How Early Does Context Reduction Become Computationally Valuable? Capacity–Value Separation in Long-Context Prefill

Anonymous authors
Paper under double-blind review

Abstract

Long-context prefill is expensive because a large active context must be processed across many Transformer layers. Context reduction creates a fundamental trade-off: earlier reduction lets more subsequent computation benefit from a shorter context, whereas deeper representations may support greater reliable reduction capacity. Yet early placement alone is insufficient: enough context must be removed reliably to yield a latency benefit. We ask whether the depth with the greatest reliable reduction capacity also minimizes measured prefill latency.

Under a common reliability requirement, we determine at each depth the minimum token-retention ratio satisfying the same controlled evidence-retention threshold and measure prefill latency at that operating point. For Llama-3.1-8B and Ministral-8B, the capacity-optimal depths are L10 and L12, whereas the speedup-optimal depths occur earlier, at L2–L3 and L2. Crucially, these faster operating points retain 10–22 percentage points more context than their capacity-optimal counterparts yet achieve lower measured prefill latency. We call this mismatch capacity–value separation.

Across the evaluated configurations, this separation persists when sufficient reliable reduction is available early and diminishes when reliable early reduction is limited. On LongBench, aggregate scores remain within 1.46 points of their downstream-only baselines across the evaluated early-layer grid, although category-level responses are more heterogeneous. These findings motivate treating the first context-reduction point as a conditional optimization decision, separate from later downstream selection.

Keywords: Long-context inference, Context reduction, Prefill acceleration, Reduction depth, Reliable reduction capacity, Evidence retention, Capacity–value separation

1 Introduction

Long-context inference substantially increases Transformer prefill cost because a long prompt must be processed through attention and token-wise computation across many layers (Vaswani et al., 2017; Dao et al., 2022; Dao, 2024; Jiang et al., 2024a; Fu, 2024). Recent work reduces this cost through sparse prefill attention, token pruning, and reduced prompt-depth computation (Jiang et al., 2024a; Qiao et al., 2025; He et al., 2026). Context reduction offers a complementary opportunity: once the active sequence becomes shorter, subsequent Transformer layers operate on fewer tokens. Where this reduction first becomes effective, however, creates a fundamental tradeoff. Earlier reduction provides greater computational opportunity because more remaining computation can benefit from the shorter context, whereas deeper representations may support greater reliable reduction capacity. Early placement alone is therefore insufficient: an early reduction must remove enough context reliably to produce a meaningful latency benefit. Choosing the first context-reduction depth requires balancing these two effects (Figure 1).

Prior work has shown that token-selection behavior and context-reduction policies vary with Transformer depth, motivating intermediate, adaptive, and progressive reduction under method-specific criteria (Shi et al., 2026; Jin et al., 2026; Long et al., 2026; Jo et al., 2026; Taniguchi et al., 2026; Fu

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.

et al., 2024; Yang et al., 2024; Cai et al., 2025; McDanel et al., 2026; Chen et al., 2026). These methods demonstrate that reduction can be depth-dependent, but they do not establish whether the depth supporting the greatest reliable reduction also provides the greatest computational value. Moreover, apparent early-layer reliability can depend on how relevance is scored and converted into a retained token set, so candidate depths must be compared under a controlled evaluation procedure. We therefore ask: under a common reliability requirement, does the depth supporting the greatest reliable reduction capacity also minimize measured prefill latency?

Answering this question requires comparing candidate depths at reliability-matched operating points. A fixed token-retention ratio is insufficient because retaining the same fraction of context does not imply preserving the same amount of query-relevant evidence at different depths. For each candidate first context-reduction depth, we therefore estimate the minimum token-retention ratio satisfying the same controlled evidence-retention threshold and measure prefill latency at the resulting operating point. This separates reliable reduction capacity (how much context can be removed while satisfying the reliability requirement) from computational value, operationalized here as measured prefill latency.

Our results show that these two objectives can favor substantially different depths. For Llama-3.1-8B, the capacity-optimal depth is L10, whereas the speedup-optimal depth is L2 or L3 depending on the evidence-retention threshold. For Ministral-8B, the corresponding depths are L12 and L2. Crucially, across the evaluated model–threshold conditions, the speedup-optimal operating points retain 10–22 percentage points more context than their capacity-optimal counterparts yet achieve lower measured prefill latency. Their latency advantage therefore cannot be attributed to stronger reduction. Instead, a smaller reliable reduction can provide greater computational value when it becomes effective sufficiently early. We refer to this empirical mismatch between the two optima as capacity–value separation.

Capacity–value separation is conditional rather than universal. Across the evaluated configuration controls, it persists when sufficient reliable reduction is available at early Transformer depths and diminishes when reliable early reduction is limited; under several such conditions, the capacity- and speedup-optimal depths coincide. The relevant condition is therefore not early placement alone, but whether sufficient context can be removed reliably early enough to offset the benefit of waiting for greater later-stage reduction capacity. This boundary suggests separating the first context-reduction point from later downstream selection: an earlier, more conservative reduction can reduce intervening Transformer computation while preserving later selection based on richer representations.

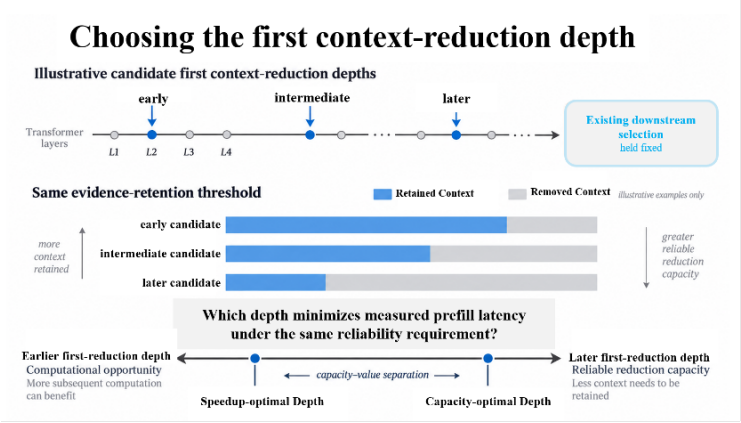


Figure 1: **Reliability–computation tradeoff across reduction depth.** Earlier reduction benefits more subsequent computation, whereas deeper reduction may support greater reliable reduction capacity. We ask whether the capacity-optimal depth also minimizes prefill latency. Candidate depths and retained-context bars are illustrative.

We further examine downstream sensitivity to the exact early-layer operating point. Using FastKV and GemFilter as representative downstream execution settings (Jo et al., 2026; Shi et al., 2026), we evaluate a local early-layer depth–retention grid on LongBench (Bai et al., 2024). Aggregate performance remains relatively close to the corresponding downstream-only baselines across multiple evaluated settings, providing a local downstream sensitivity check rather than a second optimization criterion.

Our contributions are: (1) Reliability-matched reduction-depth analysis. We compare candidate first context-reduction depths under a common controlled evidence-retention requirement, separating reliable reduction capacity from computational value without conflating depth with reliability differences. (2) Capacity–value separation. We show that the depth maximizing reliable reduction capacity can occur substantially later than the depth minimizing prefill latency, even though the faster operating point retains more context. (3) An empirical boundary and design implication. We show that this separation persists when sufficient reliable reduction is available early and diminishes when early reliability is limited, motivating separate treatment of the first context-reduction point and later selection or pruning stages.

2 Reliability-Matched Evaluation of Reduction Depth

We compare candidate first context-reduction depths under a common reliability requirement rather than at a fixed token-retention ratio. For each depth, we estimate the minimum context retention required to satisfy the same evidence-retention threshold and measure prefill latency at the resulting reliability-matched operating point. This separates reliable reduction capacity from computational value.

2.1 Reduction Depth and Evidence Retention

Let ℓ denote the first context-reduction depth along the evaluated execution path, and let $\rho \in (0, 1]$ denote the token-retention ratio immediately after this reduction. Smaller ρ corresponds to stronger context reduction. The downstream execution path may contain additional selection stages; throughout the principal cross-depth comparison, these stages are held fixed so that varying ℓ changes only where the active context first becomes shorter.

We use a controlled Needle-in-a-Haystack (NIAH) protocol to measure whether designated query-relevant evidence survives context reduction, following long-context evaluations based on controlled evidence placement (Liu et al., 2024; Li et al., 2025). Let $E(\ell, \rho)$ denote evidence retention, defined as the mean fraction of designated evidence tokens preserved after reduction across the controlled instances. Given an evidence-retention threshold τ, an operating point satisfies the reliability requirement when

$$E(\ell, \rho) \geq \tau.$$

Throughout the depth-wise analysis, *reliable* refers specifically to satisfying this controlled evidence-retention criterion and should not be interpreted as downstream task accuracy.

2.2 Reliability-Matched Operating Points and Reliable Reduction Capacity

A fixed token-retention ratio does not ensure comparable reliability across depths. We therefore define the minimum reliable retention ratio at depth ℓ as

$$\rho^*(\ell; \tau) = \min \left\{ \rho \in [\rho_{\min}, 1] : \widetilde{E}(\ell, \rho) \geq \tau \right\},$$

where $\widetilde{E}(\ell, \rho)$ denotes the piecewise-linear interpolation of the measured evidence-retention values over the evaluated token-retention ratios, and $\rho_{\min}$ is the smallest evaluated retention ratio. The interpolation procedure and boundary handling are detailed in Appendix A.2.

A smaller $\rho^*(\ell; \tau)$ indicates greater reliable reduction capacity, because more context can be removed while satisfying the same reliability requirement. The capacity-optimal depth is therefore

$$\ell_{\text{cap}}(\tau) = \arg\min_{\ell \in \mathcal{D}} \rho^*(\ell; \tau),$$

where $\mathcal{D}$ denotes the evaluated set of candidate first context-reduction depths.

The reliability-matched operating points are computed from the full retention-ratio sweep under the fixed Full evaluation configuration. The fixed-budget and earliest-reliable-depth views in Figure 2 are diagnostic summaries and do not themselves define these operating points.

2.3 Computational Value and Capacity–Value Separation

Let $T(\ell, \rho)$ denote measured prefill latency when the first context reduction occurs at depth ℓ with token-retention ratio ρ, while the downstream execution path is held fixed. The speedup-optimal depth under evidence-retention threshold τ is

$$\ell_{\mathrm{spd}}(\tau) = \arg\min_{\ell \in \mathcal{D}} T(\ell, \rho^*(\ell; \tau)) .$$

Thus, computational value is assessed by measured prefill latency at each depth's reliability-matched operating point. Because the reference latency is fixed within each evaluated downstream execution setting, minimizing measured prefill latency is equivalent to maximizing the corresponding prefill speedup.

The two optima capture different properties of the first context-reduction depth. ℓ_{cap} identifies the depth with the greatest reliable reduction capacity, whereas ℓ_{spd} identifies the depth with the lowest measured prefill latency under the same reliability requirement. We define capacity–value separation as the empirical case

$$\ell_{\mathrm{cap}}(\tau) \neq \ell_{\mathrm{spd}}(\tau).$$

This definition does not impose an ordering between the two optima; the finding that $\ell_{\mathrm{spd}}(\tau) < \ell_{\mathrm{cap}}(\tau)$ in the principal setting is empirical rather than part of the definition. Section 3 examines whether this separation occurs across the evaluated models and configurations, and under what conditions.

3 When Does Reduction Become Computationally Valuable?

We now use the reliability-matched framework in Section 2 to ask whether reliable reduction capacity and computational value favor the same first context-reduction depth.

Unless otherwise stated, the principal depth-wise analysis evaluates Llama-3.1-8B (Grattafiori et al., 2024) and Ministral-8B under a fixed evaluation configuration. Reliability is measured using the controlled Needle-in-a-Haystack protocol, and computational value is assessed by measured prefill latency at 128K. Full implementation, evaluation, and timing details are provided in Appendix A.

3.1 Depth-wise Reliability Depends on the Evaluation Configuration

We first ask whether weak evidence retention within the first few Transformer layers is intrinsic to those depths. Figure 2 compares several relevance-scoring and selection configurations for Llama-3.1-8B and Ministral-8B. The configurations differ in how relevance is read from model representations and how the resulting scores are converted into retained tokens; in particular, the comparison includes Pre-RoPE and Post-RoPE readouts, where RoPE denotes rotary positional encoding (Su et al., 2024). Unless otherwise stated, the principal analysis uses the Full configuration (Pre-RoPE + Query Aggregation + PDSC + Blocking + Head/Tail Retention, where PDSC denotes position-dependent score calibration), with component definitions and implementation details provided in Appendix A.3.

The top row fixes the token-retention ratio at $\rho = .60$. Because all configurations retain the same fraction of the input context, differences in evidence retention reflect which tokens are retained, rather than differences in token budget. Evidence retention varies substantially across configurations at the same depth. In particular, the Post-RoPE, last-query configuration performs worse at several first-few-layer depths, while alternative configurations preserve substantially more evidence under the same retention budget. The profiles are also non-monotonic: deeper processing can improve evidence retention, but deeper reduction depths do not uniformly dominate shallower ones.

This configuration dependence is consistent with prior work showing that token relevance, ranking stability, and useful selection behavior can vary substantially across Transformer depth (Shi et al., 2026; Jo et al., 2026; Taniguchi et al., 2026; McDanel et al., 2026). Our purpose here, however, is not to rank relevance-scoring methods, but to determine whether sufficient reliable reduction can be exposed within the first few Transformer layers under a fixed evaluation procedure.

The bottom row fixes the evidence-retention threshold at $\tau = .95$ and reports, for each token-retention ratio, the earliest evaluated depth satisfying that requirement. More aggressive reduction

generally shifts the earliest reliable depth later, but the relationship again depends strongly on the evaluation configuration. Some configurations expose reliable reduction within the first few Transformer layers at retention ratios for which others require substantially deeper processing.

These results establish the premise for the principal analysis: the amount of reliable reduction available within the first few Transformer layers depends jointly on depth and the evaluation configuration. Poor evidence retention under one configuration therefore does not by itself imply that those layers lack usable relevance information. Figure 2 is used to characterize this configuration dependence rather than to rank relevance-scoring methods. In the principal comparison below, the evaluation configuration is fixed across candidate depths; the Figure 2 views are diagnostic and do not themselves define the reliability-matched operating points used in the capacity–value analysis.

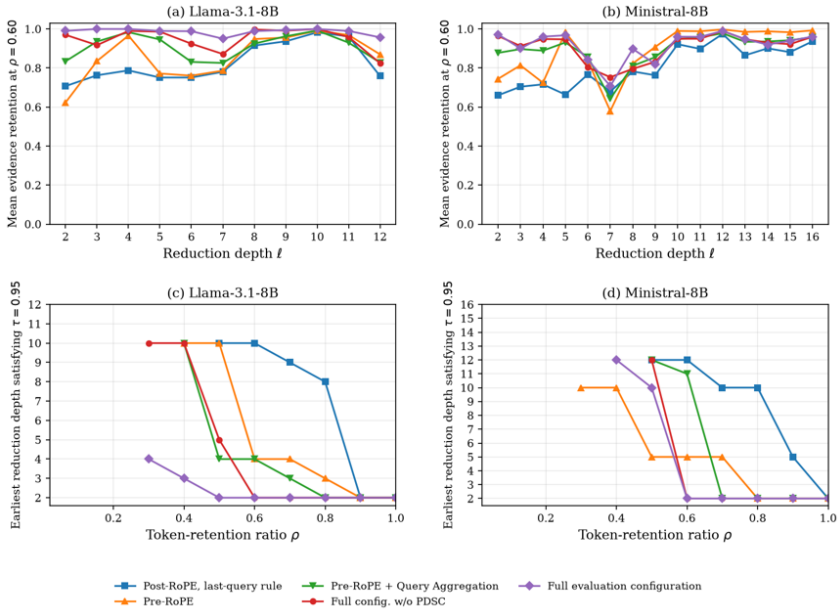


Figure 2: **Reliability across Transformer depth depends on the evaluation configuration.** Top: mean evidence retention at fixed $\rho = .60$. Bottom: earliest evaluated depth satisfying $\tau = .95$ as retention varies. These diagnostic views do not define the reliability-matched operating points.

3.2 Reliable Reduction Capacity Peaks at Deeper Depths

We next measure how much context can be removed reliably at each candidate first reduction depth. For each depth ℓ, we determine its minimum reliable retention ratio $\rho^*(\ell; \tau)$ under the common evidence-retention requirement using the interpolation procedure defined in Section 2.2.

Figure 3(a,b) shows the resulting reliable-reduction-capacity profiles for the strict representative threshold $\tau = .98$. Lower values of $\rho^*(\ell; \tau)$ indicate greater reliable reduction capacity. The profiles vary substantially with Transformer depth and are not monotonic: some deeper depths support substantially lower reliable retention ratios than the first few Transformer layers, but increasing depth does not uniformly increase reliable reduction capacity. The two models also exhibit different depth-wise profiles, indicating that reliable reduction capacity cannot be represented by a single common depth schedule.

Most importantly, the greatest reliable reduction capacity is not obtained within the first few Transformer layers. Across all three evaluated evidence-retention thresholds, the capacity-optimal depth is L10 for Llama-3.1-8B and L12 for Ministral-8B. At $\tau = .98$, the estimated minimum reliable retention ratio reaches .28 at L10 for Llama-3.1-8B and .50 at L12 for Ministral-8B. Results for all three evidence-retention thresholds are summarized in Table 1, with complete depth-wise profiles reported in Appendix C.

Thus, the first few Transformer layers do not maximize reliable reduction capacity. The strongest reliable reduction becomes available substantially later.

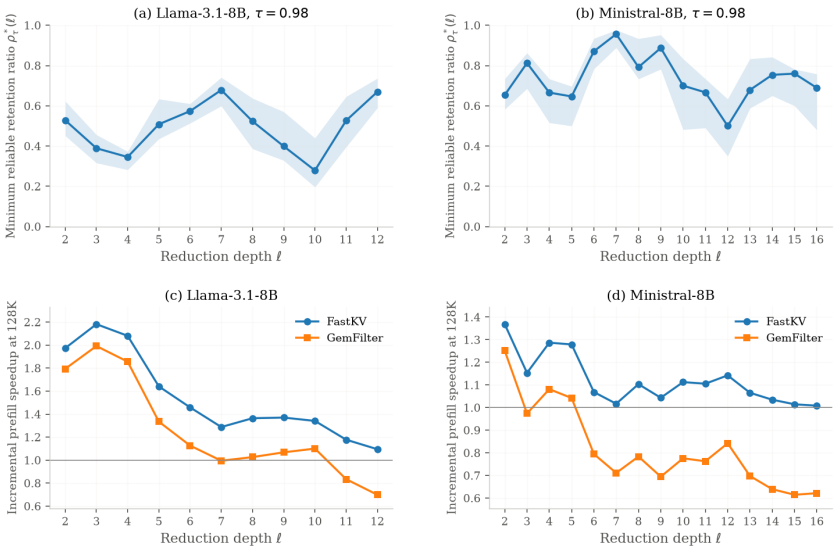


Figure 3: **Reliable reduction capacity and computational value at $\tau = .98$.** Top: minimum reliable retention ratio $\rho^*(\ell; .98)$. Bottom: incremental 128K prefill speedup over the corresponding FastKV- or GemFilter-only setting at the same operating points; $1\times$ denotes no added acceleration.

3.3 Computational Value Peaks Earlier

We next evaluate the computational consequence of operating each candidate first reduction depth at its reliability-matched retention ratio. For each ℓ, context reduction first becomes effective at that depth while the remaining downstream execution path is held fixed.

Figure 3(c,d) reports incremental prefill speedup at 128K relative to the corresponding downstream execution setting without the added first context-reduction point. A value above $1\times$ indicates lower measured prefill latency. As in the capacity panels, the main figure reports the representative threshold $\tau = .98$.

The computational-value profiles differ markedly from the capacity profiles. For Llama-3.1-8B, the speedup-optimal depth is L2 or L3 depending on the evidence-retention threshold; for Ministral-8B, it is L2 across all three thresholds. We use FastKV and GemFilter as two representative downstream execution settings (Jo et al., 2026; Shi et al., 2026). Although their absolute speedups differ, both identify the same speedup-optimal depth for every evaluated model–threshold pair. This agreement serves as a robustness check on the depth relationship rather than a comparison between the two downstream methods.

Crucially, the speedup-optimal operating points retain more context than the corresponding capacity-optimal points. Their lower latency therefore cannot be attributed to stronger context reduction. Instead, a smaller reliable reduction becomes effective early enough for more of the remaining Transformer computation to operate on the shortened context. Deeper reduction may support greater reliable reduction capacity, but more prefill computation has already been performed before that reduction takes effect.

Thus, the speedup optimum is determined jointly by how much context can be removed reliably and how much subsequent computation remains to benefit from that reduction.

3.4 Capacity–Value Separation

The capacity and latency profiles therefore instantiate the capacity–value separation defined in Section 2: the first reduction depth with the greatest reliable reduction capacity is not the depth with the lowest measured prefill latency.

Table 1 summarizes the magnitude of this separation across all three evidence-retention requirements. Here, $\tau \in \{.90, .95, .98\}$ denotes the required evidence-retention level, whereas the retention columns report $\rho^*(\ell; \tau)$, the estimated minimum fraction of context tokens retained while

satisfying that requirement. These quantities therefore represent different forms of retention. For example, $\tau = .98$ with $\rho^*(\ell;\tau) = .39$ means that the interpolated estimate of the minimum context retention satisfying the 98% evidence-retention requirement is approximately 39%; .39 is not an evidence-retention score.

Across the six model–threshold conditions, the speedup-optimal operating points retain approximately 10–22 percentage points more context than the corresponding capacity-optimal points, yet achieve lower measured prefill latency. For Llama-3.1-8B, the capacity-optimal depth remains L10 across $\tau = .90, .95, .98$, with retention ratios .15, .21, and .28, whereas the speedup-optimal depths occur at L2 or L3 with retention ratios .37, .31, and .39. For Ministral-8B, the capacity-optimal depth remains L12 with retention ratios .28, .36, and .50, whereas L2 is speedup-optimal with retention ratios .48, .55, and .65.

Table 1 also reports prefill speedup relative to full-attention prefill at the speedup-optimal and capacity-optimal operating points. Across the evaluated conditions, the ratio between these speedups ranges from 1.20–1.78 for FastKV and 1.49–2.04 for GemFilter. Because the two speedups within each comparison share the same full-attention baseline, their ratio is equivalently the latency ratio between the capacity-optimal and speedup-optimal operating points.

The central result is therefore not that a particular first-few-layer depth is universally preferable. Rather, under a common reliability requirement, maximizing reliable reduction capacity need not minimize measured prefill latency. A first reduction that removes less context can nevertheless provide greater computational value when it becomes effective sufficiently early.

Table 1: **Capacity- and speedup-optimal operating points under three evidence-retention thresholds.** Retention columns report $\rho^*(\ell;\tau)$, obtained by piecewise-linear interpolation. Speedups are versus full attention at the speedup- and capacity-optimal depths; parentheses give their ratio computed from unrounded latencies.

Model	τ	Capacity-opt. depth	Retention at cap.	Speedup-opt. depth	Retention at spd.	FastKV speedup: spd./cap.	GemFilter speedup: spd./cap.
Llama	.90	L10	.15	L2	.37	4.61× / 2.61× (**1.76×**)	5.47× / 2.67× (**2.04×**)
Llama	.95	L10	.21	L3	.31	4.42× / 2.48× (**1.78×**)	5.12× / 2.59× (**1.98×**)
Llama	.98	L10	.28	L3	.39	3.89× / 2.39× (**1.63×**)	4.36× / 2.41× (**1.81×**)
Ministral	.90	L12	.28	L2	.48	2.94× / 2.14× (**1.37×**)	3.25× / 2.10× (**1.55×**)
Ministral	.95	L12	.36	L2	.55	2.68× / 2.08× (**1.28×**)	2.89× / 1.92× (**1.51×**)
Ministral	.98	L12	.50	L2	.65	2.36× / 1.97× (**1.20×**)	2.46× / 1.66× (**1.49×**)

3.5 Capacity–Value Separation Depends on Reliable Early Reduction

The separation above is observed under the Full evaluation configuration. We therefore test whether it persists across configurations that expose different amounts of reliable reduction at early Transformer depths. Table 2 summarizes the resulting speedup-optimal and capacity-optimal depths across the three evaluation configurations and evidence-retention thresholds. Removing PDSC changes the minimum reliable retention ratios and can shift the exact speedup-optimal depth, particularly for Llama-3.1-8B. However, the capacity-optimal depths remain L10 for Llama-3.1-8B and L12 for Ministral-8B, while the speedup-optimal depth remains earlier in all six evaluated model–threshold conditions. Capacity–value separation therefore persists without PDSC, although the exact operating points change.

The Post-RoPE, last-query configuration provides a contrasting case in which reliable reduction at early depths is substantially weaker. At $\tau = .95$ and .98, the speedup- and capacity-optimal depths coincide for both models. At $\tau = .90$, they also coincide for Ministral-8B, while Llama-3.1-8B retains a smaller separation, with L3 speedup-optimal and L10 capacity-optimal. Overall, the two optima coincide in five of the six evaluated model–threshold conditions under this configuration.

Taken together, these controls show that early placement alone does not guarantee capacity–value separation. Across the evaluated configurations, separation persists when sufficient reliable reduction is available at early depths and diminishes when reliable early reduction is limited. This observed boundary suggests that an added early reduction is computationally worthwhile only when sufficient context can be removed reliably early enough to outweigh the benefit of waiting for greater later-stage reduction capacity; otherwise, the existing downstream selection path can be retained.

Table 2: **Boundary of capacity–value separation across evaluation configurations.** Entries report $\ell_{\mathrm{spd}}/\ell_{\mathrm{cap}}$; unequal depths indicate separation and equal depths coincident optima. Full reliability-matched retention results are in Appendix B.3.

Model	Configuration	$\tau = .90$	$\tau = .95$	$\tau = .98$
Llama-3.1-8B	Full	L2 / L10	L3 / L10	L3 / L10
Llama-3.1-8B	without PDSC	L2 / L10	L2 / L10	L4 / L10
Llama-3.1-8B	Post-RoPE last-query	L3 / L10	L10 / L10	L10 / L10
Ministral-8B	Full	L2 / L12	L2 / L12	L2 / L12
Ministral-8B	without PDSC	L2 / L12	L2 / L12	L2 / L12
Ministral-8B	Post-RoPE last-query	L12 / L12	L12 / L12	L12 / L12

4 Downstream Performance Sensitivity

The reliability-matched analysis in Section 3 identifies computationally valuable first context-reduction depths under a controlled evidence-retention requirement. We next examine how sensitive downstream performance is to changes in early-layer depth and retention, and whether the observed behavior is confined to a single precisely tuned operating point.

We evaluate the same 3×3 LongBench grid for Llama-3.1-8B and Ministral-8B, varying the first context-reduction depth over $\ell \in \{\mathrm{L2}, \mathrm{L3}, \mathrm{L4}\}$ and the token-retention ratio over $\rho \in \{.60, .70, .80\}$. At each operating point, an added first context-reduction stage precedes an otherwise unchanged FastKV or GemFilter downstream selection procedure. The added reduction is activated only for prompts longer than 8192 tokens; shorter prompts follow the corresponding downstream-only execution path.

Table 3 summarizes the score-change ranges across both models, with complete model-specific grids and category-level results reported in Appendix D. Across both models, FastKV score changes range from -1.46 to -0.17, whereas GemFilter changes range from -1.01 to $+0.01$ relative to their corresponding downstream-only baselines. Higher retention generally yields smaller score degradation, although the pattern is not uniformly monotonic across depths and downstream settings. Overall, aggregate performance remains relatively close to the corresponding baselines across multiple early-layer depth–retention settings.

These results suggest that the observed downstream behavior is not confined to a single finely tuned early-layer operating point. The LongBench grid remains a local sensitivity analysis rather than a reliability-matched optimization: its operating points are not selected using a common NIAH threshold and therefore do not redefine ℓ_{cap} or ℓ_{spd}.

Table 3: **LongBench downstream sensitivity across models.** Score-change ranges over the 3×3 early-layer depth–retention grid are relative to each downstream-only baseline. Complete model-specific grids and category-level results are reported in Appendix D.

Model	FastKV baseline	FastKV change	GemFilter baseline	GemFilter change
Llama-3.1-8B	47.53	-1.37 to -0.17	41.65	-0.69 to $+0.01$
Ministral-8B	49.34	-1.46 to -0.35	45.87	-1.01 to -0.08

5 Related Work

Long-context inference has been accelerated through active-context reduction, adaptive or progressive token pruning, sparse attention, KV-cache management, prompt compression, and reduced prompt-depth computation. These approaches optimize different computational objects, so a reported Transformer depth may denote a relevance-scoring point, an in-path sequence-reduction point, an adaptive pruning decision, or a depth after which prompt computation is reduced. We therefore distinguish prior work by what is reduced, when reduction becomes effective, and how execution proceeds afterward. Appendix E provides a detailed comparison of reported reduction depths and execution semantics.

5.1 Depth-Dependent Context Selection and Reduction

Several methods exploit depth-dependent token relevance or reduce the active context at intermediate layers. GemFilter uses an intermediate representation to select a reduced raw-token subsequence that is subsequently re-encoded, whereas FastKV begins token-selective propagation (TSP) at an in-path TSP layer and continues with selected hidden states (Shi et al., 2026; Jo et al., 2026). PromptDistill performs query-conditioned intermediate retention, while SlimInfer progressively prunes active hidden states across depth (Jin et al., 2026; Long et al., 2026). Thus, intermediate or multi-stage context reduction itself is not our contribution.

Other approaches make reduction adaptive or aggregate information across layers. ASL selects pruning depth according to input-dependent ranking behavior, LazyLLM progressively varies token participation with possible later recovery, and DASH applies token-specific halting based on representation stability (Taniguchi et al., 2026; Fu et al., 2024; Chen et al., 2026). CLAA aggregates relevance signals across layers, while PyramidInfer and PyramidKV allocate retained context or KV capacity nonuniformly with depth (McDanel et al., 2026; Yang et al., 2024; Cai et al., 2025).

For Llama-3.1-8B, our capacity-optimal depth (L10) lies numerically near the reported operating depths of GemFilter (L13) and FastKV (L15), but this proximity is descriptive rather than causal because these depths correspond to different operations and selection criteria.

Our analysis asks a different question: under the same evidence-retention requirement, does the depth supporting the greatest reliable reduction capacity also minimize measured prefill latency? This reliability-matched comparison explicitly separates reliable reduction capacity from computational value.

5.2 Complementary Long-Context Efficiency Techniques

Complementary approaches reduce other components of long-context cost. Sparse-prefill methods reduce attention computation without necessarily shortening the active hidden-state sequence (Jiang et al., 2024a). KV-oriented methods instead compress, retrieve, or allocate cached context through token-, head-, layer-, or query-dependent mechanisms (Li et al., 2024; Fu et al., 2025; Feng et al., 2025; Hao et al., 2025). Prompt-compression methods shorten textual input before ordinary Transformer execution (Jiang et al., 2023; 2024b; Pan et al., 2024; Li et al., 2023). Other approaches reduce the amount of Transformer depth applied to prompt tokens rather than the number of active tokens (Qiao et al., 2025; Oh et al., 2026; He et al., 2026). These directions are complementary to our setting, which isolates when the active context should first become shorter under a common reliability requirement.

6 Discussion and Conclusion

Under a common reliability requirement, our results show that reliable reduction capacity and computational value can favor different first context-reduction depths. Crucially, the speedup-optimal operating points retain more context than their capacity-optimal counterparts yet achieve lower prefill latency. Their advantage therefore comes from acting earlier rather than from stronger reduction, showing that maximizing reliable reduction capacity need not minimize measured prefill latency.

Across the evaluated configurations, this separation persists when sufficient reliable reduction is available at early depths and diminishes when early reliability is limited. This boundary suggests treating an added early reduction as conditional: when sufficient context can be removed reliably, a conservative early reduction can reduce intervening computation while preserving later selection based on richer representations; otherwise, the existing downstream selection path can be retained. LongBench provides complementary local sensitivity evidence: aggregate performance remains relatively close to downstream-only baselines across multiple early-layer settings, although category-level responses are more heterogeneous.

The present evidence is limited to two 8B-scale models. NIAH measures evidence survival rather than final-answer accuracy, and the LongBench grid probes only a local operating region. We therefore do not claim a universal optimal early depth or retention threshold. Overall, these results motivate treating the first context-reduction point as a separate, conditional optimization decision from later selection or pruning stages.

AI USE STATEMENT

To aid or polish writing. Generative AI was used to assist with preliminary drafting, translation, refinement of key arguments, paragraph organization, wording, and language polishing. All AI-assisted text was reviewed and revised by the authors.

For retrieval and discovery. Generative AI was used to assist in identifying potentially relevant literature and in organizing downloaded papers to support subsequent reading and discussion. The authors independently examined the literature and assessed its relevance.

For research ideation or execution. The research ideas, questions, and methodological direction were developed by the authors. AI was used to assist with experimental planning and discussion. All implementation, experimental execution, and data collection were carried out by the authors. AI also assisted with organizing, analyzing, and presenting experimental results, with all analyses and interpretations reviewed and verified by the authors.

To draft sections of the paper. The authors specified the research problems, objectives, key points, and experimental findings. AI was used to assist with drafting section outlines, organizing manuscript structure, planning content allocation, and condensing material.

Generative AI was additionally used to assist with data organization, preparation of figures and statistical visualizations, consistency checks for terminology and notation, and review of the manuscript for clarity, rigor, correctness, and completeness. All AI-assisted outputs were reviewed and revised by the authors, who retain full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce our results in the appendix. Table A1 lists the model revisions, candidate reduction depths, retention-ratio grid, evidence-retention thresholds, downstream execution settings, and hardware. Appendix A.2 describes the controlled Needle-in-a-Haystack construction, Appendix A.3 specifies every component and hyperparameter of the Full evaluation configuration and its controls, and Appendix A.4 details the latency measurement protocol. Complete capacity and latency profiles, repeated timing measurements, and full LongBench results are reported in Appendices C and D.

Source code, configuration files, and evaluation scripts are not included in the submission; if the paper is accepted, we will release them publicly in the camera-ready version.

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.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Yucheng Li, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. In *Second Conference on Language Modeling*, 2025.

Yujie Chen, Tailai Chen, Yifeng Gao, Zoe Wanying He, Yijue Xu, Shaobo Wang, and Linfeng Zhang. Stability implies redundancy: Delta attention selective halting for efficient long-context prefilling. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26830–26848, 2026.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*, 2024.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16344–16359, 2022.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S. Kevin Zhou. Ada-KV: Optimizing KV cache eviction by adaptive budget allocation for efficient LLM inference. In *Advances in Neural Information Processing Systems*, volume 38, pp. 125462–125498, 2025.

Qichen Fu, Minsik Cho, Thomas Merth, Sachin Mehta, Mohammad Rastegari, and Mahyar Najibi. LazyLLM: Dynamic token pruning for efficient long context LLM inference, 2024. arXiv:2407.14057 [cs.CL].

Yao Fu. Challenges in deploying long-context transformers: A theoretical peak performance analysis, 2024. arXiv:2405.08944 [cs.LG].

Yu Fu, Zefan Cai, Abedelkadir Asi, Wayne Xiong, Yue Dong, and Wen Xiao. Not all heads matter: A head-level KV cache compression method with integrated retrieval and reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao

Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. arXiv:2407.21783 [cs.AI].

Jitai Hao, Yuke Zhu, Tian Wang, Jun Yu, Xin Xin, Bo Zheng, Zhaochun Ren, and Sheng Guo. OmniKV: Dynamic context selection for efficient long-context LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025.

Junhui He, Zhihui Fu, Jun Wang, and Qingan Li. POP: Prefill-only pruning for efficient large model inference. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 5864–5877, 2026.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376, 2023.

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, 2024a.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, 2024b.

Weisheng Jin, Maojia Song, Tej Deep Pala, Yew Ken Chia, Dorien Herremans, and Soujanya Poria. PromptDistill: Query-based selective token retention in intermediate layers for efficient large language model inference. In *Proceedings of the 19th International Natural Language Generation Conference*, 2026.

Dongwon Jo, Jiwon Song, Yulhwa Kim, and Jae-Joon Kim. FastKV: Decoupling of context reduction and KV cache compression for prefill-decoding acceleration. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 32167–32186, 2026.

Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. FlexPrefill: A context-aware sparse attention mechanism for efficient long-sequence inference. In *The Thirteenth International Conference on Learning Representations*, 2025.

Mo Li, Songyang Zhang, Taolin Zhang, Haodong Duan, Yunxin Liu, and Kai Chen. NeedleBench: Evaluating LLM retrieval and reasoning across varying information densities. *Transactions on Machine Learning Research*, 2025.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6342–6353, 2023.

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*, volume 37, pp. 22947–22970, 2024.

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

Lingkun Long, Rubing Yang, Yushi Huang, Desheng Hui, Ao Zhou, and Jianlei Yang. SlimInfer: Accelerating long-context LLM inference via dynamic token pruning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(38):32284–32292, 2026. ISSN 2374-3468.

Bradley McDanel, Steven Li, and Harshit Khaitan. CLAA: Cross-layer attention aggregation for accelerating LLM prefill, 2026. arXiv:2602.16054 [cs.CL].

Jungsuk Oh, Hyeseo Jeon, Hyunjune Ji, Kyongmin Kong, and Jay-Yoon Lee. Shallow prefill, deep decoding: Efficient long-context inference via layer-asymmetric KV visibility, 2026. arXiv:2605.06105 [cs.AI].

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 963–981, 2024.

Aurick Qiao, Zhewei Yao, Samyam Rajbhandari, and Yuxiong He. SwiftKV: Fast prefill-optimized inference with knowledge-preserving model transformation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25734–25753, 2025.

Zhenmei Shi, Yifei Ming, Xuan-Phi Nguyen, Yingyu Liang, and Shafiq Joty. Discovering the gems in early layers: Accelerating long-context LLMs with 1000x input token reduction. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 13839–13857, 2026.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. ISSN 0925-2312.

Rei Taniguchi, Yuyang Dong, Makoto Onizuka, and Chuan Xiao. Adaptive layer selection for layer-wise token pruning in LLM inference. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 967–986, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024.

Dongjie Yang, Xiaodong Han, Yan Gao, Yao Hu, Shilin Zhang, and Hai Zhao. PyramidInfer: Pyramid KV cache compression for high-throughput LLM inference. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3258–3270, 2024.

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, pp. 76397–76413, 2025.

A Experimental Setup and Reproducibility

This appendix provides the implementation and evaluation details needed to reproduce the reliability-matched depth analysis in Sections 2–3 and the downstream sensitivity analysis in Section 4. We first summarize the model-specific experimental settings and evaluated operating ranges (Appendix A.1), then describe the controlled Needle-in-a-Haystack construction and reliability-matching procedure (Appendix A.2). Appendix A.3 specifies the Full evaluation configuration and its controls, and Appendix A.4 details the latency measurement protocol and speedup definitions.

Unless otherwise stated, the same model-specific evaluation procedure is held fixed across candidate first context-reduction depths. This allows differences in reliable reduction capacity and measured prefill latency to be attributed to the evaluated depth and its corresponding reliability-matched operating point rather than to changes in the surrounding evaluation configuration.

A.1 Experimental Configurations

Table A1 summarizes the principal experimental settings. Unless otherwise stated, the same model-specific configuration is used throughout the corresponding experiments so that candidate first context-reduction depths are compared under a common evaluation procedure.

Table A1: Experimental configurations. Candidate first context-reduction depths define the set $\mathcal{D}$ over which the capacity-optimal and speedup-optimal depths are identified. The wider diagnostic depth range is used only for the evidence-retention analyses in Appendix B and does not enlarge $\mathcal{D}$.

Item	Llama-3.1-8B	Ministral-8B
Model revision (Hugging Face snapshot)	`meta-llama/Meta-Llama-3.1-8B-Instruct 0e9e39f2`	`mistralai/Ministral-8B-Instruct-2410 2f494a19`
Transformer layers / hidden size / KV heads	32 / 4096 / 8	36 / 4096 / 8
Candidate first context-reduction depths $\mathcal{D}$	L2–L12	L2–L16
Diagnostic evidence-retention depths	L0–L25	L0–L25
Token-retention-ratio grid	$\{1, .9, .8, .7, .6, .5, .4, .3, .2, .156, .1, .05, .02\}$	same
Evidence-retention thresholds τ	$\{.90, .95, .98\}$	same
Controlled NIAH instances	100	100
Controlled NIAH background length	100K tokens after model-specific tokenization	same
Controlled NIAH full prompt length	$\approx$ 100,066 tokens	$\approx$ 100,067 tokens
Full evaluation configuration	Pre-RoPE + Query Aggregation + PDSC + Blocking + Head/Tail Retention (Table A2)	same
Principal latency input	128K-token background + one fixed question	same
FastKV (TSP layer / TSP rate / KV-retain rate)	15 / 0.2 / 0.1	17 / 0.2 / 0.1
GemFilter (selection layer / k)	13 / 1024	17 / 1024
Batch size / precision / attention backend	1 / fp16 / FlashAttention-2	same
Hardware, latency	NVIDIA RTX A6000 48 GB	same
Hardware, evidence-retention evaluation	NVIDIA RTX PRO 4500 32 GB	same
Hardware, LongBench evaluation	NVIDIA RTX PRO 4500 32 GB	same
Prompting, LongBench	model-specific chat template	same
Prompting, controlled NIAH	no chat template	same

The candidate-depth set $\mathcal{D}$ is restricted to first context-reduction depths that precede the downstream selection stage in both evaluated execution settings. This yields L2–L12 for Llama-3.1-8B and L2–L16 for Ministral-8B. Evidence retention is additionally evaluated over the wider L0–L25 range for diagnostic analysis. These additional depths are not included when determining either $\ell_{\mathrm{cap}}(\tau)$ or $\ell_{\mathrm{spd}}(\tau)$.

The evaluated token-retention-ratio grid provides the measured evidence-retention samples from which the reliability-matched operating points are estimated. For each candidate depth ℓ and evidence-retention threshold τ, the minimum reliable retention ratio $\rho^*(\ell;\tau)$ is obtained from piecewise-linear interpolation of the measured evidence-retention values between adjacent retention ratios, as defined in Section 2.2 and detailed in Appendix A.2. Consequently, $\rho^*(\ell;\tau)$ need not coincide with a sampled value in the retention-ratio grid. These interpolated reliability-matched operating points are used throughout the principal capacity–value analysis.

For the principal latency experiments, the 128K setting refers to a 128K-token background followed by one fixed question, so the total timed prompt length is slightly greater than 128K tokens. Full timing boundaries and measurement procedures are given in Appendix A.4.

A.2 Controlled NIAH Construction

We use a controlled Needle-in-a-Haystack (NIAH) protocol to measure whether designated query-relevant evidence survives context reduction, following long-context evaluations based on controlled evidence placement (Liu et al., 2024; Li et al., 2025). The purpose of this protocol is to provide a common evidence-retention criterion for comparing candidate first context-reduction depths rather than to measure downstream answer accuracy.

We use the Paul Graham essay collection as the source background for both models. For each model, the concatenated source text is tokenized using that model's tokenizer and truncated to a 100K-token background before evidence insertion. A single evidence sentence,

> *The best thing to do in San Francisco is eat a sandwich and sit in Dolores Park on a sunny day.*

is inserted at a controlled position. The evidence occupies one contiguous span. The resulting background is wrapped using the fixed diagnostic template

> *This is a very long story book: <book> … </book>. Based on the content of the book, Question: What is the best thing to do in San Francisco? Answer:*

with no chat template applied. After evidence insertion and prompt wrapping, the resulting prompt contains approximately $100{,}066$ tokens for Llama-3.1-8B and $100{,}067$ tokens for Ministral-8B.

The evaluation contains 100 controlled instances generated using the same stratified procedure for both models. The 100K-token background, excluding the prompt wrapper, is divided into 10 position bins, with 10 instances assigned to each bin. Evidence insertion positions are sampled uniformly within each bin using random seed 0. This provides broad positional coverage while avoiding concentration at a small number of fixed locations.

To vary the local document context surrounding the inserted evidence, the source-essay ordering is also varied across the 100 instances. We organize the construction into 10 repeats, indexed by $r = 0, \ldots, 9$, with one instance from each position bin in every repeat. Repeat $r = 0$ uses the canonical essay ordering, whereas repeats $r = 1, \ldots, 9$ use essay permutations generated using r as the permutation seed. The same essay ordering is shared by the 10 position-bin instances within a repeat. The source corpus, evidence sentence, query, position-bin construction, and randomization procedure are shared across the two models, while tokenization and final prompt construction are performed separately using each model's tokenizer.

For a candidate first context-reduction depth ℓ and token-retention ratio ρ, evidence retention is defined as the fraction of tokens in the inserted evidence span that remain in the retained set, averaged over the 100 controlled instances. We denote this quantity by $E(\ell, \rho)$. Within each model, the same controlled instances are reused across candidate depths, retention ratios, and evaluation configurations. For each instance, relevance scores for all evaluated depths and readouts are obtained from the same underlying forward-pass data, and the evaluated retention ratios are subsequently applied to the corresponding recorded scores. Consequently, within-model differences in $E(\ell, \rho)$ reflect the evaluated first context-reduction depth, retention ratio, and relevance-scoring/selection configuration rather than differences in test instances. The bootstrap intervals reported in Appendix B therefore use paired resampling of these shared instances.

Let $\mathcal{R}$ denote the evaluated token-retention-ratio grid and let $\rho_{\min}$ denote its smallest value. Because an evidence-retention threshold can be crossed between two adjacent sampled ratios, we estimate

the minimum reliable retention ratio using piecewise-linear interpolation of the measured evidence-retention values. Let $\widetilde{E}(\ell, \rho)$ denote the piecewise-linear interpolation of $E(\ell, \rho)$ over the evaluated retention-ratio grid at depth ℓ. For an evidence-retention threshold τ, the reliability-matched operating point is defined as

$$\rho^*(\ell; \tau) = \min \left\{ \rho \in [\rho_{\min}, 1] : \widetilde{E}(\ell, \rho) \geq \tau \right\}. \tag{A.1}$$

Interpolation is performed independently at each candidate depth. When the threshold crossing lies between two adjacent evaluated retention ratios, the corresponding interpolated crossing point is used as $\rho^*(\ell; \tau)$. Consequently, the reliability-matched operating point need not coincide with a sampled value in $\mathcal{R}$. If $\widetilde{E}(\ell, \rho)$ does not reach τ anywhere over the evaluated interval $[\rho_{\min}, 1]$, we set

$$\rho^*(\ell; \tau) = 1.$$

A smaller $\rho^*(\ell; \tau)$ indicates greater reliable reduction capacity because a larger fraction of the context can be removed while satisfying the same evidence-retention requirement. These interpolated reliability-matched operating points are used in the principal capacity–value analysis and in determining $\ell_{\mathrm{cap}}(\tau)$ and $\ell_{\mathrm{spd}}(\tau)$.

Throughout the paper, *reliable* refers specifically to satisfying the controlled evidence-retention criterion used in this analysis. The NIAH protocol measures whether designated evidence survives context reduction; it should not be interpreted as downstream answer accuracy or as a general guarantee of downstream performance.

A.3 Full Evaluation Configuration

The principal depth-wise experiments use a fixed Full evaluation configuration so that differences across candidate first context-reduction depths are not confounded by changes in the relevance-scoring or token-selection procedure. The Full configuration consists of Pre-RoPE relevance scoring, Query Aggregation, Position-Dependent Score Calibration (PDSC), Blocking, and Head/Tail Retention. These components define a common experimental setting for cross-depth comparison; they are not presented as a separate context-reduction method or as an optimized token-selection mechanism.

For reproducibility, we describe the scoring and selection operations below. At a candidate first context-reduction depth ℓ, let n denote the number of context tokens, N_h the number of attention heads, and d_h the per-head dimension. Let $q_{h,i}^{(\ell)}$ and $k_{h,j}^{(\ell)}$ denote the query and key representations for attention head h, query position i, and context position j, respectively. Unless otherwise stated, these representations are taken before RoPE is applied, and all scoring and selection hyperparameters described below are held fixed across candidate depths.

A.3.1 Pre-RoPE Relevance Scoring and Query Aggregation

The Full evaluation configuration computes token relevance from query–key similarity before rotary positional encoding (RoPE) is applied (Su et al., 2024). In particular, the relevance readout uses the pre-RoPE query and key representations introduced above, rather than query–key similarity after the rotary positional transformation. This keeps the relevance signal used for token selection separate from the relative-position transformation introduced by RoPE.

Query Aggregation uses multiple recent query positions rather than only the final query position. Let $\mathcal{Q}$ denote the set of the final $w = 32$ query positions. For each attention head h, we first average their query representations:

$$\bar{q}_h^{(\ell)} = \frac{1}{|\mathcal{Q}|} \sum_{i \in \mathcal{Q}} q_{h,i}^{(\ell)}. \tag{A.2}$$

The raw relevance score assigned to context token j is then obtained by summing query–key similarity across attention heads:

$$s_j^{(\ell)} = \sum_{h=1}^{N_h} \left(\bar{q}_h^{(\ell)} \right)^\top k_{h,j}^{(\ell)}. \tag{A.3}$$

The dot product is left unscaled. The conventional $1/\sqrt{d_h}$ attention scaling factor is omitted because it is constant across context positions and therefore does not affect their ranking.

Before positional calibration, the resulting scalar score sequence is smoothed using one-dimensional average pooling with kernel size 5, stride 1, and padding 2, so that the sequence length is unchanged. This local smoothing is treated as an implementation detail of the relevance readout rather than as a separate experimental factor.

A.3.2 Position-Dependent Score Calibration

The scalar relevance-score sequence obtained after query aggregation and local smoothing can contain a slowly varying position-dependent component. Position-Dependent Score Calibration (PDSC) removes a locally estimated baseline from this sequence before token selection. For notational simplicity, let s_j denote the smoothed scalar relevance score at context position j. We estimate the local baseline as

$$\hat{b}_j = \frac{1}{|\mathcal{N}_j|} \sum_{u \in \mathcal{N}_j} s_u, \tag{A.4}$$

where $\mathcal{N}_j$ is a window of W positions centered around j and truncated at the sequence boundaries. The calibrated score is then

$$\tilde{s}_j = s_j - \hat{b}_j. \tag{A.5}$$

PDSC therefore measures the relevance of token j relative to the local score baseline of the current input rather than relative to a globally fixed calibration curve. Operationally, it acts as input-adaptive local detrending of the Pre-RoPE scalar relevance-score sequence.

In our implementation, the window width scales with the context length:

$$W = \left\lfloor \frac{n}{10} \right\rfloor. \tag{A.6}$$

More precisely, the local window is

$$\mathcal{N}_j = \{u : j - \lfloor W/2 \rfloor \le u \le j - \lfloor W/2 \rfloor + W - 1\} \cap \{1, \dots, n\}. \tag{A.7}$$

The moving average is implemented in linear time using cumulative sums. At the two sequence boundaries, the window is truncated rather than zero-padded.

PDSC is treated only as one component of the fixed Full evaluation configuration. We do not interpret it as a proposed positional-debiasing method or assume that capacity–value separation depends on this particular calibration mechanism. Section 3.5 and Appendix B.3 therefore include a without-PDSC control in which the calibration step is removed while the remaining relevance-scoring and token-selection settings are held unchanged.

A.3.3 Head/Tail Retention

After relevance scoring and, when applicable, positional calibration, tokens near the beginning and end of the context are protected from score-based selection. Let

$$\mathcal{H}_{\text{head}} = \{1, \dots, f\}, \qquad \mathcal{T}_{\text{tail}} = \{n - t + 1, \dots, n\} \tag{A.8}$$

denote the head and tail regions, respectively. Tokens in $\mathcal{H}_{\text{head}} \cup \mathcal{T}_{\text{tail}}$ are retained unconditionally and excluded from the score-based ranking of the remaining interior context.

In the principal configuration, we use

$$f = t = 256, \tag{A.9}$$

so that 256 tokens are protected at each boundary. The same boundary-retention setting is used throughout the capacity–value experiments, including the Full, without-PDSC, and Post-RoPE last-query configurations. Thus, differences among these configurations are not induced by changes in the number of protected boundary tokens.

Head/Tail Retention is treated as a fixed selection control, rather than as part of the relevance-scoring definition. The tail region protects the query-side boundary, including the fixed query used in the controlled NIAH evaluation, from competing directly with the interior context under score-based selection. The head region is likewise protected to avoid confounding the depth comparison with known boundary effects at the beginning of long sequences, including attention-sink behavior in which initial tokens can receive disproportionate attention (Xiao et al., 2024). Both protected regions are removed from the interior ranking before block-based selection is applied.

A.3.4 Blocking and Block-Level Selection

After Head/Tail Retention, the remaining interior context is partitioned into contiguous blocks rather than ranked independently token by token. Let $B_1, \ldots, B_M$ denote consecutive interior blocks of size b, except that the final block may contain fewer than b tokens. Let r_j denote the relevance score passed to the selection stage at context position j. Under the Full configuration, $r_j = \tilde{s}_j$, where $\tilde{s}_j$ is the PDSC-calibrated score defined in Eq. A.5. The score of block B_r is the mean score of its constituent tokens:

$$a_r = \frac{1}{|B_r|} \sum_{j \in B_r} r_j. \tag{A.10}$$

To reduce sensitivity to small shifts in where a high-scoring span crosses a block boundary, we additionally pool block scores over a local neighborhood. With pooling radius p,

$$g_r = \max_{u \,:\, |u-r| \le p} a_u. \tag{A.11}$$

Blocks are ranked according to g_r, and selected blocks are retained in their entirety. For a target token-retention ratio ρ, the nominal retained-token budget is

$$m = \lfloor \rho n \rfloor. \tag{A.12}$$

After reserving $f + t$ tokens for Head/Tail Retention, the number of score-selected interior blocks is

$$K = \left\lfloor \frac{m - f - t}{b} \right\rfloor. \tag{A.13}$$

Because selection operates on complete blocks and the final interior block may be partial, the realized retained-token count can differ slightly from the nominal budget m due to blockwise rounding. If the interior context length $n - f - t$ is not divisible by b, the remaining tokens at the end of the interior region are treated as a final partial block. After block selection, the retained token indices are restored to their original context order before subsequent Transformer computation.

In all principal capacity–value experiments, we use

$$b = 8, \qquad p = 2, \tag{A.14}$$

together with Head/Tail Retention of $f = t = 256$ tokens. These block-selection settings are held fixed across the Full, without-PDSC, and Post-RoPE last-query configurations. The generic notation r_j allows the same selection procedure to be applied across these controls: under the Full configuration, r_j is the PDSC-calibrated score; under the without-PDSC configuration, the calibration step is omitted before block scoring; and under the Post-RoPE last-query configuration, the corresponding smoothed Post-RoPE relevance score is used. Thus, the controls alter the relevance signal supplied to selection while keeping Head/Tail Retention, Blocking, block-neighborhood pooling, and output ordering unchanged.

A.3.5 Configuration Summary and Controls

Table A2 summarizes the Full evaluation configuration. Across candidate first context-reduction depths, the relevance readout, score calibration, smoothing, and token-selection settings are held fixed. In the principal reliability-matched analysis, the token-retention ratio used at each candidate depth is $\rho^*(\ell; \tau)$, estimated from the full retention-ratio sweep by the piecewise-linear interpolation procedure defined in Section 2.2 and Appendix A.2. Thus, candidate depths are compared under the same evidence-retention requirement while the surrounding evaluation procedure remains unchanged.

Table A2: Full evaluation configuration.

Component	Setting	Role
RoPE position for relevance scoring	Pre-RoPE	Relevance readout
Query aggregation	Final 32 query positions	Relevance readout
Cross-head aggregation	Sum across attention heads	Relevance readout
Local score smoothing	Average pooling, kernel size 5	Readout implementation
PDSC	Local baseline subtraction	Score calibration
PDSC baseline width	$W = \lfloor n/10 \rfloor$	Score calibration
Head retention	256 tokens	Selection control
Tail retention	256 tokens	Selection control
Blocking	$b = 8$ tokens	Selection control
Block-neighborhood pooling	Max pooling, radius $p = 2$ blocks	Selection control
Output ordering	Original context order	Selection control
Activation threshold	$n > 8192$ (LongBench only)	Evaluation protocol

For the LongBench downstream-sensitivity analysis, the added first context-reduction stage is activated only for prompts longer than 8192 tokens. Shorter prompts are processed using the corresponding downstream execution setting alone. This activation rule applies only to the LongBench evaluation; the controlled NIAH and 128K latency experiments exceed this threshold throughout.

The LongBench operating points are deliberately distinct from the reliability-matched operating points used in the principal capacity–value analysis. For LongBench, we independently evaluate the local grid

$$\ell \in \{\mathrm{L2}, \mathrm{L3}, \mathrm{L4}\}, \qquad \rho \in \{.60, .70, .80\},$$

without selecting these operating points according to a common NIAH evidence-retention threshold. The LongBench grid therefore characterizes local downstream-performance sensitivity and does not redefine $\rho^*(\ell;\tau)$, $\ell_{\mathrm{cap}}(\tau)$, or $\ell_{\mathrm{spd}}(\tau)$.

The added first context reduction is integrated into the two downstream execution settings using their respective parameterizations. On the GemFilter path, the retained-token budget is expressed as

$$\left\lfloor \frac{n}{r} \right\rfloor, \qquad r = \frac{1}{\rho},$$

whereas on the FastKV path, the added first reduction is implemented as an additional token-selective-propagation stage with retention rate ρ (Shi et al., 2026; Jo et al., 2026). These parameterizations target the same nominal token-retention ratio. Under the Full configuration, the realized retained-token count is subject to the blockwise rounding described in Appendix A.3.4. The original downstream FastKV or GemFilter selection stage otherwise retains its specified location and configuration.

Two alternative configurations are used as controls in the boundary analysis. The without-PDSC configuration removes only the calibration step in Appendix A.3.2 while retaining the Pre-RoPE aggregated readout, local score smoothing, Head/Tail Retention, Blocking, and block-neighborhood pooling unchanged. This comparison therefore isolates the effect of PDSC while preserving the remaining relevance-scoring and selection procedure.

The Post-RoPE last-query configuration replaces the Pre-RoPE aggregated readout in Appendix A.3.1 with query–key similarity computed after RoPE using only the final query position. PDSC is disabled, while the same local score smoothing, Head/Tail Retention, Blocking, and block-neighborhood pooling are retained. This readout corresponds to the Post-RoPE final-query relevance signal used by GemFilter (Shi et al., 2026), but the surrounding selection controls remain those of the common evaluation configuration. It is therefore a controlled relevance-readout variant rather than a reproduction of the complete GemFilter filtering procedure.

The Full, without-PDSC, and Post-RoPE last-query configurations are used to examine how changes in the amount of reliable reduction available within the first few Transformer layers affect the observed capacity–value relationship. They serve as diagnostic and boundary controls rather than as competing context-reduction methods.

A.4 TIMING PROTOCOL

All reported latency measurements are performed on an NVIDIA RTX A6000 48 GB under a common hardware and timing protocol. The timing input consists of a Paul Graham background truncated to 128K tokens, followed by one question; the total prompt length therefore slightly exceeds 128K tokens. Tokenization and CPU-side preprocessing are completed before timing and are excluded from the measured interval. GPU execution is synchronized immediately before and after each timed interval using `torch.cuda.synchronize()`, and elapsed wall-clock time is recorded.

Each evaluated arm uses a fresh model load. The full-attention baseline is measured first in a separate load. For the principal depth sweeps reported in Figure 3 and Tables C2a–C2f, each operating point is measured twice and the lower observed latency is reported; no separate warm-up run is used. To assess run-to-run timing variability, the operating points reported in Tables C3a–C3f are measured repeatedly: 25 times at $\tau = .98$ and 10 times at $\tau = .90$ and .95. The corresponding downstream-only reference is measured with the same number of repetitions as its augmented counterparts, while full attention is measured 10 times. For these repeated measurements, we report the median, mean, and 95% t-interval of the mean. Confidence intervals for speedup ratios are obtained from 10,000 bootstrap resamples using the corresponding repeated reference measurements.

The timed interval includes all model computation required by the evaluated execution path. Full attention consists of a single forward pass of the unmodified model.

For FastKV, the downstream-only execution uses a single forward pass with token-selective propagation beginning at its designated TSP layer (Jo et al., 2026). Adding the earlier first context-reduction point preserves the single-pass execution structure and introduces no restart. For both the downstream-only and augmented FastKV paths, timing ends when the forward pass returns the output logits.

GemFilter has different execution semantics. The downstream-only GemFilter path consists of two passes with one restart: the full context is first processed to the designated selection layer, 1024 tokens are selected, and the retained raw-token subsequence is then re-encoded from the beginning of the model (Shi et al., 2026). Adding the earlier first context-reduction point introduces a third pass and a second restart. All passes and restart overheads are included in the measured latency.

The released GemFilter implementation exposes a different timing boundary from the returned-logit boundary used for FastKV. For both the downstream-only and augmented GemFilter paths, the timed interval extends through generation of the first output token and one subsequent decoding step. The additional decoding step contributes less than 0.1% of the measured time and therefore slightly understates the reported GemFilter speedups. This implementation-specific timing boundary is applied consistently within all GemFilter comparisons. Consequently, absolute latency values are not used to compare FastKV and GemFilter directly; candidate first context-reduction depths are compared only within the same downstream execution setting.

Within each downstream execution setting, the downstream-only baseline and its augmented counterpart are evaluated on the same input and under the same timing boundary, with a fresh model load for each arm. They differ only in the presence of the added earlier first context-reduction point. Incremental prefill speedup is defined as

$$S_{\text{inc}} = \frac{T_{\text{downstream}}}{T_{\text{augmented}}}, \tag{A.15}$$

where $T_{\text{downstream}}$ denotes the latency of the corresponding downstream execution setting alone and $T_{\text{augmented}}$ denotes the latency after adding the earlier first context-reduction point. Thus, $S_{\text{inc}} > 1$ indicates that introducing the earlier reduction lowers measured latency relative to the corresponding downstream-only execution.

For completeness, speedup relative to full-attention prefill is defined as

$$S_{\text{full}} = \frac{T_{\text{full}}}{T_{\text{augmented}}}, \tag{A.16}$$

where T_{full} denotes the latency of the unmodified full-attention model under the same input setting.

For the principal depth sweeps, T_{full} is a single per-model reference shared across evidence-retention thresholds and downstream execution settings. The reference is the fastest full-attention measurement observed across the six measurement sessions: 68.08 s for Llama-3.1-8B and 42.64 s for

Ministral-8B. Using one reference per model keeps speedups versus full attention comparable across thresholds and downstream execution settings. The repeated-measurement analysis in Appendix C.3 instead normalizes by its own repeated full-attention reference.

Table A3: Timing protocol.

Item	Setting
GPU / driver	NVIDIA RTX A6000 48 GB / 580.82.07
CUDA / PyTorch / Transformers / FlashAttention	12.1 / 2.4.0 / 4.43.3 / 2.6.3
Precision / attention / batch size	fp16 / FlashAttention-2 / 1
Input	Paul Graham background truncated to 128K tokens, followed by one question; total prompt length slightly exceeds 128K tokens
Excluded from timing	Tokenization and CPU-side preprocessing
Synchronization	`torch.cuda.synchronize()` at both timing boundaries; wall-clock time
Principal depth sweeps	2 measurements per cell; lower latency reported; no separate warm-up
Repeated measurements	25 per cell at $\tau = .98$; 10 per cell at $\tau = .90, .95$; downstream-only repetitions matched to augmented counterparts; 10 for full attention
Reported uncertainty	Median, mean, 95% t-interval; 95% bootstrap interval for speedup ratios (10,000 resamples)
Model-loading policy	Fresh model load for each arm; full-attention baseline measured first in a separate load
Full-attention path	One forward pass of the unmodified model
FastKV path	One forward pass; added first reduction introduces no restart; timing ends at returned logits
GemFilter path	Two passes / one restart downstream-only; three passes / two restarts with added first reduction; timing includes the first output token and one subsequent decoding step
Comparison scope	Candidate depths compared within the same downstream execution setting; no absolute FastKV–GemFilter latency comparison
Incremental speedup	$T_{\text{downstream}}/T_{\text{augmented}}$
Speedup vs. full attention	$T_{\text{full}}/T_{\text{augmented}}$

B Reliability and Configuration Controls

This appendix provides the extended reliability and configuration analyses supporting Sections 3.1 and 3.5. First, we characterize how evidence retention at a fixed token budget varies jointly with first context-reduction depth and the relevance-scoring/selection configuration. Second, we examine how the earliest evaluated depth satisfying a fixed evidence-retention requirement changes as the token-retention ratio varies. Third, we test how configuration-induced changes in early reliable reduction affect the observed capacity–value relationship.

All analyses use the controlled evidence-retention criterion and experimental setup described in Appendix A. Within each comparison, the relevant non-varied components are held fixed so that the effects of reduction depth, retention ratio, and evaluation configuration can be isolated. These analyses are intended to characterize configuration dependence and the empirical boundary of capacity–value separation, rather than to introduce or rank alternative relevance-scoring methods.

B.1 Full Evidence-Retention Profiles Across Depth and Configuration

Figure B1 provides the full depth-wise evidence-retention profiles for Llama-3.1-8B and Ministral-8B across the diagnostic depth range L0–L25 and all five evaluated relevance-scoring/selection configurations. To isolate which tokens are retained from how many tokens are retained, all configurations are compared at the same token-retention ratio, $\rho = .60$. Thus, every configuration receives the same token budget at a given depth. Differences in evidence retention therefore reflect how model representations are read and how the resulting relevance information is converted into the retained token set, rather than differences in retained-context size.

The profiles vary substantially across both Transformer depth and evaluation configuration. At the same reduction depth and token budget, some configurations retain considerably more designated evidence than others. In particular, weak evidence retention under the Post-RoPE, last-query configuration at several first-few-layer depths is not reproduced by all alternative configurations. This

matched-budget comparison therefore shows that poor retention under one configuration does not, by itself, establish that an early-layer representation intrinsically lacks usable relevance information. The first three diagnostic configurations use plain top-ρn selection, whereas the Full and without-PDSC configurations additionally use Head/Tail Retention and Blocking. Accordingly, the Post-RoPE last-query curve in Figure B1 is distinct from the Post-RoPE last-query boundary control in Appendix B.3, which retains the common selection controls.

The depth-wise profiles are also non-monotonic. Additional Transformer processing can improve evidence retention, but deeper reduction depths do not uniformly dominate shallower ones. Consequently, neither representation depth alone nor a single monotonic depth schedule adequately characterizes the reliable reduction exposed by the evaluated configurations.

Taken together, these results support the interpretation in Section 3.1: first-few-layer reliability depends jointly on reduction depth and the relevance-scoring/selection configuration. Figure B1 is therefore not intended as a leaderboard among scoring or selection rules. Its role is to establish that the amount of reliable reduction exposed at an early depth can change materially with the evaluation configuration. For the principal capacity–value analysis, we accordingly fix the Full evaluation configuration across candidate depths and derive the reliability-matched operating points from its complete retention-ratio sweep.

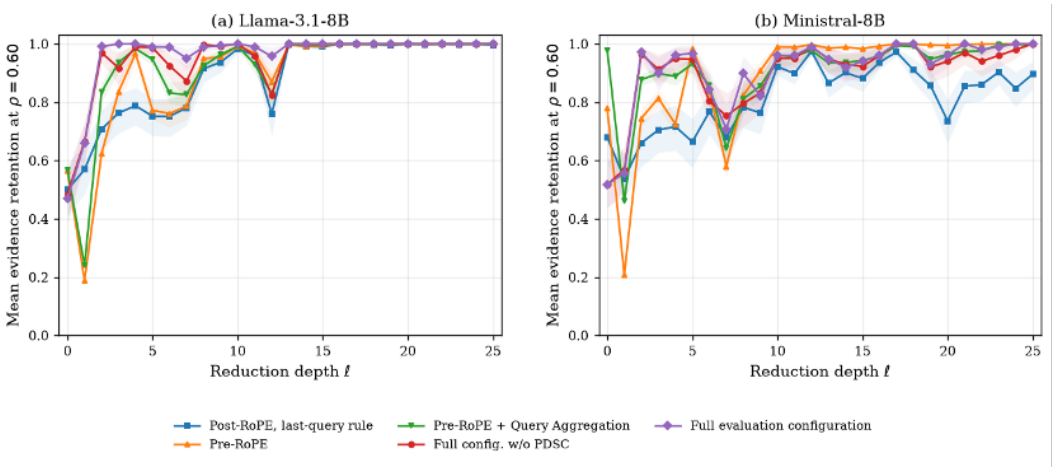


Figure B1: **Full evidence-retention profiles across diagnostic reduction depth and evaluation configuration.** Mean evidence retention is shown at fixed token retention $\rho = .60$ over L0–L25 for both models and all five configurations in Figure 2. All configurations use matched token budgets and identical controlled instances; shaded bands show 95% bootstrap confidence intervals from 1,000 paired resamples of the shared 100-instance evaluation set. The non-monotonic profiles show that evidence retention at a fixed budget depends jointly on reduction depth and relevance-scoring/selection configuration.

B.2 Earliest Reliable Depth Across Retention Ratios

Figure B2 provides a complementary view of the configuration dependence examined in Appendix B.1. For each token-retention ratio ρ on the evaluated grid, we identify the earliest evaluated candidate first context-reduction depth at which mean evidence retention reaches a specified threshold τ. Results are reported for both Llama-3.1-8B and Ministral-8B across $\tau \in \{.90, .95, .98\}$. The candidate-depth ranges are L2–L12 for Llama-3.1-8B and L2–L16 for Ministral-8B.

Across the evaluated configurations, more moderate context reduction can satisfy the same evidence-retention requirement within the first few Transformer layers, whereas more aggressive reduction generally shifts the earliest reliable depth later. The token-retention ratio at which this transition occurs, however, differs substantially across configurations. At the same ρ and τ, some configurations satisfy the requirement within the first few layers, while others require substantially deeper processing.

These profiles reinforce the conclusion of Appendix B.1 that early reliable reduction is not determined by Transformer depth alone. Whether a given amount of reduction becomes reliable early

depends jointly on the retention ratio, candidate reduction depth, and relevance-scoring/selection configuration. Retention ratios for which no evaluated candidate depth reaches the specified threshold are left blank. The resulting profiles characterize the evaluated models and configurations and should not be interpreted as defining a universal ordering of Transformer depths.

Figure B2 is diagnostic and does not define the reliability-matched operating points used in the principal capacity–value analysis. Here, ρ is restricted to the evaluated retention-ratio grid and the analysis asks for the earliest evaluated depth satisfying $E(\ell, \rho) \geq \tau$. In contrast, the principal analysis fixes ℓ and estimates the minimum reliable retention ratio $\rho^*(\ell; \tau)$ by interpolation over the complete retention-ratio sweep. The two views therefore summarize the same underlying reliability measurements from complementary directions.

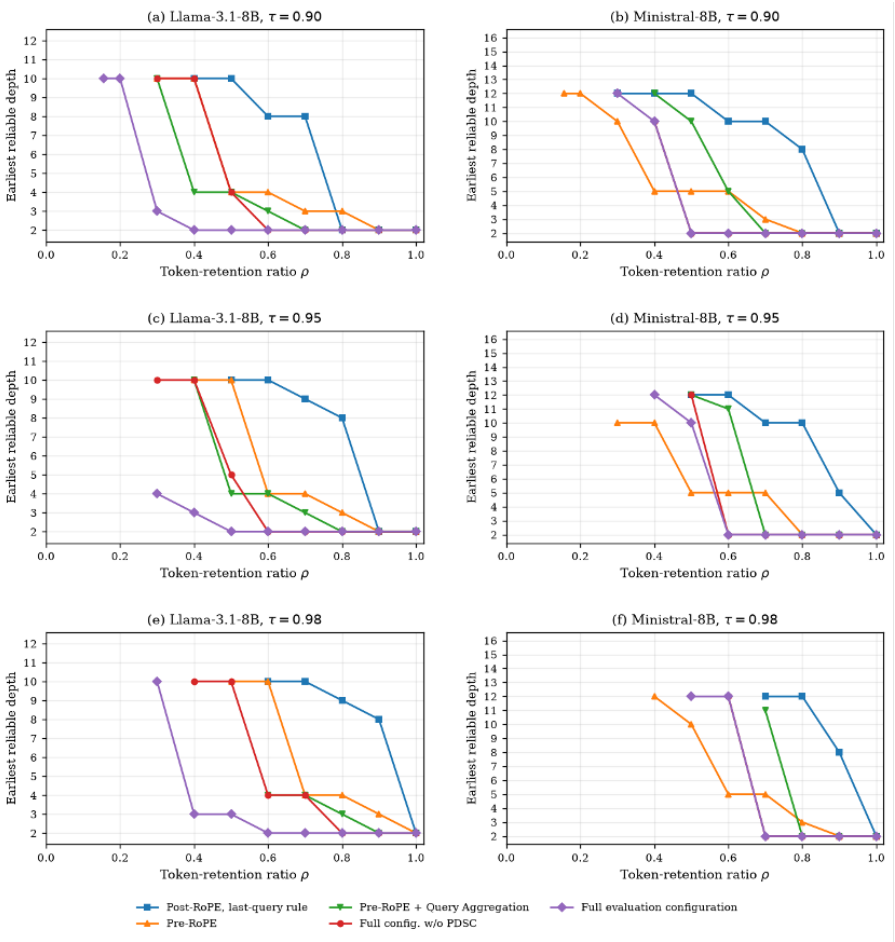


Figure B2: **Earliest evaluated first context-reduction depth satisfying each evidence-retention requirement across token-retention ratios and evaluation configurations.** Results are shown for $\tau \in \{.90, .95, .98\}$. More aggressive reduction generally shifts the earliest reliable depth later, but the relationship depends strongly on the evaluation configuration. Blank entries indicate that no evaluated candidate depth satisfies the specified threshold.

B.3 Configuration Sensitivity and Boundary Cases

The principal analysis establishes capacity–value separation under the Full evaluation configuration. We next ask whether this separation follows automatically from placing the first context reduction early, or whether it changes with the amount of reliable reduction available within the first few Transformer layers.

We consider two controls defined in Appendix A.3. The without-PDSC control removes only the position-dependent score-calibration step while leaving the Pre-RoPE aggregated readout and remaining selection controls unchanged. The Post-RoPE last-query control replaces the Pre-RoPE

aggregated readout with query–key similarity computed after RoPE using only the final query position. PDSC is disabled, while local score smoothing, Head/Tail Retention, Blocking, and block-neighborhood pooling remain unchanged. This readout corresponds to the Post-RoPE final-query relevance signal used by GemFilter (Shi et al., 2026), but the surrounding selection controls remain those of the common evaluation configuration; it is therefore a controlled readout variant rather than a reproduction of the complete GemFilter procedure. Together, these controls vary the amount of reliable reduction exposed within the first few layers while preserving as much of the surrounding evaluation procedure as possible.

Table B1 reports the reliability-matched depth optima for both models and all three evidence-retention thresholds under the Full configuration and the two controls.

Table B1: **Capacity- and speedup-optimal operating points under the Full evaluation configuration and two contrasting controls.** For each model and evidence-retention threshold τ, the table reports the speedup-optimal depth ℓ_{spd}, the capacity-optimal depth ℓ_{cap}, and the corresponding minimum reliable token-retention ratios $\rho^*(\ell;\tau)$. The final column indicates whether the speedup-optimal depth precedes ($<$) or coincides with ($=$) the capacity-optimal depth. Complete latency and speedup measurements underlying these optima are reported in Appendix C.

Model	τ	Configuration	ℓ_{spd}	Retention at spd.	ℓ_{cap}	Retention at cap.	Relation
Llama-3.1-8B	.90	Full	L2	.37	L10	.15	$<$
	.90	without PDSC	L2	.51	L10	.27	$<$
	.90	Post-RoPE last-query	L3	.77	L10	.36	$<$
	.95	Full	L3	.31	L10	.21	$<$
	.95	without PDSC	L2	.57	L10	.30	$<$
	.95	Post-RoPE last-query	L10	.43	L10	.43	$=$
	.98	Full	L3	.39	L10	.28	$<$
	.98	without PDSC	L4	.58	L10	.37	$<$
	.98	Post-RoPE last-query	L10	.51	L10	.51	$=$
Ministral-8B	.90	Full	L2	.48	L12	.28	$<$
	.90	without PDSC	L2	.50	L12	.31	$<$
	.90	Post-RoPE last-query	L12	.23	L12	.23	$=$
	.95	Full	L2	.55	L12	.36	$<$
	.95	without PDSC	L2	.58	L12	.43	$<$
	.95	Post-RoPE last-query	L12	.28	L12	.28	$=$
	.98	Full	L2	.65	L12	.50	$<$
	.98	without PDSC	L2	.70	L12	.55	$<$
	.98	Post-RoPE last-query	L12	.35	L12	.35	$=$

B.3.1 WITHOUT PDSC

Removing PDSC reduces the amount of reliable reduction available at the reported operating points and changes the resulting reliability-matched operating points. At the capacity-optimal depth, the minimum reliable retention ratio increases by 9–12 percentage points for Llama-3.1-8B and 3–7 percentage points for Ministral-8B relative to the Full configuration. For Llama-3.1-8B, the speedup-optimal depths become L2, L2, and L4 for $\tau = .90, .95, .98$, respectively, compared with L2, L3, and L3 under the Full configuration. For Ministral-8B, the speedup-optimal depth remains L2 across all three thresholds.

Despite these changes, the capacity-optimal depths remain L10 for Llama-3.1-8B and L12 for Ministral-8B, and the speedup-optimal depth remains earlier than the capacity-optimal depth in all six model–threshold conditions. Thus, capacity–value separation persists without PDSC, although the exact reliability-matched retention ratios and speedup-optimal depths can change. This result should not be interpreted as establishing a PDSC-independent optimum.

B.3.2 POST-ROPE LAST-QUERY

The Post-RoPE last-query configuration provides a contrasting boundary case in which reliable reduction within the first few Transformer layers is substantially weaker. At $\tau = .95$ and $.98$, the speedup-optimal and capacity-optimal depths coincide for both models: L10 for Llama-3.1-8B and L12 for Ministral-8B. At $\tau = .90$, the two optima also coincide at L12 for Ministral-8B, while

Llama-3.1-8B retains a smaller separation, with L3 speedup-optimal and L10 capacity-optimal. Overall, the two optima coincide in five of the six evaluated model–threshold conditions.

This contrasting case shows that capacity–value separation is not guaranteed merely by allowing reduction to occur early. Across the evaluated controls, separation persists when sufficient reliable reduction is available within the first few Transformer layers and diminishes when reliable early reduction is limited. The observed boundary is consistent with whether sufficient reliable reduction becomes available early enough for the computational benefit of acting sooner to outweigh the benefit of waiting for greater later-stage reduction capacity. This is an empirical pattern over the evaluated configurations rather than a universal necessary or sufficient condition.

C Full Capacity–Value Measurements

This appendix reports the complete numerical measurements underlying the capacity–value analysis in Section 3. Section C.1 gives the reliability-matched capacity profiles across all candidate first context-reduction depths and evidence-retention thresholds. Section C.2 reports the corresponding 128K prefill latencies and computational-value profiles under the FastKV and GemFilter downstream execution settings. Section C.3 provides repeated timing measurements to assess whether the observed latency differences between the speedup-optimal and capacity-optimal depths are larger than run-to-run timing variation.

Together, these results separate three aspects of the principal analysis: how much context can be removed reliably at each depth, how much computational value is obtained at the corresponding reliability-matched operating point, and whether the resulting depth relationship remains stable under repeated timing. The operating points reported here are those used in the principal analysis; this appendix provides their complete depth-wise numerical profiles rather than defining a separate optimization procedure.

C.1 Full Reliability-Matched Capacity Profiles

Table C1 reports the complete depth-wise profiles of the minimum reliable token-retention ratio under the Full evaluation configuration. For each candidate first context-reduction depth ℓ and evidence-retention threshold $\tau \in \{.90, .95, .98\}$, we report $\rho^*(\ell; \tau)$, the interpolated minimum reliable token-retention ratio defined in Eq. A.1. These operating points are those used in the principal capacity–value analysis and are derived from the same underlying controlled retention-ratio sweep reported in Appendix B, using the Full evaluation configuration.

The two retention quantities in Table C1 have different meanings. The threshold τ specifies the required fraction of designated evidence that must be preserved, whereas $\rho^*(\ell; \tau)$ specifies the minimum fraction of context tokens that must be retained to satisfy that requirement. Lower values of $\rho^*(\ell; \tau)$ therefore indicate greater reliable reduction capacity. For example, $\tau = .98$ with $\rho^*(\text{L10}; .98) = .28$ for Llama-3.1-8B means that approximately 28% of the context must be retained at L10 to satisfy the 98% evidence-retention requirement; .28 is not itself an evidence-retention score.

The capacity profiles are strongly depth dependent and non-monotonic. Deeper processing can support substantially stronger reliable reduction than some first-few-layer depths, but increasing depth does not uniformly reduce the required context retention. The two models also exhibit different depth-wise profiles, indicating that reliable reduction capacity cannot be represented by a single common depth schedule.

Importantly, the capacity-optimal depths are not specific to the representative $\tau = .98$ setting shown in the main figure. For Llama-3.1-8B, the minimum $\rho^*(\ell; \tau)$ occurs at L10 for all three evaluated thresholds, with retention ratios .15, .21, and .28 for $\tau = .90, .95, .98$, respectively. For Ministral-8B, the corresponding minimum occurs at L12 for all three thresholds, with retention ratios .28, .36, and .50. Thus, across the evaluated reliability requirements, the greatest reliable reduction capacity consistently occurs beyond the first few Transformer layers. The corresponding latency measurements at these same reliability-matched operating points are reported in Section C.2.

Table C1: Full reliability-matched capacity profiles under the Full evaluation configuration. For each candidate first context-reduction depth ℓ, entries report the minimum token-retention ratio $\rho^*(\ell;\tau)$ satisfying the indicated evidence-retention threshold τ. Column headers denote the required evidence-retention level, whereas table entries denote the fraction of context tokens retained. Lower values indicate greater reliable reduction capacity. Bold entries identify the capacity-optimal depth for each model and threshold; dashes denote depths outside the evaluated candidate-depth range.

	Llama-3.1-8B			Ministral-8B		
Reduction depth ℓ	$\tau = .90$	$\tau = .95$	$\tau = .98$	$\tau = .90$	$\tau = .95$	$\tau = .98$
L2	.37	.45	.53	.48	.55	.65
L3	.28	.31	.39	.60	.69	.81
L4	.23	.29	.35	.42	.55	.67
L5	.36	.44	.51	.46	.54	.65
L6	.42	.49	.57	.71	.78	.87
L7	.55	.60	.68	.82	.89	.96
L8	.33	.39	.52	.61	.74	.79
L9	.28	.34	.40	.71	.79	.89
L10	**.15**	**.21**	**.28**	.40	.49	.70
L11	.35	.40	.53	.43	.51	.67
L12	.53	.59	.67	**.28**	**.36**	**.50**
L13	—	—	—	.50	.60	.68
L14	—	—	—	.55	.66	.75
L15	—	—	—	.52	.68	.76
L16	—	—	—	.38	.50	.69

C.2 Full Prefill Latency and Computational-Value Profiles

Tables C2a–C2f report the complete latency profiles corresponding to the reliability-matched operating points in Table C1. For each candidate first context-reduction depth ℓ and evidence-retention threshold τ, context reduction is applied using the minimum reliable token-retention ratio $\rho^*(\ell;\tau)$ identified in the principal capacity analysis. We then measure the resulting 128K prefill latency with FastKV and GemFilter as two representative downstream execution settings.

The primary computational comparison is the incremental prefill speedup relative to the corresponding downstream execution setting alone. A value greater than $1\times$ indicates that adding the earlier first context-reduction point lowers measured latency relative to that downstream setting without the added reduction. FastKV and GemFilter are evaluated separately; their absolute latency or speedup values are not used to rank the two downstream methods. For completeness, the tables also report speedup versus full attention, which is a second normalization of the same latency measurements rather than a separate timing experiment. The latency protocol and implementation-specific timing boundaries are described in Appendix A.4.

The latency profiles differ substantially from the reliable-reduction-capacity profiles in Table C1. For Llama-3.1-8B, the speedup-optimal depth occurs at L2 or L3 depending on τ, whereas the capacity-optimal depth remains L10. For Ministral-8B, L2 is speedup-optimal across all three thresholds, whereas L12 remains capacity-optimal. FastKV and GemFilter yield different absolute speedups but identify the same speedup-optimal depth for every evaluated model–threshold pair. This agreement is used as a check on the depth relationship rather than as a comparison between the two downstream execution settings.

Crucially, the lower latency at the speedup-optimal depth is not explained by stronger context reduction. Across the six model–threshold conditions, the speedup-optimal operating points retain approximately 10–22 percentage points more context than the corresponding capacity-optimal points. They therefore perform a smaller reliable reduction, yet achieve lower measured prefill latency. The computational advantage arises because this smaller reduction becomes effective earlier, allowing a larger fraction of the remaining Transformer computation to operate on the shortened context. By contrast, waiting for a deeper representation may permit stronger reliable reduction, but more full-context prefill computation has already been performed before that reduction takes effect.

This distinction is also visible in the speedup-versus-full-attention measurements. Across the evaluated conditions, the ratio between the speedups at the speedup-optimal and capacity-optimal operating points ranges from $1.20\times$ to $1.78\times$ for FastKV and from $1.49\times$ to $2.04\times$ for GemFilter. Because the two values within each comparison share the same full-attention reference, this ratio is

equivalently the latency ratio between the capacity-optimal and speedup-optimal operating points. The ratios are computed from the underlying unrounded latency measurements and may therefore differ in the last digit from the quotient of the displayed rounded speedups.

Taken together, the complete profiles show that computational value is determined jointly by how much context can be removed reliably and how early that reduction becomes effective. The depth with the greatest reliable reduction capacity therefore need not minimize measured prefill latency. Tables C2a–C2f provide the complete numerical measurements underlying this capacity–value separation; Section C.3 next examines whether the observed latency ordering is stable under repeated timing.

Tables C2a–C2f: **Full reliability-matched latency and computational-value profiles.** For each candidate first context-reduction depth ℓ, ρ^* denotes the minimum reliable token-retention ratio $\rho^*(\ell;\tau)$ satisfying the indicated evidence-retention threshold. FastKV and GemFilter columns report measured 128K prefill latency, incremental prefill speedup relative to the corresponding downstream execution setting alone, and speedup versus full attention. The Full attention rows report the common per-model reference latency T_{full} defined in Appendix A.4. Bold marks the speedup-optimal depth ℓ_{spd}, and † marks the capacity-optimal depth ℓ_{cap}. Displayed speedups are rounded; ℓ_{spd} is determined from the underlying unrounded measured latency.

Table C2a: Llama-3.1-8B, $\tau = .90$.

		FastKV			GemFilter		
ℓ	ρ^*	Latency (s)	Incremental	vs. full	Latency (s)	Incremental	vs. full
Full attention		68.08			68.08		
Downstream alone		37.20	1.00	1.83	30.26	1.00	2.25
2	0.37	**14.76**	**2.52**	**4.61**	**12.46**	**2.43**	**5.47**
3	0.28	14.79	2.52	4.60	12.61	2.40	5.40
4	0.23	15.87	2.34	4.29	13.79	2.19	4.94
5	0.36	19.78	1.88	3.44	18.64	1.62	3.65
6	0.42	22.56	1.65	3.02	22.44	1.35	3.03
7	0.55	26.27	1.42	2.59	28.21	1.07	2.41
8	0.33	24.55	1.51	2.77	24.37	1.24	2.79
9	0.28	25.98	1.43	2.62	25.42	1.19	2.68
10†	0.15	26.04	1.43	2.61	25.45	1.19	2.67
11	0.35	30.15	1.23	2.26	31.35	0.97	2.17
12	0.53	32.99	1.13	2.06	38.70	0.78	1.76

Table C2b: Llama-3.1-8B, $\tau = .95$.

		FastKV			GemFilter		
ℓ	ρ^*	Latency (s)	Incremental	vs. full	Latency (s)	Incremental	vs. full
Full attention		68.08			68.08		
Downstream alone		37.21	1.00	1.83	30.30	1.00	2.25
2	0.45	16.99	2.19	4.01	14.91	2.03	4.57
3	0.31	**15.41**	**2.41**	**4.42**	**13.30**	**2.28**	**5.12**
4	0.29	16.80	2.22	4.05	14.87	2.04	4.58
5	0.44	21.48	1.73	3.17	20.97	1.44	3.25
6	0.49	23.90	1.56	2.85	24.50	1.24	2.78
7	0.60	27.34	1.36	2.49	30.03	1.01	2.27
8	0.39	25.47	1.46	2.67	26.02	1.16	2.62
9	0.34	26.59	1.40	2.56	26.85	1.13	2.54
10†	0.21	27.40	1.36	2.48	26.29	1.15	2.59
11	0.40	30.59	1.22	2.23	32.99	0.92	2.06
12	0.59	33.44	1.11	2.04	40.46	0.75	1.68

Table C2c: Llama-3.1-8B, $\tau = .98$.

		FastKV			GemFilter		
ℓ	ρ^*	Latency (s)	Incremental	vs. full	Latency (s)	Incremental	vs. full
Full attention		68.08			68.08		
Downstream alone		38.22	1.00	1.78	31.13	1.00	2.19
2	0.53	19.37	1.97	3.51	17.35	1.79	3.92
3	0.39	**17.52**	**2.18**	**3.89**	**15.63**	**1.99**	**4.36**
4	0.35	18.37	2.08	3.71	16.77	1.86	4.06
5	0.51	23.29	1.64	2.92	23.32	1.33	2.92
6	0.57	26.20	1.46	2.60	27.63	1.13	2.46
7	0.68	29.68	1.29	2.29	31.29	0.99	2.18
8	0.52	28.01	1.36	2.43	30.33	1.03	2.24
9	0.40	27.90	1.37	2.44	29.15	1.07	2.34
10†	0.28	28.50	1.34	2.39	28.29	1.10	2.41
11	0.53	32.45	1.18	2.10	37.30	0.83	1.83
12	0.67	34.91	1.09	1.95	44.53	0.70	1.53

Table C2d: Ministral-8B, $\tau = .90$.

		FastKV			GemFilter		
ℓ	ρ^*	Latency (s)	Incremental	vs. full	Latency (s)	Incremental	vs. full
Full attention		42.64			42.64		
Downstream alone		24.45	1.00	1.74	21.45	1.00	1.99
2	0.48	**14.49**	**1.69**	**2.94**	**13.11**	**1.64**	**3.25**
3	0.60	17.26	1.42	2.47	16.98	1.26	2.51
4	0.42	14.83	1.65	2.87	14.11	1.52	3.02
5	0.46	16.21	1.51	2.63	16.25	1.32	2.62
6	0.71	20.36	1.20	2.09	23.06	0.93	1.85
7	0.82	22.17	1.10	1.92	26.89	0.80	1.59
8	0.61	19.95	1.23	2.14	23.06	0.93	1.85
9	0.71	21.56	1.13	1.98	26.80	0.80	1.59
10	0.40	19.12	1.28	2.23	20.73	1.03	2.06
11	0.43	20.16	1.21	2.11	22.68	0.95	1.88
12†	0.28	19.88	1.23	2.14	20.31	1.06	2.10
13	0.50	21.92	1.12	1.95	26.58	0.81	1.60
14	0.55	22.80	1.07	1.87	28.92	0.74	1.47
15	0.52	23.29	1.05	1.83	29.41	0.73	1.45
16	0.38	23.72	1.03	1.80	27.37	0.78	1.56

Table C2e: Ministral-8B, $\tau = .95$.

		FastKV			GemFilter		
ℓ	ρ^*	Latency (s)	Incremental	vs. full	Latency (s)	Incremental	vs. full
Full attention		42.64			42.64		
Downstream alone		24.49	1.00	1.74	21.46	1.00	1.99
2	0.55	**15.92**	**1.54**	**2.68**	**14.74**	**1.46**	**2.89**
3	0.69	18.93	1.29	2.25	19.09	1.12	2.23
4	0.55	16.95	1.44	2.52	16.94	1.27	2.52
5	0.54	17.39	1.41	2.45	17.96	1.20	2.37
6	0.78	21.38	1.15	1.99	24.63	0.87	1.73
7	0.89	23.23	1.05	1.84	28.61	0.75	1.49
8	0.74	21.46	1.14	1.99	26.07	0.82	1.64
9	0.79	22.31	1.10	1.91	28.39	0.76	1.50
10	0.49	19.95	1.23	2.14	22.82	0.94	1.87
11	0.51	20.81	1.18	2.05	24.49	0.88	1.74
12†	0.36	20.45	1.20	2.08	22.21	0.97	1.92
13	0.60	22.55	1.09	1.89	29.04	0.74	1.47
14	0.66	23.25	1.05	1.83	31.46	0.68	1.36
15	0.68	23.75	1.03	1.80	33.20	0.65	1.28
16	0.50	23.91	1.02	1.78	30.18	0.71	1.41

Table C2f: Ministral-8B, $\tau = .98$.

		FastKV			GemFilter		
ℓ	ρ^*	Latency (s)	Incremental	vs. full	Latency (s)	Incremental	vs. full
Full attention		42.64			42.64		
Downstream alone		24.75	1.00	1.72	21.67	1.00	1.97
2	0.65	**18.09**	**1.37**	**2.36**	**17.32**	**1.25**	**2.46**
3	0.81	21.46	1.15	1.99	22.25	0.97	1.92
4	0.67	19.24	1.29	2.22	20.04	1.08	2.13
5	0.65	19.35	1.28	2.20	20.80	1.04	2.05
6	0.87	23.17	1.07	1.84	27.27	0.79	1.56
7	0.96	24.34	1.02	1.75	30.48	0.71	1.40
8	0.79	22.43	1.10	1.90	27.66	0.78	1.54
9	0.89	23.73	1.04	1.80	31.20	0.69	1.37
10	0.70	22.25	1.11	1.92	27.94	0.78	1.53
11	0.67	22.39	1.11	1.90	28.41	0.76	1.50
12†	0.50	21.67	1.14	1.97	25.73	0.84	1.66
13	0.68	23.24	1.07	1.83	31.07	0.70	1.37
14	0.75	23.92	1.03	1.78	33.91	0.64	1.26
15	0.76	24.40	1.01	1.75	35.26	0.61	1.21
16	0.69	24.55	1.01	1.74	34.86	0.62	1.22

C.3 Timing Repeatability and Uncertainty

The principal latency profiles in Tables C2a–C2f follow the depth-sweep protocol in Appendix A.4, in which each operating point is timed twice and the lower observed latency is used for the sweep. We therefore perform additional repeated measurements to assess whether the observed separation between the speedup-optimal and capacity-optimal depths could be explained by run-to-run timing variation.

For the repeated measurements, each evaluated configuration at $\tau = .98$ is timed 25 times. Additional measurements at $\tau = .90$ and .95 use 10 repetitions per evaluated configuration. The corresponding downstream-only reference is repeated the same number of times as its augmented counterparts, and the full-attention reference is measured 10 times. For each latency measurement, Tables C3a–C3f report the median, mean, and 95% t-interval of the mean. Speedup intervals are obtained from 10,000 bootstrap resamples using the corresponding repeated reference measurements.

These repeated measurements serve only to assess the stability of the depth relationship. Because they summarize repeated runs rather than the minimum of two measurements used in the principal depth sweep, small numerical differences between Tables C2a–C2f and C3a–C3f are expected. The capacity- and speedup-optimal depths in the principal analysis therefore remain defined by the common depth-sweep protocol rather than being re-estimated from the repeated measurements.

The repeated measurements show that the latency differences underlying capacity–value separation are substantially larger than the observed timing uncertainty. At $\tau = .98$, for example, Llama-3.1-8B with FastKV achieves an incremental prefill speedup of $2.18\times$ $[2.17, 2.19]$ at the speedup-optimal depth L3, compared with $1.33\times$ $[1.33, 1.34]$ at the capacity-optimal depth L10. For Ministral-8B, the corresponding FastKV values are $1.37\times$ $[1.37, 1.38]$ at the speedup-optimal depth L2 and $1.15\times$ $[1.14, 1.15]$ at the capacity-optimal depth L12. Thus, in both representative comparisons, the confidence intervals around the speedup-optimal and capacity-optimal operating points are well separated.

The repeated measurements also resolve differences among nearby first-few-layer candidates. For Llama-3.1-8B at $\tau = .98$, FastKV gives $1.96\times$ incremental speedup at L2 and $2.18\times$ at L3, whereas Ministral-8B gives $1.37\times$ at L2 and $1.16\times$ at L3. These measurements support the depth ordering identified by the principal sweep rather than indicating that the selected first-few-layer optimum is caused by small timing fluctuations.

Taken together, the repeated measurements show that the lower latency observed at the speedup-optimal depth is stable under repeated timing and is not explained by measurement noise around the capacity-optimal depth. Exact latency values differ slightly between the depth sweep and repeated measurements because the two protocols summarize repeated runs differently, but the qualitative ordering relevant to capacity–value separation remains unchanged.

Tables C3a–C3f: **Repeated timing measurements for the reliability-matched operating points.** Each table reports median and mean prefill latency, the 95% t-interval of the mean, incremental prefill speedup relative to the corresponding downstream-only setting, and speedup versus full attention. Speedup intervals are 95% bootstrap intervals from 10,000 resamples. Measurements use $n = 25$ repetitions for $\tau = .98$ and $n = 10$ for $\tau = .90$ and .95. These repeated measurements assess timing stability and do not redefine the speedup-optimal depths identified by the principal depth sweep.

Table C3a: Llama-3.1-8B, $\tau = .98$, $n = 25$.

Downstream	ℓ	Median (s)	Mean (s)	95% CI of mean	Incremental [95% CI]	vs. full [95% CI]
FastKV	alone	39.16	39.09	[38.94, 39.23]		1.82 [1.82, 1.83]
	2	19.96	19.92	[19.84, 20.00]	1.96 [1.95, 1.97]	3.58 [3.56, 3.60]
	3	17.96	17.92	[17.83, 18.00]	2.18 [2.17, 2.19]	3.98 [3.96, 4.00]
	4	18.85	18.82	[18.73, 18.91]	2.08 [2.07, 2.09]	3.79 [3.77, 3.81]
	5	23.86	23.81	[23.72, 23.90]	1.64 [1.63, 1.65]	2.99 [2.98, 3.01]
	10	29.32	29.28	[29.16, 29.40]	1.33 [1.33, 1.34]	2.43 [2.42, 2.45]
	12	35.98	35.94	[35.79, 36.09]	1.09 [1.08, 1.09]	1.98 [1.97, 2.00]
GemFilter	alone	31.29	31.29	[31.19, 31.39]		2.28 [2.27, 2.29]
	2	17.82	17.80	[17.72, 17.87]	1.76 [1.75, 1.77]	4.01 [3.99, 4.03]
	3	16.01	15.99	[15.91, 16.07]	1.96 [1.95, 1.97]	4.46 [4.43, 4.49]
	4	17.15	17.13	[17.05, 17.21]	1.83 [1.82, 1.84]	4.16 [4.14, 4.19]
	5	23.88	23.86	[23.76, 23.95]	1.31 [1.31, 1.32]	2.99 [2.97, 3.00]
	10	28.97	28.96	[28.83, 29.08]	1.08 [1.08, 1.09]	2.46 [2.45, 2.48]
	12	45.48	45.41	[45.25, 45.56]	0.69 [0.69, 0.69]	1.57 [1.56, 1.58]

Table C3b: Ministral-8B, $\tau = .98$, $n = 25$.

Downstream	ℓ	Median (s)	Mean (s)	95% CI of mean	Incremental [95% CI]	vs. full [95% CI]
FastKV	alone	25.14	25.13	[25.09, 25.17]		1.73 [1.72, 1.73]
	2	18.32	18.30	[18.25, 18.34]	1.37 [1.37, 1.38]	2.37 [2.36, 2.38]
	3	21.69	21.68	[21.64, 21.72]	1.16 [1.16, 1.16]	2.00 [2.00, 2.01]
	4	19.46	19.44	[19.40, 19.48]	1.29 [1.29, 1.30]	2.23 [2.23, 2.24]
	5	19.57	19.55	[19.51, 19.59]	1.29 [1.28, 1.29]	2.22 [2.21, 2.23]
	10	22.42	22.40	[22.35, 22.44]	1.12 [1.12, 1.12]	1.94 [1.93, 1.94]
	12	21.91	21.90	[21.85, 21.94]	1.15 [1.14, 1.15]	1.98 [1.98, 1.99]
GemFilter	alone	22.04	22.02	[21.98, 22.06]		1.97 [1.96, 1.98]
	2	17.53	17.51	[17.48, 17.55]	1.26 [1.25, 1.26]	2.48 [2.47, 2.49]
	3	22.57	22.55	[22.50, 22.59]	0.98 [0.97, 0.98]	1.92 [1.92, 1.93]
	4	20.21	20.22	[20.18, 20.25]	1.09 [1.09, 1.09]	2.15 [2.14, 2.15]
	5	20.95	20.93	[20.89, 20.97]	1.05 [1.05, 1.05]	2.07 [2.07, 2.08]
	10	28.28	28.26	[28.21, 28.31]	0.78 [0.78, 0.78]	1.54 [1.53, 1.54]
	12	26.01	26.01	[25.96, 26.05]	0.85 [0.84, 0.85]	1.67 [1.66, 1.67]

Table C3c: Llama-3.1-8B, $\tau = .90$, $n = 10$.

Downstream	ℓ	Median (s)	Mean (s)	95% CI of mean	Incremental [95% CI]	vs. full [95% CI]
FastKV	alone	38.98	38.79	[38.43, 39.14]		1.84 [1.83, 1.86]
	2	15.58	15.49	[15.31, 15.66]	2.50 [2.48, 2.54]	4.60 [4.56, 4.65]
	3	15.67	15.56	[15.37, 15.75]	2.49 [2.46, 2.53]	4.58 [4.54, 4.64]
	4	16.89	16.76	[16.54, 16.99]	2.31 [2.29, 2.35]	4.25 [4.21, 4.31]
	5	20.89	20.79	[20.56, 21.01]	1.87 [1.85, 1.89]	3.43 [3.40, 3.47]
	10	27.24	27.15	[26.90, 27.40]	1.43 [1.41, 1.45]	2.63 [2.61, 2.65]
	12	34.66	34.53	[34.14, 34.92]	1.12 [1.11, 1.14]	2.06 [2.05, 2.09]
GemFilter	alone	31.27	31.19	[30.90, 31.48]		2.29 [2.27, 2.31]
	2	13.15	13.05	[12.90, 13.19]	2.39 [2.36, 2.42]	5.46 [5.42, 5.52]
	3	13.45	13.35	[13.17, 13.52]	2.34 [2.31, 2.37]	5.34 [5.29, 5.41]
	4	14.76	14.65	[14.46, 14.83]	2.13 [2.10, 2.16]	4.87 [4.82, 4.93]
	5	19.81	19.70	[19.48, 19.93]	1.58 [1.56, 1.60]	3.62 [3.59, 3.66]
	10	26.84	26.67	[26.35, 26.99]	1.17 [1.16, 1.19]	2.67 [2.65, 2.71]
	12	40.15	40.04	[39.64, 40.43]	0.78 [0.77, 0.79]	1.78 [1.77, 1.80]

Table C3d: Llama-3.1-8B, $\tau = .95$, $n = 10$.

Downstream	ℓ	Median (s)	Mean (s)	95% CI of mean	Incremental [95% CI]	vs. full [95% CI]
FastKV	alone	39.11	38.98	[38.59, 39.36]		1.83 [1.82, 1.85]
	2	17.87	17.77	[17.59, 17.96]	2.19 [2.17, 2.22]	4.01 [3.98, 4.05]
	3	16.28	16.17	[15.99, 16.36]	2.41 [2.38, 2.44]	4.41 [4.37, 4.46]
	4	17.73	17.62	[17.41, 17.82]	2.21 [2.19, 2.24]	4.05 [4.01, 4.09]
	5	22.52	22.44	[22.22, 22.67]	1.74 [1.72, 1.76]	3.18 [3.15, 3.21]
	10	28.84	28.67	[28.36, 28.98]	1.36 [1.34, 1.38]	2.49 [2.47, 2.51]
	12	35.25	35.12	[34.76, 35.48]	1.11 [1.10, 1.12]	2.03 [2.02, 2.05]
GemFilter	alone	31.31	31.17	[30.89, 31.46]		2.29 [2.27, 2.31]
	2	15.70	15.59	[15.41, 15.76]	2.00 [1.98, 2.03]	4.57 [4.54, 4.63]
	3	14.12	14.01	[13.83, 14.20]	2.22 [2.20, 2.26]	5.09 [5.04, 5.15]
	4	15.77	15.68	[15.48, 15.88]	1.99 [1.96, 2.02]	4.55 [4.50, 4.61]
	5	22.12	21.98	[21.73, 22.23]	1.42 [1.40, 1.44]	3.24 [3.22, 3.28]
	10	27.76	27.60	[27.29, 27.92]	1.13 [1.12, 1.14]	2.58 [2.56, 2.61]
	12	42.26	42.12	[41.72, 42.51]	0.74 [0.73, 0.75]	1.69 [1.68, 1.71]

Table C3e: Ministral-8B, $\tau = .90$, $n = 10$.

Downstream	ℓ	Median (s)	Mean (s)	95% CI of mean	Incremental [95% CI]	vs. full [95% CI]
FastKV	alone	25.17	25.13	[25.05, 25.21]		1.73 [1.72, 1.73]
	2	14.90	14.86	[14.77, 14.96]	1.69 [1.68, 1.70]	2.92 [2.90, 2.94]
	3	17.77	17.74	[17.66, 17.83]	1.42 [1.41, 1.42]	2.45 [2.43, 2.46]
	4	15.28	15.22	[15.13, 15.31]	1.65 [1.64, 1.66]	2.85 [2.84, 2.87]
	5	16.64	16.60	[16.52, 16.68]	1.51 [1.51, 1.52]	2.61 [2.60, 2.63]
	10	19.66	19.61	[19.50, 19.72]	1.28 [1.28, 1.29]	2.21 [2.20, 2.23]
	12	20.45	20.42	[20.30, 20.53]	1.23 [1.22, 1.24]	2.13 [2.12, 2.14]
GemFilter	alone	22.06	22.02	[21.93, 22.12]		1.97 [1.96, 1.98]
	2	13.50	13.48	[13.39, 13.57]	1.63 [1.62, 1.64]	3.22 [3.20, 3.24]
	3	17.49	17.45	[17.36, 17.54]	1.26 [1.25, 1.27]	2.49 [2.47, 2.50]
	4	14.52	14.49	[14.41, 14.56]	1.52 [1.51, 1.53]	2.99 [2.98, 3.01]
	5	16.71	16.67	[16.58, 16.76]	1.32 [1.31, 1.33]	2.60 [2.59, 2.62]
	10	21.31	21.29	[21.18, 21.40]	1.03 [1.03, 1.04]	2.04 [2.03, 2.05]
	12	20.90	20.86	[20.77, 20.96]	1.06 [1.05, 1.06]	2.08 [2.07, 2.09]

Table C3f: Ministral-8B, $\tau = .95$, $n = 10$.

Downstream	ℓ	Median (s)	Mean (s)	95% CI of mean	Incremental [95% CI]	vs. full [95% CI]
FastKV	alone	25.15	25.11	[24.98, 25.23]		1.73 [1.72, 1.74]
	2	16.33	16.30	[16.21, 16.38]	1.54 [1.53, 1.55]	2.66 [2.65, 2.68]
	3	19.47	19.40	[19.30, 19.51]	1.29 [1.29, 1.30]	2.24 [2.23, 2.25]
	4	17.39	17.36	[17.27, 17.45]	1.45 [1.44, 1.46]	2.50 [2.49, 2.51]
	5	17.85	17.81	[17.74, 17.88]	1.41 [1.40, 1.42]	2.44 [2.43, 2.45]
	10	20.48	20.45	[20.36, 20.54]	1.23 [1.22, 1.23]	2.12 [2.11, 2.13]
	12	20.97	20.94	[20.86, 21.03]	1.20 [1.19, 1.20]	2.07 [2.06, 2.08]
GemFilter	alone	22.05	22.04	[21.93, 22.14]		1.97 [1.96, 1.98]
	2	15.14	15.10	[15.02, 15.18]	1.46 [1.45, 1.47]	2.87 [2.86, 2.89]
	3	19.60	19.57	[19.47, 19.68]	1.13 [1.12, 1.13]	2.22 [2.21, 2.23]
	4	17.46	17.41	[17.31, 17.51]	1.27 [1.26, 1.27]	2.49 [2.48, 2.51]
	5	18.46	18.43	[18.35, 18.51]	1.20 [1.19, 1.20]	2.35 [2.35, 2.37]
	10	23.41	23.38	[23.28, 23.49]	0.94 [0.94, 0.95]	1.86 [1.85, 1.87]
	12	22.79	22.75	[22.62, 22.87]	0.97 [0.96, 0.97]	1.91 [1.90, 1.92]

D LongBench Downstream Performance Sensitivity

The LongBench analysis in Section 4 examines downstream sensitivity to local changes in the depth and strength of an added early context reduction. For both Llama-3.1-8B and Ministral-8B, we evaluate the same 3×3 grid,

$$\ell \in \{\mathrm{L2}, \mathrm{L3}, \mathrm{L4}\}, \qquad \rho \in \{.60, .70, .80\},$$

with FastKV and GemFilter as representative downstream execution settings. At each operating point, the added first context reduction precedes an otherwise unchanged downstream selection procedure. The reduction is activated only for prompts longer than 8192 tokens; shorter prompts follow the corresponding downstream-only execution path. The reported LongBench scores therefore characterize benchmark-level performance under this deployment policy.

These operating points are not matched under a common NIAH evidence-retention threshold and therefore do not redefine $\rho^*(\ell;\tau)$, ℓ_{cap}, or ℓ_{spd}. FastKV and GemFilter are each compared only with their own downstream-only baseline. Section D.1 reports the complete Overall-score grids for both models, while Section D.2 provides the corresponding category-level results across single-document QA, multi-document QA, summarization, few-shot learning, synthetic tasks, and code.

D.1 Overall Sensitivity Across Early-Layer Operating Points

Tables D1a and D1b report the complete 3×3 Overall LongBench sensitivity grids for Llama-3.1-8B and Ministral-8B, respectively. For Llama-3.1-8B, FastKV score changes range from -1.37 to -0.17, while GemFilter changes range from -0.69 to $+0.01$. For Ministral-8B, the corresponding ranges are -1.46 to -0.35 and -1.01 to -0.08. Higher retention generally yields smaller degradation, although the pattern is not uniformly monotonic across depths and downstream settings. Section D.2 further examines the category-level behavior underlying these aggregate results.

Table D1a: **Overall LongBench sensitivity to an added early context-reduction stage for Llama-3.1-8B.** Rows vary the first context-reduction depth ℓ, columns vary the token-retention ratio ρ, and the downstream FastKV or GemFilter procedure remains unchanged. Entries report the Overall score, with the change from the corresponding downstream-only baseline shown in parentheses. Changes are computed from unrounded scores.

	FastKV (47.53)			GemFilter (41.65)		
ℓ	$\rho = .60$	$\rho = .70$	$\rho = .80$	$\rho = .60$	$\rho = .70$	$\rho = .80$
L2	46.33 (−1.20)	46.88 (−0.64)	46.75 (−0.77)	41.42 (−0.23)	41.66 (+0.01)	41.53 (−0.13)
L3	46.43 (−1.10)	46.53 (−1.00)	47.36 (−0.17)	41.45 (−0.20)	40.97 (−0.69)	41.20 (−0.45)
L4	46.16 (−1.37)	46.53 (−0.99)	46.88 (−0.65)	41.10 (−0.56)	41.34 (−0.31)	41.45 (−0.20)

Table D1b: **Overall LongBench sensitivity to an added early context-reduction stage for Ministral-8B.** Rows vary the first context-reduction depth ℓ, columns vary the token-retention ratio ρ, and the downstream FastKV or GemFilter procedure remains unchanged. Entries report the Overall score, with the change from the corresponding downstream-only baseline shown in parentheses. Changes are computed from unrounded scores.

	FastKV (49.34)			GemFilter (45.87)		
ℓ	$\rho = .60$	$\rho = .70$	$\rho = .80$	$\rho = .60$	$\rho = .70$	$\rho = .80$
L2	47.88 (−1.46)	48.32 (−1.02)	48.71 (−0.63)	45.06 (−0.80)	45.30 (−0.56)	45.78 (−0.08)
L3	47.96 (−1.38)	48.33 (−1.01)	48.68 (−0.67)	45.00 (−0.87)	44.85 (−1.01)	45.46 (−0.40)
L4	48.22 (−1.12)	48.59 (−0.75)	48.99 (−0.35)	44.97 (−0.89)	44.88 (−0.98)	45.40 (−0.46)

D.2 Category-Level Downstream Sensitivity

Tables D2a–D2d decompose the Overall LongBench results in Section D.1 into six task categories: single-document QA, multi-document QA, summarization, few-shot learning, synthetic tasks, and code. For each model and downstream execution setting, the tables report the downstream-only baseline followed by all nine combinations of first context-reduction depth and token-retention ratio. The reported ΔS is computed from the Overall score relative to the corresponding downstream-only baseline.

The category-level results show greater variation than the aggregate Overall scores. Some categories remain nearly unchanged across the evaluated early-layer depth–retention grid, whereas others are more sensitive to depth or retention, with the pattern depending on both the model and downstream execution setting. In particular, Ministral-8B summarization remains close to its corresponding baseline across both settings, while larger variations appear in several QA, code, and synthetic-task conditions. Thus, the relatively small Overall-score changes in Section D.1 should be interpreted as benchmark-level sensitivity rather than as evidence that all LongBench categories respond uniformly to early context reduction.

Table D2a: **Category-level LongBench sensitivity for Llama-3.1-8B with FastKV.** Rows report the FastKV-only baseline and the nine added first context-reduction configurations over $\ell \in \{\mathrm{L2}, \mathrm{L3}, \mathrm{L4}\}$ and $\rho \in \{.60, .70, .80\}$. Columns report category averages, Overall score, and ΔS relative to the FastKV-only baseline. ΔS is computed from unrounded Overall scores; displayed scores are rounded to two decimal places.

ℓ	ρ	Single-doc QA	Multi-doc QA	Summarization	Few-shot	Synthetic	Code	Overall	ΔS
—	alone	41.90	43.63	24.65	69.33	53.45	57.49	47.53	0.00
L2	.60	38.99	43.67	24.63	68.00	51.43	56.27	46.33	−1.20
L2	.70	40.67	44.06	24.70	68.07	52.02	56.79	46.88	−0.64
L2	.80	40.52	42.94	24.72	68.40	52.07	57.08	46.75	−0.77
L3	.60	39.26	43.33	24.63	68.11	51.95	56.51	46.43	−1.10
L3	.70	39.63	42.99	24.77	68.09	52.01	57.01	46.53	−1.00
L3	.80	41.05	44.57	24.94	68.54	53.18	57.03	47.36	−0.17
L4	.60	38.88	42.83	24.66	67.49	51.96	56.52	46.16	−1.37
L4	.70	39.79	43.22	24.51	68.23	52.10	56.53	46.53	−0.99
L4	.80	40.13	43.72	24.76	68.64	52.26	56.90	46.88	−0.65

Table D2b: **Category-level LongBench sensitivity for Llama-3.1-8B with GemFilter.** Rows report the GemFilter-only baseline and the nine added first context-reduction configurations over $\ell \in \{\mathrm{L2}, \mathrm{L3}, \mathrm{L4}\}$ and $\rho \in \{.60, .70, .80\}$. Columns report category averages, Overall score, and ΔS relative to the GemFilter-only baseline. ΔS is computed from unrounded Overall scores; displayed scores are rounded to two decimal places.

ℓ	ρ	Single-doc QA	Multi-doc QA	Summarization	Few-shot	Synthetic	Code	Overall	ΔS
—	alone	33.80	38.77	24.85	65.41	45.17	43.81	41.65	0.00
L2	.60	31.97	39.79	24.78	65.38	45.55	42.94	41.42	−0.23
L2	.70	32.99	39.86	25.04	66.16	43.85	43.39	41.66	+0.01
L2	.80	33.18	39.37	24.86	64.94	45.68	43.04	41.53	−0.13
L3	.60	32.38	39.01	24.86	65.45	45.72	43.37	41.45	−0.20
L3	.70	31.82	38.72	24.88	64.83	44.25	43.10	40.97	−0.69
L3	.80	31.98	39.46	24.83	64.53	45.21	43.21	41.20	−0.45
L4	.60	32.40	38.30	24.86	65.08	44.99	42.83	41.10	−0.56
L4	.70	32.86	39.47	24.77	65.58	43.65	43.08	41.34	−0.31
L4	.80	32.32	39.58	24.76	65.03	45.99	43.13	41.45	−0.20

Table D2c: **Category-level LongBench sensitivity for Ministral-8B with FastKV.** Rows report the FastKV-only baseline and the nine added first context-reduction configurations over $\ell \in \{\mathrm{L2}, \mathrm{L3}, \mathrm{L4}\}$ and $\rho \in \{.60, .70, .80\}$. Columns report category averages, Overall score, and ΔS relative to the FastKV-only baseline. ΔS is computed from unrounded Overall scores; displayed scores are rounded to two decimal places.

ℓ	ρ	Single-doc QA	Multi-doc QA	Summarization	Few-shot	Synthetic	Code	Overall	ΔS
—	alone	39.70	49.71	23.47	70.70	55.25	64.11	49.34	0.00
L2	.60	38.00	48.02	23.27	69.49	53.25	61.63	47.88	−1.46
L2	.70	39.10	48.67	23.36	69.90	53.00	62.05	48.32	−1.02
L2	.80	39.48	49.60	23.39	69.92	53.75	62.38	48.71	−0.63
L3	.60	38.37	47.35	23.54	69.76	53.00	62.14	47.96	−1.38
L3	.70	39.51	47.44	23.62	69.86	53.25	62.74	48.33	−1.01
L3	.80	39.64	48.64	23.47	69.88	54.00	62.96	48.68	−0.67
L4	.60	39.16	47.54	23.44	69.46	53.25	63.09	48.22	−1.12
L4	.70	40.06	47.94	23.48	70.05	53.25	63.18	48.59	−0.75
L4	.80	40.26	49.37	23.53	70.20	53.50	63.39	48.99	−0.35

Table D2d: **Category-level LongBench sensitivity for Ministral-8B with GemFilter.** Rows report the GemFilter-only baseline and the nine added first context-reduction configurations over $\ell \in \{\mathrm{L2}, \mathrm{L3}, \mathrm{L4}\}$ and $\rho \in \{.60, .70, .80\}$. Columns report category averages, Overall score, and ΔS relative to the GemFilter-only baseline. ΔS is computed from unrounded Overall scores; displayed scores are rounded to two decimal places.

ℓ	ρ	Single-doc QA	Multi-doc QA	Summarization	Few-shot	Synthetic	Code	Overall	ΔS
—	alone	39.72	52.47	25.54	65.35	41.00	51.29	45.87	0.00
L2	.60	38.89	50.94	25.48	64.37	41.25	49.74	45.06	−0.80
L2	.70	38.16	51.39	25.53	64.92	42.50	49.94	45.30	−0.56
L2	.80	39.18	50.94	25.51	65.98	43.25	50.60	45.78	−0.08
L3	.60	38.08	50.38	25.60	64.53	42.00	50.10	45.00	−0.87
L3	.70	39.15	50.82	25.59	65.22	37.75	49.91	44.85	−1.01
L3	.80	40.28	51.13	25.60	65.54	39.25	50.64	45.46	−0.40
L4	.60	38.34	50.89	25.49	64.88	40.50	49.89	44.97	−0.89
L4	.70	39.18	49.90	25.59	65.44	39.00	49.89	44.88	−0.98
L4	.80	39.38	51.05	25.63	65.65	40.25	50.39	45.40	−0.46

E Reduction Depths and Execution Semantics in Prior Work

Transformer depth has different operational meanings across long-context efficiency methods. A reported layer may denote a relevance-scoring point, an in-path context-reduction point, the onset of a progressive pruning schedule, or a token-specific halting decision. Similar layer indices can therefore correspond to different computational events and affect different portions of the remaining Transformer computation.

This distinction matters when relating prior methods to the first context-reduction depth studied in this work. We organize the comparison according to what operation occurs at the reported depth and how execution proceeds afterward. The reported layers are selected under method-specific objectives and should therefore be interpreted according to their execution semantics rather than as directly comparable estimates of a common optimal reduction depth.

E.1 Explicitly Reported Selection and Reduction Depths

Table E1 summarizes representative methods that report concrete selection, pruning, halting, or progressive-reduction depths. The examples illustrate several distinct execution patterns.

GemFilter uses an intermediate representation at approximately L13 for Llama-3.1-8B to select a reduced raw-token subsequence, which is then re-encoded from the beginning of the model (Shi et al., 2026). FastKV instead retains the full active sequence until its Token-Selective Propagation layer (approximately L15 for Llama-3.1-8B and L17 for Ministral-8B), after which selected hidden states continue in-path (Jo et al., 2026). Thus, the reported GemFilter and FastKV depths correspond to different computational events even when both are described as intermediate-layer selection.

Other methods introduce additional depth semantics. PromptDistill performs intermediate hidden-state retention and can use either single- or multi-stage selection (Jin et al., 2026). CLAA aggregates relevance information across multiple layers before pruning at a designated layer (McDanel et al., 2026). DASH begins token-specific selective halting once representation stability permits individual tokens to skip later computation (Chen et al., 2026), whereas SlimInfer progressively reduces active hidden-state blocks across several layers rather than at a single reduction point (Long et al., 2026).

These distinctions are important when interpreting numerical proximity between reported depths. For Llama-3.1-8B, our capacity-optimal depth L10 lies near several previously reported operating depths or schedule onsets, including GemFilter L13, FastKV L15, PromptDistill L13, CLAA L15, and SlimInfer L10. This proximity is descriptive: the reported layers correspond to different operations and are selected under different criteria. In our analysis, ℓ_{cap} instead identifies the candidate first context-reduction depth with the smallest reliability-matched retention ratio $\rho^*(\ell; \tau)$, whereas ℓ_{spd} identifies the depth with the lowest measured prefill latency at those operating points.

Table E1: **Reported selection and reduction depths in representative depth-aware methods.** Reported layers may correspond to raw-token filtering, in-path hidden-state selection, cross-layer scoring followed by pruning, token-specific halting, or progressive reduction. The table therefore emphasizes the operation performed at each depth and the subsequent execution path rather than treating the reported layer indices as directly comparable optima.

Method	Model	First reported selection / reduction depth	Subsequent execution
GemFilter	Llama-3.1-8B, 32L	L13	selected raw tokens are re-encoded from L1
FastKV	Llama-3.1-8B, 32L	L15	selected hidden states continue in-path
	Ministral-8B, 36L	L17	selected hidden states continue in-path
PromptDistill	Llama-3.1-8B, 32L	L13	selected hidden states continue in-path
CLAA	Llama-3.1-8B, 32L	L15	pruned hidden-state sequence continues in-path
DASH	Qwen2.5-7B, 28L	L11	halted tokens skip subsequent attention and FFN
SlimInfer	Llama-3.1-8B, 32L	L10 $\rightarrow$ L20 $\rightarrow$ L30	active blocks are progressively reduced
	Qwen2.5-7B, 28L	L9 $\rightarrow$ L18 $\rightarrow$ L26	active blocks are progressively reduced

E.2 Adaptive and Progressive Reduction Policies

Several related methods do not use a single globally fixed reduction depth. Instead, they either select the reduction depth adaptively for each input or distribute reduction across multiple Transformer layers. These methods provide complementary evidence that useful token selection and retention behavior varies with depth, but their reported depths or schedules represent different decision variables from the first context-reduction depth studied here.

ASL selects the pruning depth adaptively according to input-dependent changes in token-ranking behavior (Taniguchi et al., 2026). Its operating depth is therefore determined per input rather than fixed globally. LazyLLM similarly varies token participation across depth and allows previously skipped tokens to become active again, yielding a progressive and partially recoverable execution policy rather than a permanent context cut at a single layer (Fu et al., 2024).

Other methods explicitly define multi-stage or depth-dependent reduction schedules. SlimInfer progressively reduces active hidden-state blocks across several layers, with a representative Llama-3.1-8B schedule beginning around L10 and applying further reduction near L20 and L30 (Long et al., 2026). PyramidInfer likewise varies the retained context across Transformer depth through a progressive pyramidal schedule (Yang et al., 2024). PyramidKV assigns nonuniform KV-cache budgets across layers, introducing depth-dependent context allocation without necessarily shortening the active prefill hidden-state sequence at the same depth (Cai et al., 2025).

These approaches therefore optimize adaptive depths, progressive schedules, or layer-wise context budgets rather than a single reliability-matched first reduction point. Our analysis isolates a complementary question: when should the active context first become shorter? For each candidate depth ℓ, we estimate the minimum retention ratio $\rho^*(\ell;\tau)$ satisfying the same evidence-retention requirement and measure the computational value obtained when that reduction first takes effect. This controlled comparison makes the tradeoff between reliable reduction capacity and computational value directly observable.

E.3 Different Computational Objects and Complementary Efficiency Mechanisms

Long-context efficiency methods can reduce different computational objects, including attention interactions, cached KV states, textual input length, and the Transformer depth traversed by prompt tokens. These mechanisms can all reduce inference cost, but they are operationally distinct from shortening the active hidden-state sequence during prefill.

Sparse-prefill methods such as MInference, FlexPrefill, and SpargeAttention reduce the number of attention interactions evaluated for long contexts while generally retaining the active token sequence (Jiang et al., 2024a; Lai et al., 2025; Zhang et al., 2025). Their primary optimization target is therefore the attention pattern rather than the number of token representations propagated through subsequent Transformer layers.

KV-oriented methods instead reduce, retrieve, or allocate cached context states. SnapKV performs observation-based KV selection, while HeadKV, Ada-KV, and OmniKV introduce head-, layer-, or depth-dependent KV allocation and selection mechanisms (Li et al., 2024; Fu et al., 2025; Feng et al., 2025; Hao et al., 2025). Such reductions in KV storage or access do not necessarily imply that the active prefill hidden-state sequence becomes shorter at the corresponding layer.

Prompt-compression methods, including LLMLingua, LongLLMLingua, and LLMLingua-2, shorten the textual input before ordinary Transformer execution begins (Jiang et al., 2023; 2024b; Pan et al., 2024). Their compression decision therefore precedes the depth-dependent internal representations considered in our analysis. Conversely, reduced prompt-depth methods such as SwiftKV and related shallow-prefill approaches reduce how much Transformer computation is applied to prompt tokens without necessarily removing those tokens from the active sequence (Qiao et al., 2025; Oh et al., 2026; He et al., 2026).

These approaches are complementary to the first context-reduction decision studied here. Our analysis focuses specifically on when the active context first becomes shorter during Transformer execution. For each candidate depth ℓ, we evaluate how much context can be removed under a common evidence-retention requirement and then measure the computational value obtained when that shortening takes effect. Thus, reductions in attention interactions, KV state, input length, or prompt-computation depth should not be interpreted as direct estimates of ℓ_{cap} or ℓ_{spd}.

E.4 Relation to the Capacity- and Speedup-Optimal Depths

Prior work establishes that token relevance, ranking stability, representation stability, and useful retention budgets can vary substantially across Transformer depth. The contribution of our analysis is therefore not the observation that intermediate or depth-dependent context reduction is possible. Instead, we ask whether the depth supporting the greatest reliable reduction capacity is also the depth that provides the greatest computational value under a common reliability requirement.

This distinction gives a specific interpretation to the capacity-optimal depth. For each candidate first context-reduction depth ℓ, we estimate the minimum token-retention ratio $\rho^*(\ell;\tau)$ required to satisfy the same evidence-retention threshold. The capacity-optimal depth ℓ_{cap} is the depth at which this required retention is smallest. For Llama-3.1-8B, $\ell_{\mathrm{cap}} = \mathrm{L10}$ across the evaluated thresholds, numerically near several previously reported intermediate selection or pruning depths. As discussed in Appendices E.1–E.3, this proximity is descriptive because those prior depths correspond to different operations and optimization criteria.

The key result is that the capacity-optimal depth need not maximize computational value. For Llama-3.1-8B, measured prefill latency is minimized substantially earlier, at L2 or L3 depending on the evidence-retention threshold. Moreover, these speedup-optimal operating points retain more context than the capacity-optimal point. Their advantage therefore does not arise from stronger reduction, but from making a smaller reliable reduction effective earlier, allowing more subsequent Transformer computation to operate on the shortened context.

The capacity–value distinction therefore complements existing depth-aware methods rather than replacing their selection criteria or reduction policies. It separates two design questions that can otherwise be coupled: where the active context should first become shorter, and where stronger downstream selection or further reduction should occur later in the execution path. This separation motivates treating the first computationally valuable reduction point and subsequent context-selection decisions as distinct components of long-context inference design.