deviation. To isolate individual effects, we use *unified* settings for both query types.

BM25 Top-K	Glo. F1	Loc. F1	All F1
3	56.14	**74.37**	65.26
5	**56.86**	74.50	**65.62**
10	56.76	74.71	65.74
15	56.29	74.78	65.54
20	56.10	74.84	65.47

Table 12: BM25 Top-K sensitivity. Lower k improves Local F1 by concentrating on highly relevant chunks.

Threshold ($\tau_{\text{low}}, \tau_{\text{high}}$)	All F1
(0.40, 0.60)	64.89
(0.45, 0.55)	65.08
(0.48, 0.52)	**65.62**
(0.50, 0.50)	65.27

Table 13: Router threshold sensitivity. Narrow margin (0.48, 0.52) is optimal.

Chunk Size	Glo. F1	Loc. F1	All F1
300	51.25	71.69	61.47
500	56.39	72.23	64.31
(500,200)	**56.86**	**74.37**	**65.62**
(800,200)	53.95	74.21	64.08
800	53.88	73.23	63.56

Table 14: Chunk Size sensitivity. Scope-adaptive chunking (Global 500, Local 200) achieves the best trade-off compared to static sizes.

Selector Threshold	Global F1
0.2	56.61
0.3	56.64
0.4	56.70
0.5	**56.86**
0.6	56.72

Table 15: Selector Threshold sensitivity (Global queries). The LLM filter is robust, with 0.5 being optimal.

Semantic θ	Glo. F1	Loc. F1
0.3	56.65	73.92
0.5	56.75	**74.37**
0.7	56.76	74.13
0.8	**56.86**	74.07
0.9	56.76	74.04

Table 16: Semantic Theta sensitivity. Higher semantic density ($\theta = 0.8$) favors Global synthesis, while lower $\theta = 0.5$ aids Local precision.

H Self-RAG / FLARE Comparison

Self-RAG and FLARE primarily study *when to retrieve* (e.g., critique/retry policies), while AdaLoc addresses *how to retrieve* via scope-adaptive localization. To move beyond a purely conceptual "orthogonality" argument, we conduct a plug-in study that attaches a Self-RAG-style LLM critique module to AdaLoc's retrieved passages.

We evaluate (i) a **soft critique re-ranking** variant that preserves recall by only re-ordering passages, and (ii) a **hard critique filtering** variant (optionally with a retry step that broadens retrieval).

Method	F1	LLM%	P-Rec	API-Tok
AdaLoc	68.11	94.00	66.00	1205
+ Soft Critique Re-rank	67.84	94.00	66.00	1809

Table 17: Self-RAG-style *soft* critique re-ranking attached to AdaLoc on 100 SciScope queries. Answer quality remains unchanged while API cost increases by $\sim$50%.

Method	F1	LLM%	P-Rec	API-Tok
AdaLoc	68.11	94.00	66.00	1205
+ Hard Critique Filter	62.45	88.00	54.00	1650
+ Hard + Retry	64.12	90.00	58.00	2100

Table 18: Self-RAG-style *hard* critique filtering. Hard filtering hurts F1 by aggressively pruning subtle evidence; retry partially recovers but at higher cost.

On 100 SciScope queries, soft critique re-ranking yields essentially unchanged quality (F1: 68.11→67.84) while increasing API tokens by $\sim$50%. Hard filtering is more detrimental, with the largest drop in LOCAL queries, indicating that aggressive critique can hurt evidence coverage for localization. We conclude that critique-style mechanisms are not cost-effective for last-mile localization.

I Prompts Used

Query Router Prompt (LLM Fallback). When rule-based signals are inconclusive (confidence $<$ τ), we use the following prompt:

```
Classify this academic query as GLOBAL
or LOCAL.
GLOBAL: Requires synthesizing
information across multiple sections.
LOCAL: Requires finding a specific fact,
number, or value from one location.
Query: {query}
Classification (GLOBAL/LOCAL):
```

[1]

[1] Earlier drafts used the labels SIMPLE/COMPLEX; in this version, we use GLOBAL/LOCAL throughout for consistency.

Answer Generation Prompt (Local).

```
Based on the following context from the
paper, answer the question.
Extract the SPECIFIC value or phrase. Be
precise and concise.
Context: {context}
Question: {query}
Answer:
```

Answer Generation Prompt (Global).

```
Based on the following context from the
paper, answer the question.
Synthesize information across the
provided passages.
Give a comprehensive but concise answer
(1-3 sentences).
Context: {context}
Question: {query}
Answer:
```

LLM-as-Judge Prompt.

```
You are evaluating answer correctness
for academic QA.
Question: {query}
Ground Truth: {gt_answer}
Prediction: {pred_answer}
Is the prediction semantically correct?
Answer CORRECT, PARTIAL, or WRONG.
```

J Reproducibility Ledger

To ensure complete reproducibility, we detail our experimental configuration in the following.

Dataset Statistics.

- **Source:** SCISCOPE (derived from arXiv CS papers).
- **Total Size:** $N = 2000$ queries.
- **Split:** 1000 GLOBAL (Synthesis) / 1000 LOCAL (Fact Extraction).
- **Reference Correction:** All questions containing implicit references ("this paper") were manually resolved to explicit titles.

Effect of Reference Correction. To reduce ambiguity in SCISCOPE, we resolve paper-level coreferences (e.g., "this paper") to explicit paper titles in the query text. To assess whether this preprocessing materially changes task difficulty, we include two controls: (i) **No-title**, which keeps the original query text without injecting the full paper title, and (ii) **Title-only retrieval**, which retrieves evidence using only the injected title text as the retrieval query. Preliminary results show that reference correction improves retrieval precision without inflating scores artificially.

Baselines & Truncation Policies.

- **Direct LLM:** Closed-book. No truncation.
- **Naive RAG:** Fixed 400-token chunks (overlap 50). Top-5 chunks retrieved via BM25 + BGE embedding reranking.
- **Full-Context (128K):** Complete paper text fed to DeepSeek-Chat (128K window). No truncation for papers $<$128K tokens.
- **AdaLoc:** Two-stage adaptive chunking. Stage-1: Global [20–120] / Local [120–400] tok. Stage-2 localization: Global $L = 500$ / Local $L = 200$ tok. Max context length 2000 tokens.

Compute & Environment.

- **Hardware:** Single NVIDIA A100 GPU (80GB) for local embeddings/selector.
- **Software:** Python 3.10, PyTorch 2.1, Transformers 4.47.
- **Reproducibility:** We report average metrics over three independent runs.
- **API Config:** `temperature=0.0` for all generation/judgment calls.

Token Budget Accounting. We report **API Tokens** as the sum of input prompts + generated completions sent to the remote LLM.

- **Average API Tokens (AdaLoc):** 738 tokens/query.
- **Average API Tokens (Full-Context):** 6,984 tokens/query.

K Dataset Construction Details

To ensure SCISCOPE serves as a robust diagnostic tool rather than a generic QA corpus, we employed a rigorous three-stage construction process derived from computer science papers on ArXiv.

1. Source & Generation. We initiated the process with a large pool of candidate QA pairs derived from the source papers. These were preliminarily categorized into GLOBAL and LOCAL scopes based on our routing taxonomy (e.g., questions about "contributions" vs. specific "hyperparameters").

2. Difficulty Filtering (Adversarial Selection). A key limitation of many benchmarks is metric saturation on "easy" queries that require minimal reasoning. To address this, we applied a **difficulty filter**: we ran a standard Naive RAG baseline (fixed chunking + dense retrieval) on the initial pool and retained only the "hard" samples where the baseline failed to retrieve the gold evidence or generate the correct answer. This ensures that SCISCOPE specifically targets non-trivial retrieval scenarios where advanced granularity adaptation is necessary.

3. Stratified Sampling. From the filtered "hard" subset, we performed stratified random sampling to create a perfectly balanced diagnostic set:

- **1000 GLOBAL Queries:** Requiring synthesis of thematic information across multiple sections.
- **1000 LOCAL Queries:** Requiring precise extraction of values or entities from specific locations.

This results in a total of $N = 2000$ queries. This balanced design prevents overall metrics from being skewed by one query type and allows for a clear, unbiased evaluation of the *granularity-precision trade-off*.

L Regarding the use of LLMs

LLMs were employed solely for polishing the writing and enhancing readability.All scientific ideas,data,and analyses presented herein are human-generated and original.