RESCOPE: SPARSE PREFILL WITH SEGMENT OBSERVATIONS AND QUERY UPDATES

ABSTRACT

Long-context language models incur substantial prefill costs as dense attention scales quadratically with input length. Sparse attention reduces this cost by selecting query–key interactions, but a few representative queries can miss important interactions when tasks and evidence are distributed throughout the input. Expanding observations, meanwhile, increases selection overhead. We propose ReScope, a training-free sparse prefill method that combines segment observations with temporary query updates. Segment observations cover different input regions, while a small tail window repeatedly reads the context and updates temporary query states to identify additional candidate interactions. Keys, values, and positions remain fixed during these updates. Final sparse attention uses the original Q, K, V and retains the full KV cache. On evaluation workloads constructed from real question-answering data and natural documents, including HotpotQA, NarrativeQA, and 128K scientific-paper question answering, ReScope improves F1 by 3.0–4.9 points over the respective sparse baselines. On a controlled retrieval benchmark with 128K-token inputs, optimized ReScope achieves 100% accuracy and a $2.45\times$ complete-prefill speedup over dense attention.

1 INTRODUCTION

Long-context language models process the input during prefill before generating the first output token. Dense attention makes this stage costly because it computes a quadratic number of query–key interactions. At one million tokens, attention can account for over 90% of time to first token (Zhu et al., 2025). Sparse attention reduces this work by retaining only selected interactions.

Sparse attention has two broad approaches: training-based and training-free. Training-based methods learn sparse patterns or selectors through additional training (Gao et al., 2025). Training-free methods operate on an existing model (Jiang et al., 2024; Lai et al., 2025; Zhu et al., 2025). They must identify useful interactions cheaply enough to save time. Fig. 1a contrasts two selection strategies. Full-score Top-k evaluates all causal query–key scores before pruning them. Representative-query selection instead estimates a shared pattern from a small set of queries. This saves selection work, but the selected pattern may not represent the needs of other positions.

This limitation matters when a task appears early and relevant evidence is distributed through the input. RAG prompts can place the question before retrieved documents. Agent inputs can place the task before code, logs and tool outputs (LangChain, 2024; SWE-bench, 2025). QASPER annotations also show evidence at many document positions (Dasigi et al., 2021). We study complete-prompt prefill in these layouts. In a controlled 40-document experiment near 128K, moving the same question from after the document to before it reduces uniform vertical–slash selection from 97.5% to 25.0%, while Dense remains at 100% (Fig. 1c). Moving the question changes both its distance from the tail and the context visible at its own position. This experiment shows layout sensitivity but cannot distinguish the effects of these two changes.

These observations highlight a central challenge for sparse prefill: identifying interactions needed across a long input while keeping selection overhead small enough to preserve the benefit of sparse execution. Segment sampling broadens positional coverage (Zhu et al., 2025; Yang et al., 2025; Li et al., 2025), while the query states used at those positions determine which interactions are selected. We therefore consider both where queries are observed and how additional selection queries can be constructed.

We propose RESCOPE, a training-free sparse prefill method that combines segment observations with iterative query updates. Segments observe original queries at different input positions. A small tail window additionally reads the current layer's context, updates temporary hidden states, and constructs

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

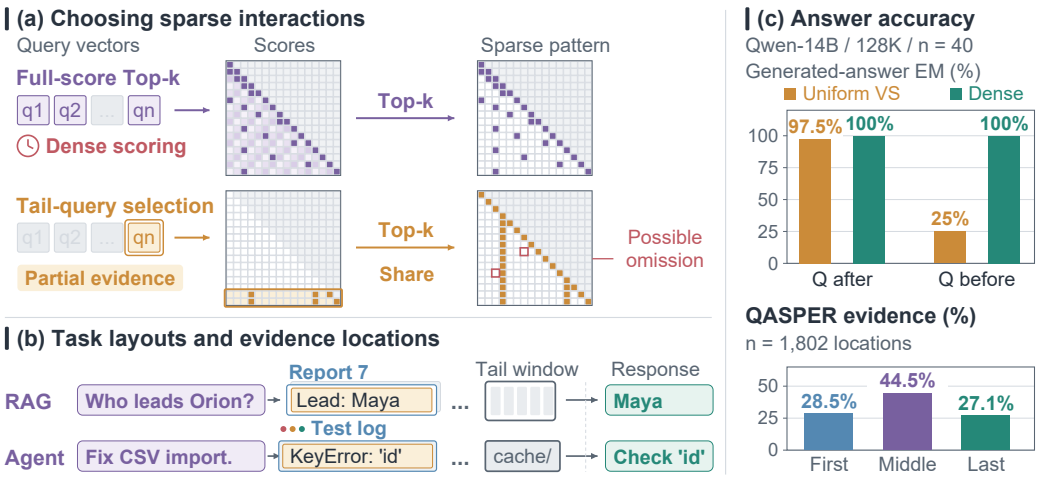


Figure 1: **Selection strategies and workload sensitivity.** (a) Full-score Top-k requires dense scoring. Representative-query selection estimates a shared pattern from a few query rows. Gray marks future keys and red boxes mark possible omissions. Shared patterns retain self positions. (b) Illustrative RAG/agent layouts and the distribution of 1,802 matched QASPER evidence-paragraph locations. Purple marks tasks and orange marks evidence. Outlines mark tail observations. Green marks responses. Generated responses follow the prompts. (c) Numeric exact-match scores on documents with Qwen2.5-14B-Instruct-1M at 128K tokens. The same question appears after or before the same document. Uniform VS denotes the uniform-budget vertical-slash control.

new selection queries. These queries can identify candidates missed by the initial observations. Updating every segment would require many context reads and projections; updating only one tail window limits this additional work. Candidates from both sources are merged for one final sparse attention pass.

We make three contributions.

- **Temporary queries for sparse selection.** ReScope combines segment observations for positional coverage with two temporary updates in a small tail window to identify additional candidates. Current-layer K, V and positions remain fixed; final attention uses the original Q, K, V and retains the full KV cache.
- **Efficient selection and execution.** Grouped score reductions, GPU index construction, and reusable buffers reduce launch, synchronization, and allocation overhead. The same generic optimizations apply to VS controls, and complete prefill timing includes query updates, selection, and sparse execution.
- **Evaluation on natural and controlled workloads.** On HotpotQA, NarrativeQA, and 128K scientific-paper question answering, optimized ReScope improves F1 by 3.0–4.9 points over the respective sparse controls. On controlled retrieval at 128K with questions first, it achieves 100% accuracy with a 2.45× complete-prefill speedup over Dense.

2 Related Work

Sparse selection and execution. Learned selectors require training or model adaptation (Gao et al., 2025; DeepSeek-AI, 2025; Zhou et al., 2026). Training-free methods estimate patterns from sampled attention (Jiang et al., 2024; Lai et al., 2025; Zhu et al., 2025), screen blocks (Xu et al., 2025; Zhang et al., 2025b), share index scans (Liu et al., 2026), or summarize omitted blocks (Fan et al., 2026). Their quality must justify both selection and execution costs (Nawrot et al., 2026). ReScope uses the existing VS representation and kernel; its addition concerns the selection queries.

Query positions and states. Segment or region sampling broadens original-query coverage (Zhu et al., 2025; Yang et al., 2025; Li et al., 2025); QuoKA selects queries that differ from the mean

(Jones et al., 2026). Lookahead methods use a smaller model, draft tokens or pseudo-responses (Liu et al., 2025; Galim et al., 2026; Wang et al., 2025). ReScope instead updates temporary states using the current layer's fixed context. It introduces neither a draft model nor future tokens. Our segment and expanded-query controls isolate components, not complete MInference or LoopServe systems. Released FlexPrefill supplies a complete external-method comparison.

Cache methods. Query-based eviction and retrieval (Li et al., 2024; Tang et al., 2024; Zhang et al., 2025a; Li et al., 2026) and sparse decoding (Gao et al., 2026; Fu et al., 2026; Lin et al., 2025) address cache access. ReScope changes prefill support and retains the full cache; it does not establish an incremental cache-update algorithm.

3 Preliminaries

Notation. For a prompt of N tokens, consider one attention head in a transformer layer with head dimension d. Let $Q, K, V \in \mathbb{R}^{N\times d}$ denote the query, key and value matrices supplied to attention, with positional transformations included in Q and K. The row q_t is the query at position t. Under causal attention, it can attend to key positions $1, \ldots, t$.

Sparse attention. Sparse attention restricts each query to a nonempty set of key positions $S_t \subseteq \{1, \ldots, t\}$ and computes

$$\widehat{o}_t = \mathrm{softmax}\left(q_t K_{S_t}^\top / \sqrt{d}\right) V_{S_t}. \tag{1}$$

Here K_{S_t} and V_{S_t} contain the selected key and value rows, and the softmax is taken over the selected keys. During prefill, attention outputs are computed for all prompt positions. The realized support fraction β measures the fraction of valid causal query–key pairs retained by the final sparse kernel.

Representative-query selection. A small set of observation queries Q_{obs} can estimate a sparse pattern shared by a larger group of query rows. In the vertical–slash (VS) pattern of MInference (Jiang et al., 2024), a vertical column selects an absolute key position and a slash selects a relative query–key offset. The estimator computes causally masked attention weights from W observation queries, sums these normalized weights by key position and relative offset, and selects the highest-scoring columns and offsets. This operation is denoted by $\mathcal{E}(Q_{\mathrm{obs}}, K; v, s)$, where v and s are the requested numbers of columns and offsets. The sparse kernel expands the selected pattern into query–key pairs subject to each row's causal mask.

Quality and cost. Support is measured after union, block expansion and causal filtering. It excludes selection work. We use matched support targets to study selection quality, and complete prefill time across budgets to study efficiency. Neither equal support nor equal query counts imply equal FLOPs.

4 Analysis of Sparse Selection

We examine query placement, the effect of temporary updates, and their computational cost.

4.1 Where to observe queries

Segment observations extend the positional coverage of tail sampling (Zhu et al., 2025; Yang et al., 2025; Li et al., 2025). On three fixed 128K Qwen development inputs, we compare one tail window with 16 segment-end windows at layers 10, 20 and 30. Each window has 64 original queries and reads its visible prefix. We measure sparse-output relative L_2 error against Dense at 256 fixed query positions per head. Separate calibration targets 1.5% support; realized means are 1.71% and 1.52%, so this comparison does not isolate positions at equal support.

Segment observations reduce error for 80 of 120 head observations at layer 20 and 106 of 120 at layer 30. At layer 10, 86 of 120 instead have higher error. The effect varies across layers and inputs.

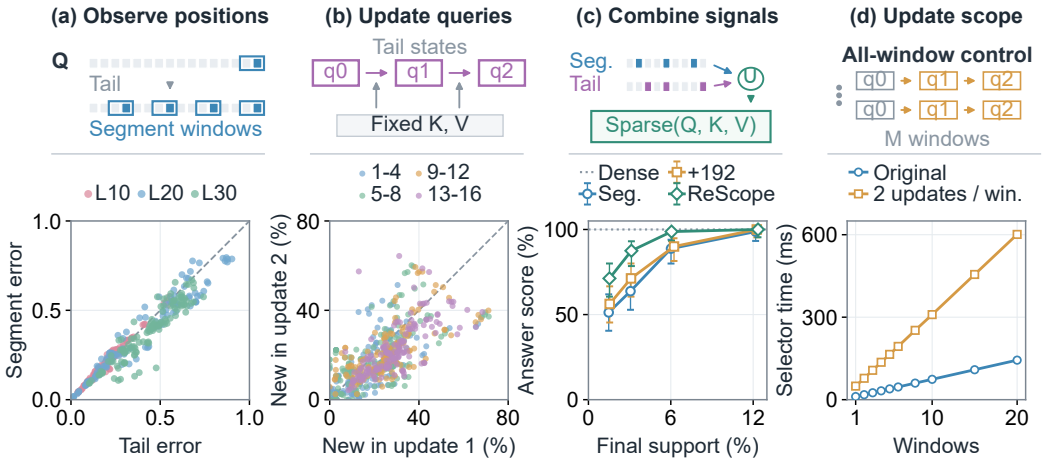


Figure 2: **Selection choices and measured effects.** Top: observations, updates, union and update scope; final attention uses original Q, K, V. Bottom, Qwen near 128K: (a) tail versus segment output error, 360 head observations from three inputs and three layers; (b) new-column fractions, 640 head–segment observations from one input, measuring turnover rather than usefulness; (c) 80-document synthetic question-first scores versus actual support, with pointwise 95% Wilson intervals; (d) selector time when updating every window, two inputs and three repeats on PRO 6000, excluding conversion and final attention. Head observations are not independent documents. Full protocols are in Appendix E.

4.2 What temporary query updates provide

Candidate changes. We update the last 64 temporary states twice while keeping the layer's K, V and positions fixed. Updated queries select columns outside the segment candidates and earlier global estimates. The median new-column fractions are 22.96% and 16.18% across the plotted head–segment pairs (Fig. 2b). Candidates are filtered to each segment's visible prefix.

Answer quality at limited support. On 80 synthetic confirmation documents at 128K, all three selectors observe 64 original queries per segment. Expanded observation adds 192 original tail queries. ReScope uses three 64-query estimates with two intervening updates. Both global branches score the same number of query rows, but only ReScope performs the extra value reads and projections. Separate development calibration targets 1.5%, 3%, 6% and 12% support. ReScope exceeds expanded observation by 15.00, 16.25 and 8.75 score points at the first three targets; both reach 100 at the last (Fig. 2c). Scores use exact matching of extracted numeric answers. These scores compare complete selectors at similar support. Separate controlled comparisons of the update branch are summarized in Appendix H.

Isolating the update branch. On two development papers, a fixed-Dense-state value intervention reduces mean local error by 0.00193 after approximate support matching; the common-scale comparison reverses direction (Appendix A.3). Separate answer-quality comparisons match selector time or realized support and intervene on candidate identity or temporary-read values (Appendix H). Updates improve F1 by at least 2 points over additional original queries at a selector-time difference of at most 5%, with a positive lower endpoint of the paired 95% interval. At matched support, replacing update-specific candidates with original-query candidates reduces F1 by at least 2 points. Real values also improve F1 by at least 2 points over permuted values under the same budget and path. These comparisons support a contribution from the updated branch beyond additional original-query scoring.

4.3 How widely to distribute updates

Cost of distributed updates. We replay the layer-20 selector over 40 heads on two 128K development inputs, using PRO 6000. Ten window counts from 1 to 20 divide fixed requests of 1,280 columns and

1,280 offsets per round; final support need not match. Original-query selection grows from 11.29 to 143.81 ms. With two updates at every window, it grows from 49.32 to 601.10 ms (Fig. 2d).

Concentrating the update computation. For M equal segments and window size W, segment-end observations form approximately

$$W \sum_{j=1}^{M} \frac{jN}{M} = \frac{WN(M+1)}{2} \tag{2}$$

score entries per head before causal masking. At one global window, the pairwise work is $O(WN)$ for a fixed number of updates and does not grow with M. Projections, ranking, and index preparation contribute further cost. ReScope distributes original-query observations across segments and restricts updates to one tail window.

5 ReScope Design and Implementation

As shown in Fig. 3, ReScope combines segment observations with temporary query updates in each prefill layer. The segment branch selects a pattern for each group of query rows. The global branch adds candidates from a small tail window whose states are repeatedly updated. The two branches share the layer's original keys and values. Their candidates are merged for one final sparse attention pass over the original Q, K, V.

5.1 Segment selection

Partition the prompt into segments I_j of at most L tokens, with endpoints $e_j = \min(jL, N)$. The final W queries of each segment estimate a pattern over its available prefix,

$$P_j^{\text{seg}} = \mathcal{E}\big(Q_{e_j-W+1:e_j}, K_{1:e_j}; v_s, s_s\big)\,. \tag{3}$$

The observation window clips to available rows for a short prefix. Each observation query retains its causal mask. Every query block in I_j shares the selected columns and relative offsets, subject to its own mask. Thus L controls the estimation scope while keys remain available throughout the visible prefix. This branch builds on existing segment and chunk observations (Zhu et al., 2025; Yang et al., 2025) and supplies the position-dependent candidates for our combined selector.

5.2 Iterative query refinement

The global branch starts from the last W original queries $\widetilde{Q}_0$ and their pre-normalization layer inputs z_0. It first estimates $P_0^g = \mathcal{E}(\widetilde{Q}_0, K; v_g, s_g)$. For $r = 1, \ldots, R-1$, it reads the context and updates the temporary states,

$$\begin{aligned} u_{r-1} &= \mathrm{DenseAttn}(\widetilde{Q}_{r-1}, K, V; M_W), \\ z_r &= z_{r-1} + W_o u_{r-1}, \qquad \widetilde{Q}_r = \mathcal{Q}_\ell(z_r; p_{\text{tail}}), \\ P_r^g &= \mathcal{E}(\widetilde{Q}_r, K; v_g, s_g). \end{aligned} \tag{4}$$

Here u concatenates the heads before the output projection W_o, and M_W preserves the window's original causal positions. The map $\mathcal{Q}_\ell$ is the layer's native query transformation: normalization, query projection (including bias), any head normalization, RoPE and model-specific position scaling, in native order. For Qwen it reduces to $\mathrm{RoPE}(W_q\,\mathrm{Norm}(z))$. Keys, values and positional indices stay fixed. Only the W temporary states are updated, with no feed-forward sublayer or training. We use $W = 64$ and $R = 3$, giving three estimates and two updates. These states guide selection during prefill and are discarded after selection.

5.3 Candidate merging and final attention

Each segment combines its pattern with all global estimates,

$$P_j = P_j^{\text{seg}} \cup \bigcup_{r=0}^{R-1} P_r^g, \qquad S_t = \mathrm{CausalExpand}(P_j, t), \quad t \in I_j. \tag{5}$$

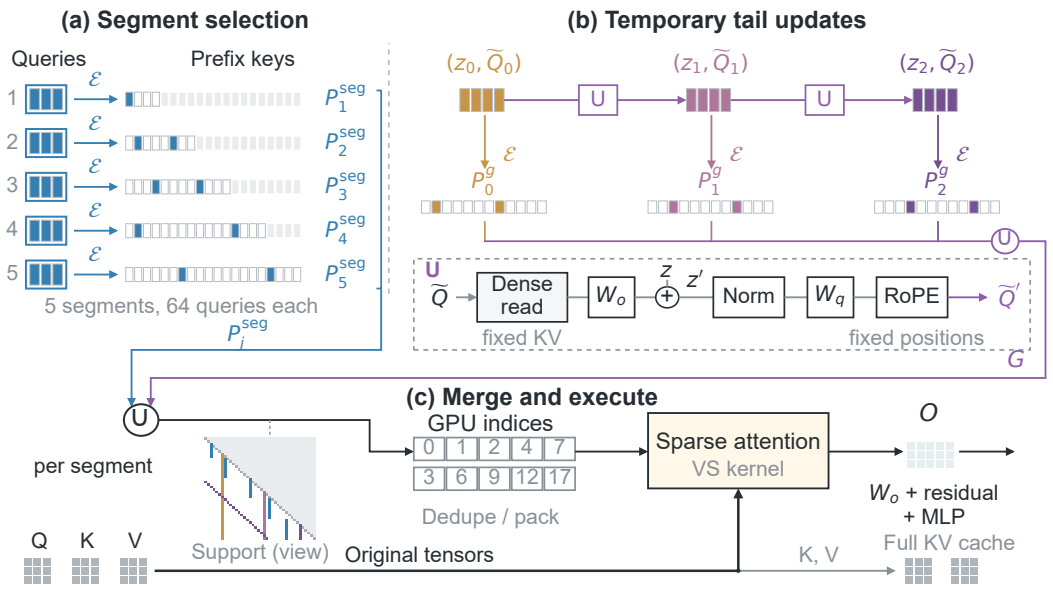


Figure 3: **ReScope in one prefill layer.** (a) Each 8K segment supplies 64 original queries over its visible prefix. (b) Two temporary tail updates produce three estimates while K, V and positions remain fixed. (c) Each segment merges its candidates with the global union. Conversion deduplicates, causally filters and packs support for final attention with original Q, K, V. Matrices and indices are schematic; no dense mask is materialized. Support includes columns and relative offsets. Here 1K is 1,024 tokens.

Vertical indices keep their absolute key positions and slash indices keep their relative offsets. The converter expands the union into kernel indices and removes duplicate support. Final attention uses the original Q, K, V in Eq. (1); its outputs enter the layer's ordinary residual and feed-forward operations. The full-length KV cache is retained in the model's standard format. The complete procedure is given in Algorithm 1.

Because overlap varies across candidates, requested counts do not add up to the final support size. We calibrate counts on separate documents using the converter's realized support, without using answer scores. Each ablated variant is calibrated separately toward the same target. Selection uses the supplied prompt as a whole, while final pairs obey the causal mask. This does not ensure prefix-causal selection at intermediate positions; Appendix C.5 describes the corresponding protocol for scoring within a prompt.

5.4 Efficient selection and index construction

We organize the implementation around the small selection windows and the merged support. Slash-score reductions are batched across eight heads, with padding created on the GPU. This reduces Python loops, small kernel launches and host-to-device transfers while preserving the scoring precision and top-k rule.

A Triton converter uses integer histograms and prefix sums to combine block deduplication, causal filtering, removal of block-covered columns, and index compaction. Each head and query block writes its final indices directly. Work buffers are allocated from shape-based capacity bounds and reused for the same shape and stream, avoiding per-segment synchronization to determine allocation sizes. This trades additional workspace for fewer allocations and synchronization points.

The resulting indices use the existing VS attention kernel (Jiang et al., 2024). The updates retain their sequential dependencies, and the implementation introduces no additional computation streams. Selection and conversion checks verify agreement with the original implementation. The same general optimizations are applied to the sparse controls in the implementation study. Complete prefill measurements include query updates, selection, index construction and final attention.

Cost and use boundary. Support does not include selection work; Appendix C.2 accounts for score entries, value reads and projections separately. Our evaluation measures complete-prompt prefill. Retaining the full KV cache does not establish efficient streaming appends or prefix-cache reuse.

6 EVALUATION

We distinguish three questions: whether query updates improve selection at limited support, whether their cost is justified across budgets, and whether the result transfers to natural documents and another model. The controlled study provides a complete quality–support–time grid. Natural studies retain their own populations and calibration protocols. The original implementation's budget grid and the optimized operating points are reported separately.

6.1 EXPERIMENTAL SETUP

Controlled retrieval. The primary study uses Qwen2.5-14B-Instruct-1M and 80 paired synthetic semantic documents at 32K and 128K. Each document contains four distinct name–number facts, with evidence depths spanning 5–95% of the body. The same question and body are used in the question-first and question-last layouts. Score is generated-answer exact match. The complete grid contains 34 method–budget configurations across the two lengths at each layout; parameters are calibrated using support counts on separate documents and frozen before evaluation (Appendix A.4).

Natural documents. The LongBench adaptations use 40 complete HotpotQA documents and 40 complete NarrativeQA documents, with no added distractors or truncation (Bai et al., 2024; Yang et al., 2018; Kočiský et al., 2018). The question is placed before or after the same document. The original study contains two models, six methods and both layouts, giving 1,920 generated responses; the main table reports Qwen, while both models appear in Appendix B.2. The QASPER evaluation uses an 80-paper cohort and a separate 40-paper cohort at approximately 128K (Dasigi et al., 2021). These inputs retain one complete target article and add training-set distractors. We report question-first inputs at a 3% support target and question-last inputs at a 6% target (Appendix A.1, Appendix A.2).

Methods and cost. Dense uses full causal attention. Segments observe 64 original queries per 8K segment. Expanded observation adds 192 original tail queries; RESCOPE uses 64 tail states and two updates. All VS methods share the representation, execution optimizations and first-256-key retention. Released FlexPrefill uses its original selector source (Lai et al., 2025). Realized support is

$$\beta = \frac{\text{executed valid causal query–key pairs}}{\text{all valid causal query–key pairs}}, \tag{6}$$

after overlap removal and block expansion. Selection work is excluded from β and included in complete prefill time. The VS controls share ReScope's kernel and generic optimizations. FlexPrefill retains its released selector and execution path; sharing a device and timing boundary does not imply identical kernels.

Complete external methods. Released FlexPrefill is the current complete external comparison. Segment and expanded-query variants are component controls.

Timing and statistical units. Complete prefill includes query updates, selection, index conversion, attention, KV construction, first logits and argmax. Loading, tokenization and subsequent decoding are excluded. Quality and timing use matching configurations, with their sample sizes reported separately. Paired statistics resample documents; heads and repeats are not independent semantic samples. Per-request calibration must be counted separately.

6.2 CONTROLLED RETRIEVAL QUALITY AND COST

In the original implementation, at matched support targets with the question first, RESCOPE improves over expanded original-query observation by 27.50 points at 32K and 16.25 points at 128K (Table 1). Both pass Holm correction across 16 predefined contrasts; their paired 95% intervals are [17.50,37.50] and [6.25,26.25]. Question-last scores are 100 for all three VS selectors.

Method	Score (%)		Support (%)		Prefill (s)	
	Before	After	Before	After	Before	After
32K, default VS support target 10%						
Dense	100.00	100.00	100.00	100.00	6.990	7.019
FlexPrefill $\gamma = 0.5$	86.25	97.50	6.29	5.93	5.119	5.084
FlexPrefill $\gamma = 0.9$	100.00	100.00	26.35	26.14	6.090	6.034
Segments	67.50	100.00	10.34	10.26	5.298	5.292
Segments + 192 queries	68.75	100.00	10.45	10.37	5.726	5.714
RESCOPE	**96.25**	100.00	10.23	10.15	6.036	6.019
128K, default VS support target 3%						
Dense	100.00	100.00	100.00	100.00	62.464	62.453
FlexPrefill $\gamma = 0.5$	65.00	98.75	3.71	3.56	21.435	21.008
FlexPrefill $\gamma = 0.9^{\dagger}$	100.00	100.00	17.91	18.57	31.829	32.106
Segments	63.75	100.00	3.06	3.04	25.309	25.263
Segments + 192 queries	71.25	100.00	3.10	3.06	27.112	27.060
RESCOPE	**87.50**	100.00	3.13	3.02	28.355	28.296
RESCOPE (6%)	**98.75**	100.00	6.06	5.89	30.528	30.447

Table 1: **Original implementation: controlled retrieval quality and complete prefill cost.** Qwen2.5-14B-Instruct-1M on 80 paired documents. Timing uses the same configurations on one A800, three separate documents per layout and two repeats. † marks supplemental timing with numerical repeat variation. The complete budget grid is in Appendix A.5.

	HotpotQA F1		NarrativeQA F1	
Method	Q before	Q after	Q before	Q after
Dense	58.3	57.4	20.3	25.5
End-probe VS	50.5	60.3	23.1	28.9
Segments	48.9	52.4	25.1	25.1
Refinement only	51.5	60.3	25.8	29.0
Local expansion	45.8	53.1	20.2	30.7
Original RESCOPE	56.4	58.7	25.6	27.6
Optimized RESCOPE	**56.4**	**63.3**	**28.8**	**33.7**

Table 2: **LongBench task adaptations with Qwen2.5-14B-Instruct-1M.** Generated-response F1 on 40 complete documents per task. Q before/after moves the same question around the same document. These controlled layout adaptations are not official LongBench leaderboard results.

Complete prefill time includes updates, selection, index conversion, sparse attention, KV construction and first logits. At 128K and 6%, RESCOPE reaches 98.75 in 30.528 s, while segments require 12% support and 32.470 s for the same score. The advantage is budget-dependent: at 32K FlexPrefill reaches 96.25 in 5.511 s, and at 128K segments at 6% score 88.75 in 27.578 s, slightly above RESCOPE at 3% in both score and speed. Tables in Appendix A.5 report every tested budget. Matching observed scores on 80 documents leaves population-level quality equivalence and uncertainty in the speedup unresolved.

Optimized complete prefill. Table 15 reports eight optimized operating points. With the question first, 32K accuracy is 96.25/100% with mean prefill times of 4.409/4.87 s; 128K accuracy is 95/100% in 19.55/25.463 s. With the question last, both lengths reach 100%, in 4.05 s at 32K and 19.30 s at 128K. These times are approximately 20% lower than the corresponding tested baselines at the same observed scores. The original budget sweep in Fig. 4 remains a separate comparison; eight operating points do not define a complete optimized frontier. The 128K FlexPrefill references retain their supplemental-timing qualification.

6.3 LONGBENCH TASK ADAPTATIONS

Optimized RESCOPE scores 56.4/63.3 F1 on HotpotQA and 28.8/33.7 on NarrativeQA (question first/last). The gains over the best listed sparse control are 4.9/3.0 and 3.0/3.0 F1, respectively; these

Method VS support target	80 papers Q before 3%	 Q after 6%	Independent 40 papers Q before 3%	 Q after 6%
Dense	40.62	51.99	28.70	41.95
FlexPrefill $\gamma = 0.5$	23.97	34.41	–	–
FlexPrefill $\gamma = 0.75$	26.00	43.56	–	–
FlexPrefill $\gamma = 0.9$	24.19	43.63	–	–
Segments	37.74	42.07	37.98	41.13
Segments + 192 queries	42.59	51.82	40.15	41.02
Original ReScope	45.57	55.93	41.40	45.53
Optimized ReScope	**45.59**	**55.93**	**43.15**	**45.53**

Table 3: **Constructed 128K QASPER evaluation.** Answer token F1 $\times 100$, averaged over papers in each cohort. Percentages specify the VS support levels. The optimized confirmation uses development-frozen settings; the historical 40-paper input-adapted protocol is reported separately. A dash denotes a configuration not evaluated in that cohort.

are descriptive mean differences. Dense retains higher question-first HotpotQA F1. The original two-model study and its paired analysis appear in Appendix B.2.

6.4 Extended QASPER evaluation

The optimized means favor ReScope over expanded original queries by 3.00/4.11 F1 on the 80-paper cohort and 3.00/4.51 on the 40-paper cohort. The paired 95% intervals for these F1 differences have positive lower endpoints in all four conditions (Appendix H). The optimized confirmation freezes calibration on development inputs and performs no additional calibration on test inputs. Actual VS support is 3% for question-first and 6% for question-last inputs, with relative support mismatch at most 1%. The historical 80-paper protocol uses two development papers; the historical 40-paper protocol calibrates each test input separately. That input-adapted study remains distinct from the new fixed configuration confirmation.

Natural-task quality and time. At comparable quality, complete prefill latency is 80–85 on a scale where the matched natural-task baseline is 100, a 15–20% reduction. These normalized measurements use the development-frozen configuration, with zero additional per-test calibration (Appendix H). They are independent of the synthetic timing measurements.

Frozen synthetic calibration does not transfer uniformly to natural inputs. For example, in the lower-budget 128K question-first transfer setting, ReScope scores 21.33 F1, versus 22.11 for expanded original queries and 25.00 for segment observations (Table 7).

6.5 Model transfer and the value of updates

The model-specific query map follows Eq. (4). Section 4 reports selector quality and fixed-state diagnostics. Additional cost-matched controls and interventions are reported in Appendix H.

7 Discussion and conclusion

ReScope combines segment observations with temporary query updates while final attention uses the original Q, K, V. Its optimized implementation reaches 96.25% accuracy at 32K in 4.409 s and 100% at 128K in 25.463 s, about 20% less prefill time than the listed baselines at the same observed scores. Natural-task confirmation improves F1 and reduces normalized complete prefill latency by 15–20% at comparable quality, with calibration frozen on development inputs and no additional per-test search. Our scope is complete-prompt prefill; streaming, cache reuse and response throughput are separate questions.

AI USE STATEMENT

Large language models were used to assist with manuscript writing, translation, and language polishing. The authors take full responsibility for the final content of this paper.

REFERENCES

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, 2024. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of NAACL-HLT*, pp. 4599–4610, 2021. URL https://aclanthology.org/2021.naacl-main.365/.

DeepSeek-AI. DeepSeek-V3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025. URL https://arxiv.org/abs/2512.02556.

Qihang Fan, Huaibo Huang, Zhiying Wu, Bingning Wang, and Ran He. FlashPrefill V2: Block-sparse prefill attention for long-context LLM serving. *arXiv preprint arXiv:2608.19758*, 2026. URL https://arxiv.org/abs/2608.19758.

Yaosheng Fu, Guangxuan Xiao, Xin Dong, Song Han, and Oreste Villa. SparDA: Sparse decoupled attention for efficient long-context LLM inference. *arXiv preprint arXiv:2606.04511*, 2026. URL https://arxiv.org/abs/2606.04511.

Kevin Galim, Ethan Ewer, Wonjun Kang, Minjae Lee, Hyung Il Koo, and Kangwook Lee. Draft-based approximate inference for LLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/28b3dc0970fa4624a63278a4268de997-Paper-Conference.pdf.

Yizhao Gao, Zhichen Zeng, Dayou Du, Shijie Cao, Peiyuan Zhou, Jiaxing Qi, Junjie Lai, Hayden Kwok-Hay So, Ting Cao, Fan Yang, and Mao Yang. SeerAttention: Self-distilled attention gating for efficient long-context prefilling. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/50e9dbc4ab68d94f15261ddc26c8ca2b-Abstract-Conference.html.

Yizhao Gao, Shuming Guo, Shijie Cao, Yuqing Xia, Yu Cheng, Lei Wang, Lingxiao Ma, Yutao Sun, Tianzhu Ye, Li Dong, Hayden Kwok-Hay So, Yu Hua, Ting Cao, Fan Yang, and Mao Yang. Sparse attention adaptation for long reasoning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/b56d827a2b8433517e722e0272c7f464-Abstract-Conference.html.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *Advances in Neural Information Processing Systems*, volume 37, pp. 52481–52515, 2024. doi: 10.52202/079017-1663. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5dfbe6f5671e82c76841ba687a8a9ecb-Abstract-Conference.html.

Dalton Jones, Junyoung Park, Matthew Morse, Mingu Lee, Chris Lott, and M. Harper Langston. QuoKA: Query-oriented KV selection for efficient LLM prefill. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/760fe89b5a5e5792b33f23a2db1b9fbf-Paper-Conference.pdf.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018. doi: 10.1162/tacl_a_00023. URL https://aclanthology.org/Q18-1023/.

Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. FlexPrefill: A context-aware sparse attention mechanism for efficient long-sequence inference. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/03645743ea35690f30d795d6bac149a5-Abstract-Conference.html.

LangChain. Build a local RAG application: v0.2 tutorial and rlm/rag-prompt, 2024. URL https://github.com/langchain-ai/langchain/blob/v0.2/docs/docs/tutorials/local_rag.ipynb. Prompt manifest revision 50442af133e6; accessed September 20, 2026.

Haoyang Li, Zhanchao Xu, Yiming Li, Xuejia Chen, Darian Li, Anxin Tian, Qingfa Xiao, Cheng Deng, Jun Wang, Qing Li, Lei Chen, and Mingxuan Yuan. LoopServe: An adaptive dual-phase LLM inference acceleration system for multi-turn dialogues. *arXiv preprint arXiv:2507.13681*, 2025. URL https://arxiv.org/abs/2507.13681.

Wenhao Li, Jinhao Dong, Hailin Zhang, Wenhang Shi, Wei Lu, and Xiaoyong Du. RaBitQ-Cache: Rotated binary quantization for KVCache in long context LLM inference. *arXiv preprint arXiv:2606.31519*, 2026. URL https://arxiv.org/abs/2606.31519.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/28ab418242603e0f7323e54185d19bde-Abstract-Conference.html.

Chaofan Lin, Jiaming Tang, Shuo Yang, Hanshuo Wang, Tian Tang, Boyu Tian, Ion Stoica, Song Han, and Mingyu Gao. Twilight: Adaptive attention sparsity with hierarchical top-p pruning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3986. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ad2f57116d62a9d8dcfaba6a168715e4-Abstract-Conference.html.

Hong Liu, Yuan Cheng, Lin Niu, Yi Su, Yufei Xue, Anmin Liu, Guanghua Yu, and Jianchen Zhu. PIVOT: Efficient query-group indexing for token-level sparse attention. *arXiv preprint arXiv:2607.24593*, 2026. URL https://arxiv.org/abs/2607.24593.

Jingyu Liu, Beidi Chen, and Ce Zhang. Speculative prefill: Turbocharging TTFT with lightweight and training-free token importance estimation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 38188–38209. PMLR, 2025. URL https://proceedings.mlr.press/v267/liu25g.html.

Piotr Nawrot, Robert Li, Renjie Huang, Sebastian Ruder, Kelly Marchisio, and Edoardo M. Ponti. The sparse frontier: Sparse attention trade-offs in transformer LLMs. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 38667–38701, 2026. doi: 10.18653/v1/2026.findings-acl.1926. URL https://aclanthology.org/2026.findings-acl.1926.pdf.

SWE-bench. SWE-agent with Claude 4 Sonnet: Archived verified evaluation trajectories, 2025. URL https://github.com/SWE-bench/experiments/tree/main/evaluation/verified/20250522_sweagent_claude-4-sonnet-20250514. SWE-agent version 1.1.0; archived per-call model inputs; accessed September 20, 2026.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. QUEST: Query-aware sparsity for efficient long-context LLM inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 47901–47911. PMLR, 2024. URL https://proceedings.mlr.press/v235/tang24l.html.

Yixuan Wang, Shiyu Ji, Yijun Liu, Yuzhuang Xu, Yang Xu, Qingfu Zhu, and Wanxiang Che. Lookahead Q-cache: Achieving more consistent KV cache eviction via pseudo query. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 34158–34174. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1732. URL https://aclanthology.org/2025.emnlp-main.1732/.

Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. XAttention: Block sparse attention with antidiagonal scoring. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 69819–69831. PMLR, 2025. URL https://proceedings.mlr.press/v267/xu25ag.html.

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, Weijia Xu, Wenbiao Yin, Wenyuan Yu, Xiafei Qiu, Xingzhang Ren, Xinlong Yang, Yong Li, Zhiying Xu, and Zipeng Zhang. Qwen2.5-1M technical report. *arXiv preprint arXiv:2501.15383*, 2025. URL https://arxiv.org/abs/2501.15383.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Hailin Zhang, Xiaodong Ji, Yilin Chen, Fangcheng Fu, Xupeng Miao, Xiaonan Nie, Weipeng Chen, and Bin Cui. PQCache: Product quantization-based KVCache for long context LLM inference. *Proceedings of the ACM on Management of Data*, 3(3):201:1–201:30, 2025a. doi: 10.1145/3725338. URL https://doi.org/10.1145/3725338.

Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. SpargeAttention: Accurate and training-free sparse attention accelerating any model inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76397–76413. PMLR, 2025b. URL https://proceedings.mlr.press/v267/zhang25ch.html.

Yanke Zhou, Yiduo Li, Hanlin Tang, Maohua Li, Kan Liu, Tao Lan, Lin Qu, Yuan Yao, and Xiaoxing Ma. Full attention strikes back: Transferring full attention into sparse within hundred training steps. *arXiv preprint arXiv:2605.16928*, 2026. URL https://arxiv.org/abs/2605.16928v2.

Qianchao Zhu, Jiangfei Duan, Chang Chen, Xiuhong Li, Siran Liu, Guanyu Feng, Xin Lv, Chuanfu Xiao, Dahua Lin, and Chao Yang. SampleAttention: Near-lossless acceleration of long context LLM inference with adaptive structured sparse attention. In *Proceedings of Machine Learning and Systems*, volume 7, 2025. URL https://proceedings.mlsys.org/paper_files/paper/2025/file/2d04d97593c8c33d415337f408ed0e1b-Paper-Conference.pdf.

A CONTROLLED VALIDATION AND NATURAL-QUESTION PROTOCOLS

This appendix keeps each evidence source separate: natural-question cohort means, a later input-calibrated study, controlled synthetic budget scans, fixed-state diagnostics, and model-transfer variants. The latter's 80 synthetic documents are independent of the earlier 40-document study in Appendix B and the earlier 80-document study in Appendix B.3. Records, configurations and independent sample counts are kept separate across these studies.

A.1 80-PAPER NATURAL-QUESTION PROTOCOL

The cohort contains one question from each of 80 QASPER papers. Paper and question order follow a fixed seed and hash ranking. Target papers are disjoint from development papers. Each approximately 128K input retains the complete target article and adds training-set distractors; only the final distractor may be truncated. A fixed eligibility filter excludes questions requiring unavailable figure or table content. The two layouts share the same question and document body. These constructed inputs are not a native-128K QASPER benchmark or production traces.

Qwen2.5-14B-Instruct-1M runs in bf16. Segments contain 8,192 tokens and observe their final 64 queries over the visible prefix. Expanded observation adds 192 original tail queries; RESCOPE makes three estimates from 64 tail states with two updates. All three VS methods share the optimized execution path and retain the first 256 key columns, counting them in support. The 80-paper protocol calibrates selectors using support counts on two development papers before applying them to the target inputs. Calibration does not use answer F1. FlexPrefill uses block size 128, minimum budget 1,024, $\tau = 0.1$, and $\gamma \in \{0.5, 0.75, 0.9\}$.

Greedy generation receives no gold answers and stops at native EOS or 256 new tokens. Official QASPER answer token F1 takes the best human-reference match, is multiplied by 100 and is averaged over papers. Abstentions, incorrect answers and capped outputs remain in the denominator. Table 3 reports question-first inputs at a 3% support budget and question-last inputs at a 6% budget. Results are descriptive mean F1 comparisons. Paired uncertainty and natural-task latency are not inferred from aggregate answer scores.

A.2 INDEPENDENT 40-PAPER NATURAL-QUESTION PROTOCOL

The second cohort contains 40 additional QASPER papers, disjoint from the 80-paper cohort and development papers. It uses the same model, context length, question layouts, generation settings and scoring rule. The protocol selects each sparse input's budget parameter from support counts, without using answer quality, rather than reusing one development-calibrated parameter across all inputs. Target support and realized support are distinct quantities; per-input calibration does not imply exact equality between methods. The two cohorts are reported separately in Table 3. This cohort evaluates answer quality under per-input support calibration. It does not establish deployment efficiency or the behavior of one fixed development-calibrated configuration on those 40 inputs.

A.3 FIXED-STATE VALUE-ASSOCIATION DIAGNOSTIC

We use two separate development papers at about 128K, both question layouts, and zero-indexed layers 10, 20 and 30 of Qwen2.5-14B-Instruct-1M. Dense upstream computation supplies 12 fixed layer snapshots. Each snapshot is tested at 3% and 6% support, giving 24 pairs and 960 repeated head observations from two papers. These observations do not provide 960 independent samples.

The real and permuted conditions share the initial states and original Q, K, V. We permute only values preceding the final 64 tokens during the two temporary reads, preserving tail values, keys and positional indices. Both conditions perform the same gather operation. Final diagnostic attention uses the original Q, K, V, and its output is not committed to the Dense backbone. Source hashes, exact real-selector parity and a real–permuted–real replay check verify the recorded execution. Full Dense-backbone logits also match the uninstrumented reference.

Comparison	Real e	Permuted e	Δe	Real lower
Support-target search	0.178268	0.180198	-0.001930	19/24
Common scale	0.178268	0.174528	$+0.003740$	1/24

Table 4: Fixed-state value-association diagnostic. Errors average all 960 head observations over both targets. $\Delta e = e_{\text{real}} - e_{\text{permuted}}$; negative values favor real associations. The last column counts pairs with a lower mean across heads, not independent documents. The target search retains its unmatched pair. The common-scale replay uses each real condition's scale for both value conditions.

For head h, we measure

$$e_h = \sqrt{\frac{\sum_{i,d}(O^{\text{sparse}}_{h,i,d} - O^{\text{dense}}_{h,i,d})^2}{\max\{\sum_{i,d}(O^{\text{dense}}_{h,i,d})^2, 10^{-30}\}}}.$$

Here i covers all original query rows, unlike the sampled-row comparison in Fig. 2(a). The support fraction is the sum of converted valid causal pairs across heads divided by $HN(N+1)/2$, including the shared 256-column prefix and removing overlap. Each condition independently scales its requested quotas using support only. The first result within 1% relative error of its target is kept; otherwise the closest of at most eight trials is retained. Forty-seven of 48 conditions meet this tolerance; both conditions meet it in 23 pairs. All conditions remain in the results. The largest paired layer-support gap is 0.0848 percentage points; support is not matched per head.

Normal associations reduce mean local error after approximate support matching. At the common scale, the direction reverses and the permuted condition has more support in 23 of 24 pairs, with a mean absolute gap of 0.3511 percentage points. Thus value associations affect candidates and local outputs, but these results do not establish a benefit independent of support. Scaling also changes segment and global quotas; we have not isolated the marginal value of update-exclusive candidates. The permutation changes value–position associations while retaining contextual information. This development diagnostic provides neither a task-quality result nor a latency benchmark.

A.4 Synthetic inputs, calibration and generation

The primary model is Qwen2.5-14B-Instruct-1M in bf16. Each synthetic semantic document contains four distinct name–number facts and a question about one randomly selected target. Five-digit answers and fact identities vary across documents. Fact depths span 5–95% of the body, following a common 1,024-token prefix. The question-first and question-last conditions move the same question block around the same body. The two lengths reuse the same 80 fact/question identities. Generation receives only the prompt; the appended reference used in input construction is removed from the tokenized sequence before the model call. The resulting prompts contain approximately 32K or 128K tokens.

We perform one prefill and generate greedily from its complete KV cache, stopping at the model's EOS or 256 new tokens. The score extractor finds five-digit numbers, including comma-formatted numbers, in the generated response. A response is correct only if there is exactly one distinct extracted number and it matches the reference. Multiple occurrences of the same number are allowed. Reference values affect neither selection nor stopping. Outputs reaching the token cap remain in the denominator.

The complete grid contains 34 length–method–budget configurations. At each length it includes Dense, four FlexPrefill settings, and four budgets for each of three vertical–slash selectors. Question placement gives 68 conditions, each with the same 80 confirmation documents. Twelve development documents are separate from these test identities. Per-configuration calibration uses two calibration documents and both layouts, seeking a mean support within 5% relative tolerance of its target in at most eight trials. Only support is used to adjust requested counts. Each configuration is frozen before its corresponding quality runs. The full grid is fixed in advance. Intermediate complete cohorts of 20 and 40 documents are retained in the audit; the manuscript uses the final complete 80-document cohort for every synthetic confirmation cell.

Segment length is 8,192 tokens and each segment observes 64 original queries. Expanded original queries retain segment observation and add 192 original tail queries. ReScope instead makes three

estimates from 64 tail queries, with two temporary updates. The same generic batched score reduction and GPU index construction apply to all three arms. The controls are component comparisons, not reproductions of LoopServe or the complete MInference system. FlexPrefill uses unchanged released selector source with block size 128, minimum budget 512, $\tau = 0.1$ and $\gamma \in \{0.25, 0.5, 0.75, 0.9\}$. The minimum budget differs from the 1,024 value used in its original paper (Lai et al., 2025). The main text reports the default quality–support–time settings in Table 1; the tables below retain the complete grid.

A.5 Complete results and paired comparisons

Tables below retain every predefined setting. Support counts are the realized valid pairs executed by final sparse attention, including required initial and nearby positions, block expansion and overlap removal. They exclude the additional score matrices and value reads used by the selector. The question-first and question-last columns use the same documents; neither heads nor layouts increase the independent sample size.

For paired uncertainty we bootstrap documents, keeping their two method outcomes together. The reported 95% intervals are unadjusted descriptive intervals. Exact McNemar tests compare the two discordant outcome counts. Holm correction uses all 16 question-first comparisons of ReScope against the two controls across the eight length–budget combinations. At 32K and 10% target, the gains over segment observations and expanded original queries are 28.75 points [18.75,38.75] and 27.50 points [17.50,37.50], with adjusted $p = 0.0000217$ and 0.0000387. At 128K and 3% target they are 23.75 points [12.50,35.00] and 16.25 points [6.25,26.25], with adjusted $p = 0.00373$ and 0.04868. Pointwise Wilson intervals in the figures describe each score separately and are not intervals for paired differences.

Method / support target	Score (%)		Support (%)		Prefill (s)	
	Before	After	Before	After	Before	After
Dense	100.00	100.00	100.00	100.00	6.990	7.019
FlexPrefill $\gamma = 0.25$	81.25	82.50	3.42	3.27	4.995	5.017
FlexPrefill $\gamma = 0.5$	86.25	97.50	6.29	5.93	5.119	5.084
FlexPrefill $\gamma = 0.75$	96.25	100.00	13.85	13.34	5.511	5.471
FlexPrefill $\gamma = 0.9$	100.00	100.00	26.35	26.14	6.090	6.034
Segments / 5%	46.25	100.00	4.94	4.87	5.072	5.065
Segments / 10%	67.50	100.00	10.34	10.26	5.298	5.292
Segments / 20%	98.75	100.00	20.79	20.42	5.834	5.797
Segments / 40%	100.00	100.00	41.44	41.01	6.777	6.767
Segments + 192 queries / 5%	56.25	100.00	5.06	4.95	5.477	5.477
Segments + 192 queries / 10%	68.75	100.00	10.45	10.37	5.726	5.714
Segments + 192 queries / 20%	100.00	100.00	20.12	19.77	6.220	6.195
Segments + 192 queries / 40%	100.00	100.00	41.71	41.27	7.277	7.276
ReScope / 5%	86.25	100.00	4.99	4.86	5.766	5.751
ReScope / 10%	96.25	100.00	10.23	10.15	6.036	6.019
ReScope / 20%	100.00	100.00	20.27	19.96	6.592	6.571
ReScope / 40%	100.00	100.00	42.07	41.55	7.719	7.690

Table 5: All predefined 32K configurations for the controlled synthetic study. Definitions and samples follow Table 1. A slash gives the requested VS support target; the measured support is reported separately. † identifies supplemental timing after the unchanged official FlexPrefill implementation failed strict numerical repeat checks. Every repeat is retained.

Method / support target	Score (%)		Support (%)		Prefill (s)	
	Before	After	Before	After	Before	After
Dense	100.00	100.00	100.00	100.00	62.464	62.453
FlexPrefill $\gamma = 0.25$	45.00	88.75	1.55	1.46	19.651	19.558
FlexPrefill $\gamma = 0.5$	65.00	98.75	3.71	3.56	21.435	21.008
FlexPrefill $\gamma = 0.75^{\dagger}$	95.00	98.75	8.04	8.01	24.435	24.308
FlexPrefill $\gamma = 0.9^{\dagger}$	100.00	100.00	17.91	18.57	31.829	32.106
Segments / 1.5%	51.25	100.00	1.48	1.46	24.142	24.119
Segments / 3%	63.75	100.00	3.06	3.04	25.309	25.263
Segments / 6%	88.75	100.00	6.02	6.05	27.578	27.506
Segments / 12%	98.75	100.00	12.18	12.25	32.470	32.326
Segments + 192 queries / 1.5%	56.25	100.00	1.55	1.51	25.934	25.854
Segments + 192 queries / 3%	71.25	100.00	3.10	3.06	27.112	27.060
Segments + 192 queries / 6%	90.00	100.00	6.23	6.21	29.643	29.730
Segments + 192 queries / 12%	100.00	100.00	12.26	12.22	34.222	34.173
ReScope / 1.5%	71.25	100.00	1.53	1.49	26.922	26.926
ReScope / 3%	87.50	100.00	3.13	3.02	28.355	28.296
ReScope / 6%	98.75	100.00	6.06	5.89	30.528	30.447
ReScope / 12%	100.00	100.00	12.35	12.19	35.880	35.871

Table 6: All predefined 128K configurations for the controlled synthetic study. Definitions and samples follow Table 1. A slash gives the requested VS support target; the measured support is reported separately. † identifies supplemental timing after the unchanged official FlexPrefill implementation failed strict numerical repeat checks. Every repeat is retained.

A.6 TIMING AND NUMERICAL CHECKS

Formal timings use one physical NVIDIA A800 with no overlapping experiment on that device. Each configuration is measured on three separate development documents in both layouts with two post-warmup repeats, giving 12 recorded prefill times. The complete set of 34 configurations has 408 such measurements. The main table reports each layout separately; the quality–latency figure pairs question-first scores with question-first latency. The current plot shows recorded means; it does not attach a timing confidence interval. Its quality sample and timing sample are distinct.

The synchronized timing interval includes the model's complete prefill, query refinement, selection, index conversion, support-count instrumentation, cache construction, first logits and argmax. Loading, tokenization, warmup and CPU record/hash export are outside the interval. Background GPU occupancy sampling remains enabled. These measurements are prefill latency, not full-response latency. Quality runs assign 76 primary documents to A800 and four to H100, keeping each document on one hardware family throughout its conditions. This assignment supports paired comparisons but does not estimate a causal hardware effect.

The initial timing gate checks execution bindings, support counts and numerical repeat behavior. Thirty-two A800 configurations satisfy its strict requirements. The unchanged official FlexPrefill implementation at 128K with $\gamma = 0.75$ and 0.9 shows repeat variation and is kept in a separately audited timing supplement. The audit checks the frozen configuration, source identity, execution and first-token behavior while retaining the numerical differences and original strict-check failures. These two points are marked with daggers in tables and open markers in the figure. They do not establish full-generation equivalence. RTX PRO 6000 timings from separate studies are not combined with this primary quality curve because cross-device numerical equivalence is not established for that pairing.

A.7 NATURAL QUESTIONS AND THE SECOND MODEL

The QASPER transfer study selects 40 existing target papers, one question per paper, from the deterministic long-input extension. Questions with malformed or missing references or requiring floating table/figure content are filtered by the input builder. All remaining answer types are retained, including extractive, abstractive, yes/no and unanswerable questions. The target article remains complete. Distractor articles come from the training split, excluding calibration articles; the final distractor may be truncated to fill the input. Target insertion is hash-based and independent of answer and evidence annotations. The 32K and 128K bodies are constructed separately from the same target paper. These inputs are not recorded production RAG traces.

Official answer F1 lowercases and removes punctuation, articles and extra whitespace, measures token overlap, takes the best match over human references, and averages over papers. We report 100 times this mean and do not report evidence F1. The 36 conditions share the same 40 papers, yielding 1,440 generated answers. Thirty-two papers use A800 and eight use H100, with each paper fixed to one hardware family. Parameters transfer from synthetic calibration. Their targets do not ensure matching realized support on QASPER; both targets per length and measured support appear below. No synthetic timing is attached to these natural-task scores.

The second-model study uses Ministral-3-14B-Instruct-2512-BF16 with its own tokenizer and chat template. Twenty new fact/question identities are distinct from the primary study and from its two calibration identities. The protocol was chosen after some primary-model results were visible and frozen before second-model generation. The two lengths and layouts reuse these 20 identities, yielding 480 answers across six methods per length. Five outputs reach the 256-token cap and remain in the scores. This is a descriptive transfer test with no formal second-model latency comparison.

The following synthetic and natural-task tables report historical implementation results. Their populations and configurations remain separate from the current method's other evaluations.

Length	Method	Setting	Answer F1 (%)		Support (%)	
			Q before	Q after	Q before	Q after
32K	Dense	Full	44.39	45.38	100.00	100.00
32K	FlexPrefill	$\gamma = 0.5$	17.50	26.17	4.39	4.31
32K	FlexPrefill	$\gamma = 0.9$	23.61	42.52	18.10	19.43
32K	Segment observations	$\beta_t = 0.1$	30.43	51.03	7.87	8.11
32K	Segment observations	$\beta_t = 0.4$	38.83	53.73	31.23	33.17
32K	Expanded original queries	$\beta_t = 0.1$	36.49	51.80	8.21	8.59
32K	Expanded original queries	$\beta_t = 0.4$	47.23	48.00	32.99	35.08
32K	**ReScope**	$\beta_t = 0.1$	40.38	50.23	8.33	9.14
32K	**ReScope**	$\beta_t = 0.4$	44.40	47.92	34.16	37.38
128K	Dense	Full	29.38	45.71	100.00	100.00
128K	FlexPrefill	$\gamma = 0.5$	16.51	29.33	2.59	2.82
128K	FlexPrefill	$\gamma = 0.9$	20.19	38.70	12.99	14.57
128K	Segment observations	$\beta_t = 0.03$	25.00	37.20	2.42	2.48
128K	Segment observations	$\beta_t = 0.12$	31.49	36.24	8.06	8.19
128K	Expanded original queries	$\beta_t = 0.03$	22.11	37.03	2.70	2.76
128K	Expanded original queries	$\beta_t = 0.12$	31.04	43.26	9.30	9.71
128K	**ReScope**	$\beta_t = 0.03$	21.33	40.72	2.73	2.94
128K	**ReScope**	$\beta_t = 0.12$	29.22	43.38	9.63	10.66

Table 7: QASPER transfer evaluation with Qwen2.5-14B-Instruct-1M on 40 test papers. The same papers are paired across lengths, question layouts and methods. We report the official answer F1 and measured final causal attention support. Long inputs contain distractor material; they are constructed workloads. Parameters are frozen from synthetic calibration, and β_t denotes that calibration target rather than a matched support on QASPER. Quality was evaluated on A800 and H100, with each paper kept on one hardware family across all conditions. No synthetic-task timing is attached to these scores.

Length	Method	Q before	Q after	Support before	Support after
32K	Dense	85.00	100.00	100.00	100.00
32K	FlexPrefill 0.5	30.00	100.00	9.21	9.10
32K	FlexPrefill 0.9	95.00	100.00	33.90	34.75
32K	Segment observations	45.00	100.00	10.14	9.89
32K	Expanded original queries	40.00	100.00	9.76	9.53
32K	ReScope	60.00	100.00	9.63	9.26
128K	Dense	100.00	100.00	100.00	100.00
128K	FlexPrefill 0.5	50.00	100.00	4.79	5.04
128K	FlexPrefill 0.9	85.00	100.00	20.59	21.22
128K	Segment observations	25.00	100.00	3.09	3.05
128K	Expanded original queries	30.00	100.00	2.91	2.85
128K	ReScope	55.00	100.00	2.97	2.89

Table 8: Historical implementation results for Ministral-3-14B-Instruct-2512-BF16 on 20 independent synthetic semantic documents, paired across lengths, layouts and all six conditions. The score uses the frozen reference-independent five-digit extraction rule. Answer scores and realized final support are percentages. Segment observations, expanded original queries and ReScope were calibrated toward 10% final causal support at 32K and 3% at 128K using two separate documents; realized confirmation support is reported in the table. The protocol was frozen after primary Qwen results were observed and before these new model outputs. Native Dense and baseline/optimized execution checks passed. All generations ran on the same H100. This is a quality transfer study; operational generation durations are not formal latency measurements.

A.8 Exploratory extension and value intervention

Exploratory natural-task extension. A later study used 40 additional QASPER papers, disjoint from the 100 previously selected targets. Both models reuse these 40 papers. The configurations and comparison families were frozen before generation. A common control, denoted P, retains the first

	32K		128K	
Method	Q before	Q after	Q before	Q after
Dense	37.45	33.35	22.31	31.04
FlexPrefill $\gamma = 0.5$	12.68	13.85	10.19	18.27
FlexPrefill $\gamma = 0.9$	16.33	27.77	13.38	26.01
Segment observations	24.32	41.02	21.00	31.43
Segments + 192 original queries	25.94	39.57	19.98	33.63
ReScope	29.14	40.01	16.81	39.07
Segment observations + P	31.82	40.03	21.19	37.96
Segments + 192 original queries + P	43.72	39.03	28.70	34.73
ReScope + P	42.64	38.59	32.41	34.78
ReScope + P, shuffled prefix values	45.65	37.48	31.02	36.74

Table 9: **Exploratory Qwen results with an unresolved execution-integrity limitation.** Official QASPER answer F1 on 40 new papers, disjoint from all 100 previously selected targets. Each paper is paired across all methods, lengths and question layouts. P retains the first 256 token columns for every VS method using it and is counted in final support. It is a generic control, not a ReScope contribution. VS settings are calibrated toward 10% at 32K and 3% at 128K on two old QASPER papers and frozen before these outputs. One worker later suffered a Python runtime failure; earlier numerical impact remains unresolved. All completed outputs are retained, so this table is exploratory and does not replace the separate ten-hour natural-task table. No latency from another workload or hardware is attached.

Method	Answer F1	Support (%)	Unanswerable
Dense	17.71	100.000	30/40
Segment observations + P	7.50	2.922	40/40
Segments + 192 original queries + P	7.50	2.933	40/40
ReScope + P	7.50	2.917	40/40
ReScope + P, shuffled prefix values	11.43	2.912	38/40

Table 10: Historical secondary natural-task results for Ministral-3-14B-Instruct-2512-BF16 at approximately 128K, with the question first. The same 40 papers as Table 9 are rendered with the model's native tokenizer and chat template; they are not 40 additional independent papers. All five methods for a paper use one assigned GPU; H100 and PRO6000 each handle 20 papers. Both workers complete normally. These descriptive scores use the same historical implementation as the earlier synthetic study. We report no cross-hardware latency claim.

256 token columns for all three vertical–slash selectors and counts them in final support. This generic prefix rule is separate from ReScope's proposed query update.

One Qwen worker subsequently failed with a Python runtime error. All completed outputs remain in the tables, including one record without a post-generation occupancy sample. The failure's cause and numerical impact are unresolved. Earlier generations and the identity gate were not all independently replayed. Four independent FlexPrefill replays reproduced two token sequences and differed in two; original scores were retained. CPU source, input, decoding and scoring checks do not certify full model execution. An independent timing worker reproduced the frozen VS configurations' calibration support counts, which verifies that narrower property.

We therefore report this campaign separately as exploratory evidence. The prefix condition improves some Qwen comparisons, but ReScope does not consistently exceed expanded original queries. On the same papers, Ministral's three unshuffled sparse conditions all produce Unanswerable and score 7.50 F1, versus Dense at 17.71. No recorded value-permutation comparison establishes a consistent benefit from intact prefix values. These results do not support a claim of broad natural-task transfer.

Predefined paired comparisons. The recorded comparisons use paper-level paired bootstrap intervals and two-sided sign-flip tests with 50,000 draws, with Holm correction within each family fixed before generation. These statistics describe the recorded outputs and do not resolve the Qwen worker failure. Intervals do not include runtime-induced or repeated-generation uncertainty.

With common prefix retention, ReScope exceeds segment observations by $+10.82$ points (95% paired interval $[+3.17, +20.10]$, Holm $p = 0.0186$) at 32K and $+11.22$ points (95% paired interval $[+4.88, +18.77]$, Holm $p = 0.0014$) at 128K, both with the question first. Against expanded original queries with the same prefix rule, the differences are -1.08 points (95% paired interval $[-5.72, +3.48]$, Holm $p = 0.6608$) and $+3.71$ points (95% paired interval $[-0.52, +8.62]$, Holm $p = 0.2553$). Thus, the comparison against additional original queries does not show a consistent gain. None of the four original-configuration comparisons passes its Holm correction.

On the new question-first inputs, actual support for ReScope+P, segment observations+P and expanded original queries+P is 9.39/10.19/10.29% at 32K and 3.02/3.04/2.99% at 128K. Shared calibration targets do not imply exactly equal support on new inputs or equal selection work.

Value intervention and scope. The four Qwen comparisons between real and permuted prefix values also do not pass their within-family Holm correction (minimum adjusted $p = 0.0933$). Ministral has identical scores for ReScope and both unshuffled sparse controls, with 40 paired ties in each primary comparison. Its real-minus-shuffled difference is -3.93 points (95% paired interval $[-10.36, +0.00]$, adjusted $p = 0.5003$). Shuffling acts only on the fully visible prefix values used during the current layer's temporary selection. It does not directly change that run's final 64 values or initial temporary state; upstream layers can produce different entering states across runs. The test therefore does not remove all value information, and a null result does not establish that value content is irrelevant. The Qwen runtime incident belongs to this later, separately launched campaign. It does not alter the earlier ten-hour cohort, frozen sources, or scores.

B HISTORICAL EVALUATION

These studies preceded the current controlled confirmation in Appendix A. They characterize earlier configurations; they are not additional independent replications of each later claim. Their populations, protocols and timing sessions are kept separate from that study and from each other.

B.1 FORTY-PAIR RETRIEVAL CONFIRMATION

Qwen2.5-14B-Instruct-1M was evaluated in bf16 on 40 paired synthetic numeric-retrieval documents at approximately 128K tokens. The same question appears before or after each fixed document. Prompt-only prefill constructs an actual KV cache, followed by dense greedy decoding to EOS or 256 tokens; gold answers affect neither selection nor stopping. The score is unique-number exact match (EM). Two disjoint calibration pairs set a mean support target of $\beta = 0.015$ within 5% relative tolerance before test predictions. Support β follows the final-attention definition in Appendix A.5; selection work is included in timing.

	EM (%)		Support β		Time	Session
Method	First	Last	First	Last	(s)	
Dense	100.0	100.0	1.0000	1.0000	35.59	A
Uniform VS	25.0	97.5	0.0154	0.0156	15.23	A
Released scaling	65.0	100.0	0.0158	0.0142	17.84	A
Local expansion	75.0	100.0	0.0162	0.0157	15.13	A
RESCOPE	77.5	100.0	0.0155	0.0151	23.63	A
Segment only	67.5	97.5	0.0151	0.0149	21.19	B
Refinement only	45.0	100.0	0.0154	0.0151	18.09	B
FlexPrefill (calibrated)	35.0	90.0	0.0155	0.0147	13.23	C
FlexPrefill ($\gamma = 0.9$)	100.0	100.0	0.1779	0.1832	18.13	C

Table 11: Historical 40-pair retrieval: question-first/last EM and support; prefill medians from separate common timing sessions. Each method uses the first five test documents, both layouts and three shuffled repeats (30 measurements). Session A is the independent confirmation. Post hoc sessions B and C reuse its test pairs. Repeated anchors are Dense/RESCOPE at 35.52/24.62 s in B, and Dense/local expansion/RESCOPE at 35.67/15.15/23.58 s in C. Accuracy is reused, not newly sampled.

Uniform VS uses one global tail estimate with equal requested counts across heads; released scaling reallocates counts using a released head configuration, and radius-8 local expansion adds nearby positions. These are support-allocation controls, not complete MInference reproductions. Relative to Uniform VS, RESCOPE gains 52.5 percentage points [37.5,67.5] in question-first EM (Holm $p < 10^{-5}$). Its gains over released scaling and local expansion are 12.5 [-10.0,35.0] and 2.5 [-15.0,20.0] points; neither is significant. Local expansion slightly exceeds the support target (8.1% relative), and the RESCOPE–local comparison has seven wins and six losses (Holm $p = 1$). All intervals here are unadjusted paired-document bootstrap 95% intervals from 20,000 resamples. Exact two-sided McNemar tests use Holm correction across the three original contrasts. Each later extension uses a separate family of two contrasts. All reported p values in these comparisons are adjusted.

Components and stronger baselines. The component and FlexPrefill extensions reuse the 40 test pairs after their outcomes were known, with separate support calibration and separate Holm families. RESCOPE improves on refinement alone by 32.5 points [17.5,47.5] ($p = 0.002$), but the 10.0-point gain over segmentation [-7.5,27.5] is unresolved ($p = 0.424$). Recalibration tests comparable final support, not additive effects at fixed component counts. FlexPrefill uses released source, block size 128, $\tau = 0.1$ and minimum budget 512 (the original paper used 1024) (Lai et al., 2025). The calibrated $\gamma = 0.2500005$ arm is 42.5 points [25.0,60.0] below RESCOPE ($p = 0.00044$); $\gamma = 0.9$ instead reaches 100%/100% at about 18% support.

Measured cost. All formal timings use the same physical RTX PRO 6000 Blackwell Server Edition, including selection, conversion, support instrumentation, prefill, cache construction and first logits; loading, tokenization, warmup and later decoding are excluded. In session C, RESCOPE is 1.51×

faster than Dense, but slower than local expansion and both FlexPrefill settings. FlexPrefill at $\gamma = 0.9$ is both more accurate and faster: RESCOPE loses 22.5 question-first points $[-35.0, -10.0]$ ($p = 0.00391$).

B.2 NATURAL-DOCUMENT QA

The historical HotpotQA and NarrativeQA adaptations use 40 documents per task, 8K–64K native text, two models, six methods and two layouts: 1,920 cached generation trajectories (Yang et al., 2018; Kočiský et al., 2018). Table 12 retains all 48 F1 cells. Realized support differs across these historical configurations; they do not use the 1.5% protocol. These are paired LongBench adaptations, not official leaderboard results (Bai et al., 2024). Both models use the same 40 complete document IDs per task, without added distractors or truncation. Generation stops at EOS or 32 tokens for HotpotQA and 128 for NarrativeQA. Official English QA normalization and maximum reference token F1 score the complete response; capped outputs remain included. Intervals use 10,000 paired-document bootstrap resamples and are exploratory, without multiplicity adjustment.

	HotpotQA F1		NarrativeQA F1	
Method	First	Last	First	Last
Ministral-3-14B-Instruct				
Dense	60.3	68.1	27.1	35.4
End-probe VS	48.8	62.3	26.1	32.9
Segment only	43.9	59.5	22.0	29.1
Refinement only	53.8	62.3	31.7	35.4
Local expansion	44.9	64.8	27.0	24.8
RESCOPE	51.4	63.4	30.5	36.3
Qwen2.5-14B-Instruct-1M				
Dense	58.3	57.4	20.3	25.5
End-probe VS	50.5	60.3	23.1	28.9
Segment only	48.9	52.4	25.1	25.1
Refinement only	51.5	60.3	25.8	29.0
Local expansion	45.8	53.1	20.2	30.7
RESCOPE	56.4	58.7	25.6	27.6

Table 12: Historical natural QA: generated-response F1 (0–100) on 40 paired documents per task. First/last denotes question position.

On question-first Qwen HotpotQA, RESCOPE gains 5.9 F1 [1.1,12.6] over end-probe VS, with no reliable gain over Dense. NarrativeQA shows no consistent improvement. On question-first Ministral HotpotQA, RESCOPE retains an 8.9-point loss to Dense (51.4 versus 60.3).

B.3 EARLIER 80-DOCUMENT CONFIRMATION

This separate Qwen study uses 80 prespecified synthetic semantic documents, two lengths, two layouts and four methods (1,280 generations). It tests the same 80 identities across both lengths and layouts, and asks whether more original observation queries substitute for refinement. `segmented64` uses 64 queries at each segment end; `static192` adds 192 original global queries; RESCOPE uses three global estimates from 64 queries with two temporary updates. Equal observation-row counts do not imply equal FLOPs. Two disjoint calibration documents set support targets of 5% at 32K and 1.5% at 128K, within 5% relative tolerance. All six sparse configurations were frozen before testing. Prompt-only cached generation uses dense greedy decoding to EOS or 256 tokens. Numeric EM requires the set of extracted five-digit values, allowing thousands commas, to contain only the gold value.

Question-first gains over `static192` are 28.75 [17.50,40.00] points at 32K (Holm $p = 0.0000169$) and 25.00 [12.50,37.50] at 128K ($p = 0.0006498$). Gains over `segmented64` are 40.00 [28.75,51.25] ($p = 0.0000001$) and 13.75 [0.00,27.50] ($p = 0.0895$), respectively. Intervals use 20,000 paired-document bootstrap resamples; Holm correction covers the four prespecified contrasts. The 128K segmentation contrast is not significant after correction. RESCOPE remains 22.5/35.0

Length	Method	EM (%) First	EM (%) Last	Support β First	Support β Last	Time (s)
32K	Dense	100.00	100.00	1.00000	1.00000	4.4302
	segmented64	37.50	100.00	0.04938	0.04859	4.2595
	static192	48.75	100.00	0.05060	0.04944	4.8208
	RESCOPE	77.50	100.00	0.04987	0.04860	5.3727
128K	Dense	100.00	100.00	1.00000	1.00000	35.6760
	segmented64	51.25	98.75	0.01474	0.01460	19.6436
	static192	40.00	98.75	0.01545	0.01513	21.4620
	RESCOPE	65.00	100.00	0.01530	0.01488	23.0668

Table 13: Earlier 80-document study: all 16 accuracy/support conditions. Times are question-first means over five prespecified test documents with three repeats each, using the same frozen configurations and PRO6000 device. They include selection through first logits and are separate from the 40-pair medians in Table 11.

points below Dense and is 21.3% slower than Dense at 32K, although $1.5466\times$ faster at 128K. Both sparse controls are faster at both lengths.

Evaluation covers 32/128/256-token prefixes. At 128K, RESCOPE question-first EM rises from 63.75% at 32 tokens to 65% at 128/256; three capped responses remain included. Strict whole-response text EM is zero in all 16 cells because responses usually include explanations. The completed-cohort summaries exclude failed calibration and an interrupted generation attempt. This study establishes neither a universal quality–latency advantage nor natural-workload gains.

B.4 DIAGNOSTIC EVIDENCE AND LIMITATIONS

Position and answer access. Historical 128K position diagnostics use 40 documents per model, with four questions scored separately by answer-isolated support recording and teacher-forced replay (Appendix C.5), not four generated answers from one cache. Across the four positions, end-probe/segment/RESCOPE EM is 32.5/89.4/98.1% for Ministral and 31.9/93.1/97.5% for Qwen; corresponding supports are 0.015/0.013/0.020 and 0.015/0.010/0.016. Dense scores 100.0/96.9%. These unequal-budget results motivate segment observations but are not calibrated generation evidence. Earlier RULER-Pos runs allow answer access; causal masking alone does not prevent information entering through the selector. Their effects cannot be attributed solely to question position. Replay is also not numerically exact: maximum logit differences are 0.47 with 3 argmax changes in 2,160 midstream checks, and 5.02 with 9 changes in 2,880 checks overall, including early final-block geometry mismatches.

Variable tracking and rounds. Under prompt-only selection and dense teacher-forced answer rows, two-hop 128K VT gives RESCOPE 71.4% on Ministral's 28 Dense-correct documents (23/40 raw successes), versus 7.1% for segmentation and 14.3% for refinement. On Qwen's 37 Dense-correct documents, RESCOPE and refinement both score 40.5% (16/40 and 17/40 raw); their paired comparison is 6:6 ($p = 1$). The component benefit is model-dependent. On the same Ministral two-hop set, answer-visible sparse scoring, prompt-only sparse scoring and prompt-only dense answer rows give RESCOPE 89.3/60.7/71.4% on the Dense-correct subset (29/20/23 raw successes out of 40). Answer access can raise or lower an arm's score. Three-hop Ministral accuracy remains 63.6% when increasing R from 3 to 4 or 5 with larger budgets; Qwen contrasts are inconclusive. Refinement rounds therefore do not correspond to demonstrated reasoning hops.

Other negative controls. At 128K, a YaRN-extended Qwen3-8B scores 37.5% with Dense and 0% with RESCOPE. General-text segmentation and global probes retain NLL penalties of 13.99% and 17.06% above Dense. At 32K, a released SeerAttention gate matches Dense's 0.75 position EM near $\beta = 0.36$, with two-hop VT EM 0.78 versus 0.85. Refining that gate was not tested. Per-head allocation and cross-layer reuse show no consistent gains.

C IMPLEMENTATION AND EVALUATION PROTOCOLS

C.1 LAYER COMPUTATION

RESCOPE changes attention support during prefill in a frozen model. Algorithm 1 uses zero-based, half-open intervals. The layer's ordinary forward produces the original Q, K, V from its input states Z. The selector retains the pre-normalization tail states and the corresponding original queries. A segment's final W queries score its full visible prefix, with a causal mask for each observation row. Segment length therefore does not limit attention to a local window. Each segment keeps its own pattern; only the global candidates are shared across segments.

The VS estimator $\mathcal{E}(q, K; v, s)$ sums row-wise softmax scores by absolute key column and by relative query–key offset, then retains the requested columns and offsets (Jiang et al., 2024). The first 30 columns and 100 most local offsets receive forced high scores before truncation. Requested counts have lower bounds of 30 columns and 50 offsets, subject to prefix length. If fewer than 100 offsets are requested, only that many forced candidates are retained. These rules also apply to the VS controls.

Algorithm 1: ReScope in one prefill layer. u concatenates head outputs; M_W and the native query map $\mathcal{Q}_\ell$ use the original tail positions.

Input: Layer inputs Z; original Q, K, V; segments I_j; window width W; number of estimates R; budgets (v_s, s_s, v_g, s_g)

$z \leftarrow Z_{[\max(0,N-W):N]}$; $\widetilde{Q} \leftarrow Q_{[\max(0,N-W):N]}$
$G \leftarrow \mathcal{E}(\widetilde{Q}, K; v_g, s_g)$
for $r = 1, \ldots, R-1$ **do**
  $u \leftarrow \mathrm{DenseAttn}(\widetilde{Q}, K, V; M_W)$
  $z \leftarrow z + W_o u$; $\widetilde{Q} \leftarrow \mathcal{Q}_\ell(z; p_{\mathrm{tail}})$
  $G \leftarrow G \cup \mathcal{E}(\widetilde{Q}, K; v_g, s_g)$
for *each segment* I_j *with endpoint* e_j **do**
  $q_j \leftarrow Q_{[\max(0,e_j-W):e_j]}$
  $P_j \leftarrow \mathcal{E}(q_j, K_{[0:e_j]}; v_s, s_s) \cup G$
$\mathcal{I} \leftarrow \mathrm{ConvertAndDeduplicate}(\{I_j, P_j\}_j, N)$
return $\mathrm{SparseAttn}(Q, K, V; \mathcal{I})$

Each temporary read uses all causally visible K, V, not the previous round's selected support. The residual update acts on hidden states z. With $W = 64$ and $R = 3$, two updates yield three global estimates. K, V and positions stay fixed within the layer. No feed-forward sublayer, additional token or training step is used. The temporary states are discarded; final attention uses the original Q, K, V, followed by the ordinary output projection, residual operations and MLP.

C.2 SPARSE EXECUTION AND COST

The existing VS kernel uses 64×64 blocks. For each query head and 64-row query block, the converter produces key-block indices from relative offsets and single-column indices from vertical candidates. It removes duplicate blocks and columns, filters causal support, and removes columns already covered by blocks. It writes each segment's union only to that segment's query blocks. No dense support mask is materialized.

The optimized selector batches slash-score reductions across eight heads and creates padding on the GPU. A Triton converter uses integer histograms and prefix sums to compact indices. Shape-based work buffers are reused for the same shape and stream, avoiding per-segment synchronization for allocation sizes. The sequential query updates remain unchanged; there are no additional computation streams. These optimizations are also applied to the sparse controls in the implementation comparison. Earlier reference-converter checks covered 48 CPU cases and recorded GPU comparisons with unchanged argmax and maximum logit difference 0.00. Those checks concern the tested conversions, not equivalence to Dense attention or every subsequent model execution.

KV heads are repeated to their associated query heads for selection and execution, while each query head selects its own support. The stored cache is not expanded. Evaluated segment lengths are multiples of 64; the last segment may be shorter. Kernel padding is trimmed from outputs

and excluded from support counts. Realized support β counts valid causal pairs after conversion, including partial blocks, and is averaged over heads, layers and inputs. Candidate overlap makes requested counts an unreliable substitute for β. Calibration uses separate documents and support counts, without answer scores.

For equal segments of length L with $L \mid N$, the number of rectangular query–key scores per head is

$$W \sum_j e_j + (2R-1)WN = \frac{WN^2}{2L} + \frac{WN}{2} + (2R-1)WN. \tag{7}$$

The global term includes R estimates and $R-1$ temporary reads. This is a score-entry count, including masked entries, not a FLOP or latency estimate. Value aggregation, W-row projections, sorting, index conversion and final sparse attention add work. Final execution depends on realized support; selection is excluded from β but included in complete prefill timing.

C.3 Model-specific query reconstruction

The evaluated models are Qwen2.5-14B-Instruct-1M and Ministral-3-14B-Instruct-2512, using bf16 and batch size one. The selector captures states before input RMSNorm and takes its initial queries directly from the attention module. Updated queries use the pretrained input RMSNorm, query projection and rotary module at the original positions; learned projection biases are retained. The code supports per-head query normalization, which neither evaluated model uses.

The reconstruction follows the native model's order of normalization, query projection, head normalization when present, rotary encoding and model-specific position scaling. Original token positions remain fixed. Estimation and temporary attention use the usual $d^{-1/2}$ dot-product factor; final attention uses the original model queries.

C.4 Generation and answer access

Cached generation performs one prompt-only prefill with `use_cache=True`. The last prompt state predicts the first token. Subsequent tokens use dense attention over the full retained cache and generated tokens; selection is not rerun. The generation runner does not receive reference answers, and stopping follows the stated EOS tokens and output cap. Cache representations still depend on the preceding sparse prefill.

Teacher forcing supplies reference answer tokens and is a separate protocol. Historical diagnostics distinguish answer-visible selection, prompt-only selection with sparse answer rows, and prompt-only selection with dense answer rows. In the last case, the prompt row predicting the first answer token remains sparse. This path is a surrogate for scoring under a gold prefix, not an actual generated trajectory. A historical 40-document, five-arm check agreed with cached generation on 199/200 answer decisions, with one bf16-tie disagreement. It does not establish general decision equivalence. Re-prefilling and reselecting after each generated token also evaluates a different algorithm.

A causal attention mask constrains attended keys, not the information used to select them. A tail query can encode later input and change the support assigned to an earlier row. Moving observation windows outside answer spans therefore does not guarantee answer isolation. In four historical streams, perturbing later answers changed first-answer logits by up to 0.59 for one-shot selection, 0.88 for global updates and 0.38 for RESCOPE, with no argmax changes; segment-only selection was unchanged in that test. Answer-visible diagnostics are not evidence for prompt-only generation, and answer access can raise or lower a score.

C.5 Answer-isolated support recording and replay

Intermediate-position diagnostics require a separate selection stream for each question. The stream preserves every token before that question's answer-prediction row, including earlier answer examples. The current and later answer spans are replaced by decoys that differ in every digit; the final answer is removed so the stream ends at the answer marker. Evidence in the document remains available. This construction removes the scored and later answer text without changing the scored row's causal prefix. Deleting all answers or replacing earlier answers would alter that prefix and evaluate a different input.

The first pass records selected columns and offsets at each layer, keyed by the segment's first query block when applicable. A second pass replays these indices on the gold document cut at the end of the 64-row block containing the scored answer. Columns retain absolute positions and slash candidates retain relative offsets. The cut preserves the relevant block geometry; invalid columns beyond it are removed. Separate replays provide sparse-answer and dense teacher-forced-answer readings. Each question therefore requires one full recording pass and two cut replays. These are diagnostics, not multiple generated answers from one shared cache.

Replay is not generally bit-exact. Historical checks recorded a maximum logit difference of 0.47 and 3 argmax changes in 2,160 intermediate-question checks. Across all 2,880 checks, the maximum was 5.02 with 9 changes. Earlier last-question runs used a partial final block in one pass and a full block in the other. Later runs padded with decoys after the answer marker while trimming the probe to the marker, aligning the block geometry without changing its inputs. Removing columns can also change floating-point summation order. These fixes do not retroactively make the historical records bit-exact. First-digit agreement, whole-answer teacher-forced accuracy and cached-generation accuracy remain distinct measurements.

D EXISTING INPUT LAYOUTS AND EVIDENCE POSITIONS

This descriptive audit supplies the source evidence for Fig. 1. It examines existing templates, recorded benchmark requests, and human annotations. It does not measure sparse attention or establish the prevalence of a layout in production. Here *task/question* denotes semantic input, *evidence* denotes supporting material, and *answer* denotes the subsequent generated output. These are distinct from attention queries Q. The schematics omit system wrappers and are not drawn to scale.

Existing templates. The LangChain v0.2 local-RAG tutorial (LangChain, 2024) uses the public `rlm/rag-prompt`. Its manifest at commit `50442af133e61576e74536c6556cefe1fac147cad032f4377b60c436e6cdcb6e` places `Question:` before `Context:`, followed by an `Answer:` cue. The QASPER authors' LED baseline likewise concatenates question, separator, and article; this is an encoder–decoder baseline, not a reproduction of decoder-only sparse prefill. Conversely, LongBench's released QASPER, HotpotQA and 2WikiMQA prompts place context before question (Bai et al., 2024). Runtime layout is a property of the template and system, not the order of fields in a dataset file. The template examples establish existing usage, not a claim that these tutorials use 128K inputs.

Recorded agent requests. We inspect the public SWE-agent 1.1.0 / Claude 4 Sonnet trajectories archived by SWE-bench (SWE-bench, 2025). Before inspecting their contents, we selected ten evenly spaced ranks from all 500 lexicographically sorted task IDs. This is a descriptive sample, not a random or population-representative sample. We analyze each recorded `trajectory[i].query`, the actual per-call input, rather than concatenating the complete trajectory history. The ten tasks contain 611 requests, of which 595 end in a tool message. Each request contains one exact copy of the original task text; later paraphrases, summaries or partial repetitions are not excluded by this test. For each task's last request, that text ends within the first 0.414–3.605% of visible input characters, with actions and tool outputs following it.

For length reference only, we tokenize visible message text and tool-call names/arguments with `cl100k_base`, excluding chat wrappers and tool schemas. Eight of ten tasks have a request reaching 32,768 reference tokens; one reaches 65,536 and none reaches 131,072. Across calls, these counts are 160, eight and zero. Repeated calls are correlated. These are neither Claude billing counts nor Qwen token counts, and do not establish naturally occurring 128K requests. All recorded requests include cache-control hints: full-context replay concerns cold/full prefill, not demonstrated latency benefits for agents with prefix-cache hits. We have not labeled which tool outputs are causally necessary evidence, and relevant material may recur.

Human evidence positions. We audit all of QASPER v0.3 dev (Dasigi et al., 2021): 281 articles, 1,005 questions and 1,764 answer annotations. The official train/dev archive has SHA256 `a28fdf966db827bcee3d873107d6b6669864fb7ca8fbf73a192f5e39191bdb5a`. We retain original `full_text` paragraph order and locate evidence by exact paragraph matching after whitespace normalization. We deduplicate evidence across annotators within each question, while

preserving every location when identical paragraphs recur. Position is the paragraph's starting character offset divided by the concatenated body length; titles, abstracts, section headings and floating figures/tables are excluded.

We match 1,802 unique question–paragraph locations covering 888 questions. Of these locations, 513 (28.47%) start in the first third, 801 (44.45%) in the middle third, and 488 (27.08%) in the final third. Percentages condition on successful matches, not all annotations. We separately retain 203 figure/table evidence items without a linear body location and 95 unmatched text evidence items; 163 answer annotations are unanswerable and 49 answerable annotations contain no evidence. These counts have different units and should not be summed into the matched-location denominator. There are 435 questions with at least one matched middle-third paragraph. This does not imply that every such paragraph is necessary or that every necessary piece of evidence was matched.

QASPER provides evidence about natural document structure rather than a native 128K workload. Tokenizing all 1,585 articles' titles, abstracts, headings and bodies in order with the Qwen tokenizer yields roughly 5.3K tokens on average and 4.9K at the median. Only two articles reach 32K, both in training, and none reaches 64K or 128K. The audit preserves original documents and does not insert answers or filler. Longer multi-document constructions, if evaluated, require a separate protocol.

Interpretation. The two audits establish that tasks can precede accumulating material and that human evidence is distributed within documents. Contextualized tail queries may still encode the task. Their effectiveness for sparse selection, and any benefit from refining them, require method comparisons under controlled support and measured latency; the audit alone establishes neither.

Controlled position test in Figure 1. The figure's answer scores come from the support-calibrated 40-document confirmation summarized in Appendix B, not from either descriptive audit above. Each token-matched pair moves one identical question block before or after the same synthetic document. Both layouts generate an answer after the complete prompt. The question-first and question-last Uniform VS scores are 25.0 and 97.5, while Dense scores 100 in both cases. The experiment uses Qwen2.5-14B-Instruct-1M near 128K, support-only calibration on two separate document pairs, and cached greedy generation to EOS or 256 tokens. Moving the question also changes its available context and distance from the tail; it does not isolate distance as a single cause. These forty pairs, the earlier 80-document study and the current primary 80-document study are separate populations.

E Protocols for the selection analysis

The panels in Fig. 2 use separate studies of Qwen2.5-14B-Instruct-1M near 128K, with different sampling units. They do not form one paired quality–latency experiment.

Fixed-layer output error. Panel (a) uses three fixed question-first development inputs, zero-indexed layers 10, 20 and 30, and all 40 heads. Dense causal attention in preceding layers supplies the same captured states to both selectors. We compare one 64-query tail window with 16 segment-end windows of 64 queries each; every window reads its full visible prefix. Each layer–selector setting is calibrated on one separate input toward 1.5% final causal support, with 5% relative tolerance. The measured test means are 1.71% for tail observation and 1.52% for segment observation, so support is not exactly matched.

For a prompt of length N, the fixed evaluation rows are $i_j = \lfloor (j + \frac{1}{2})N/256 \rfloor$, $j = 0, \ldots, 255$. For each head, we compute the relative output error

$$e_h = \left(\frac{\sum_{j,d} (O^{\text{sparse}}_{h,i_j,d} - O^{\text{dense}}_{h,i_j,d})^2}{\max\{\sum_{j,d} (O^{\text{dense}}_{h,i_j,d})^2, 10^{-20}\}} \right)^{1/2}. \tag{8}$$

The sums cover sampled rows and head dimensions in FP32. Sparse outputs use the final converted indices and captured original Q, K, V. The 360 pairs are $3 \times 3 \times 40$ correlated head observations from three inputs. These local errors do not measure generated-answer quality or accumulated errors across sparse layers.

Candidate novelty. Panel (b) reuses a saved selector trace from one 131,002-token input at layer 20. Let S_{hm} be the segment-column set and G^r_{hm} the global column set from estimate $r \in \{0, 1, 2\}$

Windows	Original	Two updates	Windows	Original	Two updates
1	11.29	49.32	6	46.16	193.76
2	18.11	77.83	8	60.22	252.21
3	25.16	106.80	10	74.00	309.41
4	32.18	136.05	15	108.91	455.38
5	39.24	164.77	20	143.81	601.10

Table 14: Mean selector time in milliseconds for the ten measured segment-end window counts. "Two updates" means two updates at every window. Each value averages three repetitions per input and then the two inputs. These are total times for separate selector strategies, not additive pipeline costs or complete prefill times.

for head h and segment m. All sets are deduplicated and filtered to the segment's visible prefix. For $r = 1, 2$, the plotted percentage is

$$100\,\frac{|G^r_{hm} \setminus (S_{hm} \cup \bigcup_{t<r} G^t_{hm})|}{|G^r_{hm}|}. \tag{9}$$

The tail window's three estimates share fixed current-layer K, V and positions. The 640 points are 40 heads times 16 segments of one input, reusing global estimates before prefix filtering. These descriptive column-set measurements precede vertical–slash expansion. They are not independent samples or final support measurements, and do not establish that new columns improve answers.

Generated-answer comparison. Panel (c) uses the current 80-document primary confirmation study in Appendix A, restricted to 128K and question-first inputs. It is distinct from the earlier 80-document study. Selectors share segment observations and use the four predefined support targets 1.5, 3, 6 and 12%. The horizontal coordinate is measured final support; vertical intervals are pointwise 95% Wilson intervals over documents. The expanded-query branch scores 192 original rows, while ReScope scores 64 rows in three estimates. Equal row counts do not imply equal FLOPs because updates also read values and apply projections. The comparison tests complete selectors, not the union operation alone.

Selector-cost replay. Panel (d) uses two earlier question-first inputs of 131,002 tokens on an RTX PRO 6000. We replay captured layer-20 states from dense upstream attention across all 40 heads. The ten measured window counts are shown in Table 14. Each 64-query window ends at its segment endpoint, rounded down to a 64-token boundary, and reads its causal prefix. Per head and round, 1,280 columns and 1,280 relative offsets are integer-partitioned across windows. Per-window quotas therefore change with the window count; neither final support nor FLOPs is matched.

One untimed warmup precedes three repetitions per input and configuration, giving $10 \times 2 \times 2 \times 3 = 120$ records. Faint marks are per-input means; lines join two-input means. All repetitions are retained. Timing includes unoptimized scoring, temporary reads, updates, projections, ranking and candidate concatenation. State capture, KV head repetition, final index conversion, sparse attention and full-model prefill are outside the interval. Updating every window is a cost control, not ReScope's single-tail-window allocation or a pipeline profile.

Updates, cost and candidate diversity. Matching final support does not match selector computation: temporary updates add value reads and projections to query scoring. Candidate turnover quantifies changes in selected columns, while final-answer quality evaluates the complete selector. These measurements do not by themselves isolate contextual content from the effect of additional computation or candidate diversity.

Model and execution scope. The native query map includes model-specific scaling after RoPE when required. Historical second-model comparisons are reported separately from the current method definition. Generic reduction, conversion and buffer optimizations are shared with the VS controls. The number of temporary updates remains an algorithmic choice, and their computation is included in complete prefill latency.

F STUDY SCOPE

The primary confirmation, transfer tests and exploratory extension in Appendix A retain their complete predefined result grids. Earlier studies are summarized in Appendix B under their original populations and scoring rules. The two groups are not pooled, and historical outcomes are not independent confirmations of choices made after observing them.

The detailed historical studies use their original populations, scoring rules and measurement boundaries. Favorable and unfavorable results follow the same inclusion rule, and the historical summary states the scope of each comparison.

G COMPLETE PREFILL MEASUREMENTS

The original controlled quality–time grid in Table 1 and Fig. 4 reports the configurations used in the initial budget scan. Its timing boundary and numerical checks are described in Appendix A.6. The optimized ReScope configurations are reported separately in Table 15; their entries give mean accuracy and complete-prefill latency. Each row compares the optimized configuration with the original baseline at the same observed accuracy.

		Original baseline			Optimized ReScope	
Length	Layout	Method	Score (%)	Time (s)	Score (%)	Time (s)
32K	Before	FlexPrefill $\gamma = .75$	96.25	5.511	96.25	4.409
32K	Before	FlexPrefill $\gamma = .9$	100.00	6.090	100	4.87
128K	Before	FlexPrefill $\gamma = .75^{\dagger}$	95.00	24.435	95.00	19.55
128K	Before	FlexPrefill $\gamma = .9^{\dagger}$	100.00	31.829	100.00	25.463
128K	Before	Segments / 6%	88.75	27.578	88.75	22.062
128K	Before	Segments / 12%	98.75	32.470	98.75	25.976
32K	After	Segments / 5%	100.00	5.065	100	4.05
128K	After	Segments / 1.5%	100.00	24.119	100	19.30

Table 15: Mean accuracy and complete-prefill latency for optimized ReScope on the synthetic workload. Before and After denote question-first and question-last inputs. Original baseline measurements retain their recorded precision; optimized mean latencies are reported in seconds. Each row has the same observed accuracy for the listed baseline and optimized ReScope. Equal observed accuracy does not establish statistical equivalence of population quality. † retains the original FlexPrefill timing supplement's numerical-repeat limitation (Appendix A.6).

Natural-question F1 results remain separate in Table 3. The 80-paper cohort uses development calibration, whereas the independent 40-paper cohort uses per-input support calibration. The fixed-configuration natural-task confirmation and paired comparison summaries are reported in Appendix H.

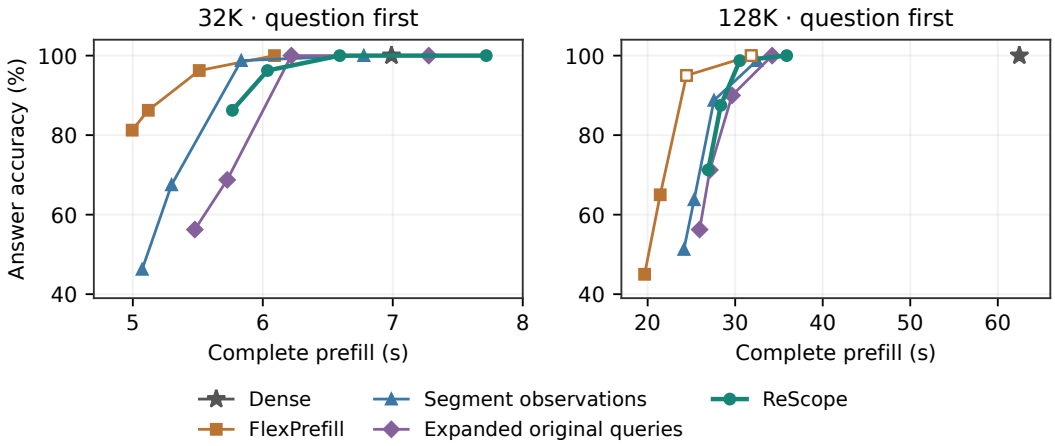


Figure 4: **Original implementation: complete measured budget grid.** Quality uses 80 documents; timing uses three separate documents and two repeats on A800. Lines join tested configurations. Hollow FlexPrefill points retain the supplemental timing's numerical-repeat limitation. The optimized operating points are reported separately in Table 15.

H CONTROLLED QUALITY, COST AND UPDATE COMPARISONS

Natural-task support and latency. The VS comparisons use 3% and 6% realized-support settings, with relative support mismatch no greater than 1%. Calibration is fixed on development inputs; test inputs require zero budget-search trials. At similar observed answer quality, normalized complete-prefill latency is 80–85 for ReScope against 100 for the corresponding baseline. These comparisons remain separate from the earlier 40-paper protocol that calibrates support per input. Similar observed quality does not by itself establish population quality equivalence.

Contribution of the update branch. Table 16 summarizes controlled answer-quality comparisons. F1 differences are in percentage points. Matching selector time and matching realized support are distinct controls; the table states the constraint for each comparison.

Comparison	Controlled quantities	Observed answer-quality difference
Updates vs. more original queries	Selector-time difference $\leq 5\%$	Update gain $\geq$ 2 F1; paired 95% interval lower endpoint > 0
Updates vs. norm-controlled perturbation	Relative realized-support difference $\leq 1\%$	Update gain ≥ 2 F1
Update-specific candidates vs. original-query replacements	Fixed segment candidates and final support	Replacement lowers F1 by ≥ 2
Real vs. permuted temporary-read values	Same budget and execution path	Real-value gain ≥ 2 F1

Table 16: Controlled update-branch comparisons. Inequalities summarize the measured effects without specifying an exact effect size or interval endpoint. Candidate replacement changes the additional branch while retaining segment candidates and final support. Value permutation applies to temporary selection reads; final attention uses the original Q, K, V.