CACHE REPAIR IN HYBRID LLMS HAS A HISTORY

Anonymous authors
Paper under double-blind review

ABSTRACT

In recurrent–attention hybrid language models, updating the same K/V locations can produce different repairs depending on which tokens execute before each update. We study this distinction in position-independent caching (PIC), holding initialization and update locations fixed while executing short predecessor windows. The extra tokens' fresh K/V is neither read during repair nor stored in the final cache. On Qwen3.5-9B/RULER with zero-initialized recurrent matrices, increasing execution from 10.0% to 31.4% of context improves macro score over selected-only scatter by 20.9 percentage points. Installing the resulting K/V alongside scatter's recurrent and convolutional query-start states retains 18.2 points. Both effects hold with last-chunk initialization and replicate under zero initialization on 650 disjoint RULER requests. The control-study gain over naive reuse is smaller (+13.2 points), with an adjusted interval including zero. Additional history harms some OLMo tasks and reduces update coverage under fixed execution budgets. The findings distinguish what a repair stores from the execution needed to produce useful K/V, including at a fixed query-start state; they do not establish a universally beneficial history or an optimal budget allocation.

1 INTRODUCTION

In retrieval-augmented and long-context applications, reused documents often follow different text or occupy different positions. Position-independent caching (PIC) enables reuse beyond an exactly matched prefix by assembling independently prefilled chunks (Yao et al., 2025; Hu et al., 2025; Wang et al., 2026). Because these chunks were computed without their new surrounding context, their cached representations may need repair. Selective recomputation updates a subset of K/V entries to restore useful representations while preserving the computational benefit of reuse.

In recurrent–attention hybrids, the computation producing these entries also depends on recurrent state and convolutional history. Choosing the update locations and an initial state therefore leaves another question open: which preceding tokens should be executed before each update? Even at the same update locations, changing this execution history can change the repaired K/V. It can also change the recurrent and convolutional states from which query processing begins, so a final score alone does not distinguish these effects.

Existing approaches address related aspects of reuse: LinearKV compares initialization under shared PIC selectors (Liu et al., 2026b), HYPIC repairs chunk boundaries (Liu et al., 2026a), and Tail-Replay reconstructs exact-prefix state (Liu et al., 2026c). Our controlled comparison separates executed tokens from updated entries: we vary history at fixed initialization, update locations, and fresh K/V exposure and writes. Factorially crossing the resulting caches and query-start states then isolates K/V utility at a fixed endpoint. An error decomposition organizes these controls.

We compare executing only selected tokens with also executing short predecessor windows. On Qwen/RULER, the added history improves retrieval at the same update locations. Installing the resulting K/V while retaining the selected-only baseline's query-start state preserves much of the gain, showing that the produced K/V is useful even without installing the state reached by the expanded execution. Both effects replicate on disjoint requests. Additional history can nevertheless degrade repair performance on some tasks, and under a fixed executed-token budget it reduces update coverage.

Our contributions distinguish which entries a repair stores from which tokens it executes to produce them:

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.

1. **Extra execution improves K/V at the same update locations.** Without reading or storing the extra tokens' fresh K/V, predecessor execution improves Qwen/RULER retrieval under both zero and last-chunk initialization. The zero-initialized effect replicates on disjoint requests.
2. **The repaired K/V improves retrieval even when the query-start state is unchanged.** Installing the repaired K/V while retaining the baseline's recurrent and convolutional states preserves much of the gain. State-error measurements and local interventions connect the changed computation to K/V production.
3. **Additional history has both failure cases and a coverage cost.** We identify cases where additional history damages successful repair. Under a fixed executed-token budget, recomputing more predecessor tokens reduces how many K/V entries can be updated.

2 Related Work

From prefix sharing to modular reuse. PagedAttention manages K/V storage (Kwon et al., 2023); SGLang shares prefixes (Zheng et al., 2024); Prompt Cache exposes reusable modules (Gim et al., 2024). COMB learns native PIC, Irminsul exploits MLA storage, and DAF decomposes fusion (Zhao et al., 2026; Ma et al., 2026; Wu et al., 2026). While positional coordinates can be adjusted via RoPE (Su et al., 2024) and head counts altered via grouped-query attention (Ainslie et al., 2023), we focus on repairing representation shifts in an unchanged recurrent–attention model.

Selecting and combining independently cached chunks. CacheBlend selects representation deviations (Yao et al., 2025); EPIC links chunk boundaries (Hu et al., 2025), where attention sinks matter (Xiao et al., 2024); ProphetKV uses the query (Wang et al., 2026); CacheClip adds guidance and grouping (Yang et al., 2026). MEPIC, InfoFlow KV and SIFT address memory, information flow and invariance (Wang et al., 2025; Teng et al., 2026; Sanovar et al., 2026). Selector applicability varies (Cestola et al., 2026); we vary execution at the *same update locations*.

Checkpoints and non-prefix state initialization. Marconi studies hybrid-cache admission/eviction (Pan et al., 2025), Sparse Prefix Caching studies checkpoint placement (Shirokikh & Nikolenko, 2026), and DASC compresses state using decay (Yu et al., 2026). HYPIC combines transition composition with chunk-boundary seam windows, including conv warm-up (Liu et al., 2026a). LinearKV's Algorithm 1 executes and updates the selector's positions under a separate initializer, acknowledging omitted transitions as an approximation (Liu et al., 2026b); these positions need not be scattered. Our LinearKV (scatter) baseline uses global top-k; our baseline evaluation also includes span64 (Appendix A). We instead expand execution around fixed updates while preserving fresh reads and writes. These paper-derived adapters evaluate the core algorithmic interventions without reproducing the original engine-integrated serving implementations.

Exact-prefix hidden replay. Tail-Replay reuses exact-prefix K/V and cached attention-output hiddens, replaying a hidden suffix independently through each linear-attention group (Liu et al., 2026c). Exact intermediate inputs change both error sources and cost; our intervention instead executes original tokens against approximate non-prefix assembly.

Cache-channel interventions. Afendulev et al. (2026) cross caches from different contexts to study retrieval and response behaviour. We apply this state-swap design to outputs of sparse repairs of the *same* request, locating changes in K/V utility at a fixed endpoint. The experimental novelty is the controlled repair comparison. Explicit references and intervention metrics guide its interpretation (Zhang & Nanda, 2024).

3 Controlled Cache-Repair Execution

3.1 Update Locations and Executed History

Independently prefilled chunks form a bank $\mathcal{A}$ covering T context tokens. Payload anchors $U \subseteq [T]$ identify update locations; $R \supseteq U$ specifies the original tokens executed in increasing position. Initialization $(S_{\text{init}}, B_{\text{init}}) = f(\mathcal{A})$ supplies recurrent matrices and conv registers. We separately

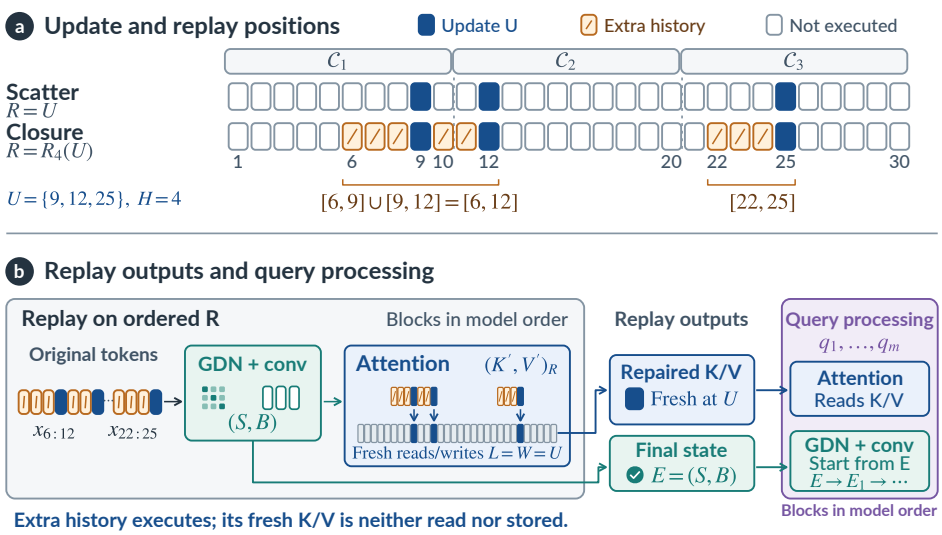


Figure 1: **Fixed updates, expanded execution.** (a) Closure adds three predecessors per update at fixed U; the windows ending at 9 and 12 merge across the chunk boundary. White positions are not executed. (b) State advances through ordered R without gap resets. All of R projects K/V, but fresh reads and writes are restricted to $L = W = U$; the lower strip is the full T-entry cache, gray retaining cached values. Replay outputs the repaired K/V and the final state $E = (S, B)$; query attention reads that K/V and query GDN/conv continues recurrence from E. Blocks execute in model order.

Table 1: Execution contracts. U marks payload anchors; L exposes fresh K/V and W finally writes it. Matched controls select P_m or Q_m at the realized count m. Shading identifies fixed-update closure. Initialization is separate; zero clears matrices but retains the common probe and conv tail.

Policy	Executed	L: exposed	W: written
Selected-only scatter	U	U	U
Target-only closure	R	U	U
Warmup-KV retain	R	U	R
Full-span recompute	R	R	R
Matched scatter	P_m	P_m	P_m
Matched span64	Q_m	Q_m	Q_m

specify the positions L whose fresh K/V is exposed during replay and W whose K/V is finally written. The primary intervention fixes $L = W = U$ while varying R (Table 1). All T attention-cache entries remain resident.

Selected-only execution uses $R = U$ (*scatter*). History expansion uses the union of up to $H - 1$ predecessors per payload:

$$R_H(U) = \bigcup_{u \in U} [\max(1, u - H + 1), u]. \tag{1}$$

We call this diagnostic intervention *closure*. Windows merge across chunk boundaries. Attention uses global causal positions; matrix and conv state advance through ordered R without gap resets, so predecessor execution restores local temporal context before each update. Lower recurrent/conv outputs can thus change later attention-layer K/V at U even when predecessors' K/V is unwritten. A disposable query probe selects U before replay. Figure 1 and Algorithm 1 (Appendix B) specify the private cache.

3.2 Initialization and Path Perturbations

At one normalized-key Gated DeltaNet head,

$$S_t = A_t S_{t-1} + P_t, \quad A_t = \alpha_t(I - \beta_t k_t k_t^\top), \quad P_t = \beta_t k_t v_t^\top. \tag{2}$$

With shared projected inputs/gates, an initialization discrepancy propagates as $\Delta S_t = A_t \cdots A_1 \Delta S_0$ and contracts under the coefficient assumptions in Appendix C.1. Selective execution also changes the transition and input path. The following identity organizes these experimental variables; it does not predict the sign of a quality change.

Proposition 1 (Initialization and execution history)**.** *Write $R = \{r_1 < \cdots < r_M\}$ and $r_0 = 0$. Let $S^\star$ be a dense reference and $\widehat{S}_m$ the replay state after position r_m. For $e_m = S^\star_{r_m} - \widehat{S}_m$,*

$$e_m = A^\star_{r_m} e_{m-1} + \zeta^{\text{gap}}_m + \zeta^{\text{input}}_m. \tag{3}$$

The two perturbations are

$$\zeta^{\text{gap}}_m = A^\star_{r_m}(S^\star_{r_m-1} - S^\star_{r_{m-1}}), \tag{4}$$

$$\zeta^{\text{input}}_m = (A^\star_{r_m} - \widehat{A}_{r_m})\widehat{S}_{m-1} + P^\star_{r_m} - \widehat{P}_{r_m}. \tag{5}$$

The gap term vanishes at consecutive positions. The input term retains changes in projected values and gates, including effects propagated from lower layers. The proof is in Appendix C.2.

Zero and *last-chunk* initialization set $S_{\text{init}} = 0$ or the last chunk's cached matrix, respectively, retaining the same final-chunk conv tail and probe. Last-chunk is the LinearKV initializer; only its selected-only baseline is labelled LinearKV (scatter). Initialization tests the starting discrepancy; history tests subsequent transitions and inputs, whose signed propagated matrices can cancel (Eq. (12) in Appendix C.2). The *gap-start* control moves predecessor windows to gap starts, preserving transition counts (Appendix B); Eq. (14) (Appendix C.2) separates convolution coordinates from projected values.

3.3 K/V Production and Final Cache Update

Replay produces cache K and endpoint $E = (S, B)$. Crossing scatter and closure outputs (K_b, E_b) and (K_c, E_c) tests the utility of each output at a fixed source of the other; each query runs its recurrence from a private cache (Figure 2).

Warmup-KV retain writes additional projected entries while retaining closure's endpoint. Full-span recompute also exposes those entries during replay to downstream representations (Figure 2).

Proposition 2 (Target-only attention invariance)**.** *In a deterministic attention-only model with tokenwise maps, fixed artifact K/V and positional transformations, expanding U to $R \supseteq U$ preserves installed K/V and query outputs in exact arithmetic when only fresh U is exposed and installed.*

Proof sketch. Induct on layers. Embeddings at U agree, yielding equal queries and fresh K/V. Attention at U reads identical fresh entries at U and fixed artifact entries elsewhere under identical causal masks and positional transformations. Tokenwise maps preserve equality. Thus installed K/V agrees, giving identical query outputs in exact arithmetic. □

Appendix C.4 gives the complete argument. Proposition 2 shows that in pure attention, predecessor history is inert under target-only writes; hybrid history sensitivity is thus mediated by recurrence and convolution. We report $r_U = |U|/T$ and $r_R = |R|/T$.

4 Experimental Setup

Models and requests. The primary model is Qwen3.5-9B, with nominal 512-token independent chunks. RULER contexts are nominally 8K, reaching 11.3K actual tokens. OLMo-Hybrid-7B and attention-only Qwen3-8B provide supporting controls. The *control study* uses 325 RULER requests (Hsieh et al., 2024) for initialization, history and output crossings (25 per task, 13 tasks), drawn from a 1,300-request initial discovery cohort. The *replication study* freezes the primary

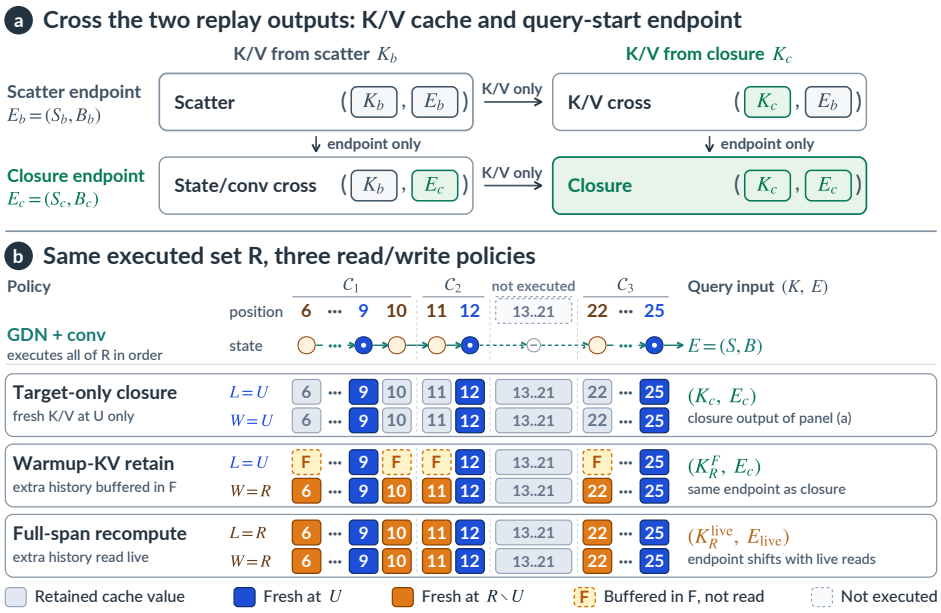


Figure 2: **Separating K/V utility from query-start state.** (a) Crossing the two replay outputs of scatter and closure: horizontal comparisons change only K/V; vertical comparisons change only the query-start endpoint $E = (S, B)$. (b) The executed set R of Figure 1 under the three replay policies of Table 1, which differ only in the fresh read set L and write set W; positions 13–21 never execute. Target-only closure exposes and writes only U. Warmup-KV retain hides the extra history's fresh K/V in buffer F during replay, keeping endpoint E_c, then writes all of R (K_R^F). Full-span recompute exposes it live, altering later layers and the endpoint (K_R^{live}, E_{live}); equal write sets need not hold equal values.

zero-initializer comparisons on 650 disjoint requests. Every contrast pairs conditions within its study; no effect subtracts scores across cohorts. Supporting studies use 96 untruncated LongBench-v2 requests (Bai et al., 2025), 420 uncapped LongBench requests (Bai et al., 2024), and a shared 65-request subset for oracle/source–position controls. LinearKV's up-to-92% recovery uses OLMo-Instruct, ≈20% recomputation and eight-category aggregation; our Qwen control uses 10% updates and thirteen-task macro (Appendices A, F.2).

Query-guided payload selection. The primary ProphetKV-derived probe runs all rendered query tokens against the assembled context, using the last chunk's matrix and conv tail. If $p_{qj}^{\ell h}$ is its attention probability from query token q to context position j, we rank positions by

$$s_j = \frac{1}{|\mathcal{F}|} \sum_{\ell \in \mathcal{F}} \frac{1}{H_\ell Q} \sum_{h=1}^{H_\ell} \sum_{q=1}^{Q} p_{qj}^{\ell h}. \tag{6}$$

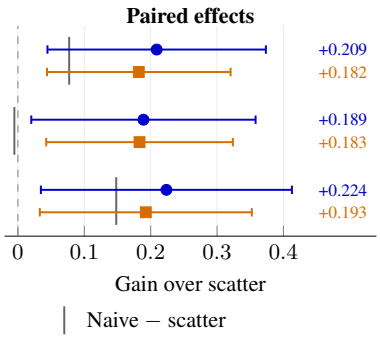
Here $\mathcal{F}$ contains Qwen's eight full-attention layers, with sixteen query heads each; all retained positions are eligible. The probe supplies the same frozen payload across comparisons and is discarded before repair; state changes never enter the replay cache (Appendix A).

Controlled comparisons. The primary payload rate is 0.1 (10%), with $H = 8$; all cells share artifacts and the probe. Output crossings preserve designated cache sources exactly. Nine budget settings cross caps $\rho \in \{0.1, 0.2, 0.4\}$ with $H \in \{4, 8, 16\}$, matching realized counts. Request costs use CPU-resident banks and a host-ready first-token boundary.

Checking the intervention. Studies verify cached tensors, selected indices and prepared endpoints before scoring: unselected K/V stays unchanged, and queries receive designated states through private caches. A separate token-by-token implementation validates execution and installation on 13 requests across 15 policies and 6 boundaries in FP32/BF16 (Appendix G).

Fresh reads/writes: 10%; control execution: 10% → 31.4%

Study / matrix init.	Full	Naive	Scatter	Closure	K/V cross
Control / zero	0.928	0.633	0.556	0.765	0.738
Control / last-chunk	0.928	0.579	0.584	0.773	0.768
Replication / zero	0.883	0.684	0.536	0.759	0.728



With zero initialization, scatter scores below naive in both studies (point estimates).

Figure 3: **Absolute repair quality and gains at fixed updates.** Rows align absolute macro scores with paired gains over scatter. Circles: closure (history expansion); squares: K/V cross (closure K/V at the initializer-matched scatter endpoint); gray ticks: naive reuse minus scatter, the zero-replay reference. Control: 325 shared requests, also used in Table 2; replication: 650 disjoint requests. Bars are familywise 95% intervals, retaining control history/K/V families of six/two and replication's family of four. Absolute scores across studies are separate references.

Estimation and uncertainty. The main estimand is the equal-weight task/subdomain-macro paired difference,

$$\widehat{\Delta} = \frac{1}{J}\sum_{k=1}^{J}\frac{1}{n_k}\sum_{i=1}^{n_k}(s_{ki}^{A} - s_{ki}^{B}). \tag{7}$$

We report nominal 95% and Bonferroni-adjusted intervals within the frozen families in Table 4 (Appendix A). Wild cluster bootstrap-t inference enumerates Rademacher task weights (Cameron et al., 2008): 2^{13} for RULER and 2^{21} for LongBench, eliminating Monte Carlo error. Intervals containing zero leave contrasts unresolved without establishing equivalence.

5 Experiments

5.1 Execution History Improves Fixed-Update Repair

With payload locations, bank and initialization fixed, closure improves control-study macro score by $+0.2090$ with zero matrices and $+0.1891$ with last-chunk matrices; both adjusted intervals are positive (Figure 3, circles). Mean executed fraction grows from 0.1000 to 0.3135 while fresh exposure and final writes remain at 0.1000.

In replication, zero-initialized closure improves by $+0.2237$ and its K/V cross by $+0.1927$. Closure over naive is smaller ($+0.0756$): because isolated scatter degrades representations below naive reuse, part of closure's gain over scatter reflects damage recovery. The control effect survives zero conv initialization ($+0.1911$) and replay-only FP32 ($+0.2159$ vs. BF16 $+0.2190$ on 65 requests; Appendices D.2, G.1).

In the control study, the difference between closure and naive reuse is $+0.1318$, with the nine-comparison adjusted CI $[-0.163, +0.427]$ (Table 6 in Appendix D); scatter versus naive is also unresolved. Thus the history gain includes damage reversal, without establishing superiority over naive reuse.

In the 325-request control cohort, removing any one task leaves a zero-initialized closure–scatter gain of $+0.1464$ to $+0.2264$. Removing the largest-gain task, single-key retrieval 3, leaves $+0.1464$ for closure and $+0.1173$ for its fixed-endpoint K/V cross. These are descriptive sensitivity checks, not additional significance tests (Appendix D).

5.2 Repaired K/V Retains Utility at a Fixed Endpoint

Figure 3's squares fix the query-start endpoint to initializer-matched scatter while installing closure K/V. The benefit persists under both initializers: $+0.1821$ with zero matrices and $+0.1832$ with last-chunk matrices. Table 2 expands the crossing to naive reuse, exposing how cache values and the starting endpoint jointly affect quality.

Table 2: **K/V source and query-start endpoint in the control study.** All cells use the same requests as Figure 3's control rows. Rows set the joint matrix/conv endpoint; columns set K/V. Preparation uses zero matrices and the common terminal conv tail. Naive retains artifact K/V; full prefill scores 0.9284. Queries run normally from private caches. Bold marks row maxima; shading identifies the scatter-endpoint row.

Query state/conv	Naive K/V	Scatter K/V	Closure K/V
Naive	0.6333	0.5889	**0.7629**
Scatter	0.6389	0.5561	**0.7382**
Closure	0.6522	0.5763	**0.7651**

At naive's endpoint, closure K/V improves on scatter K/V by $+0.1740$ (adjusted CI $[+0.0253, +0.3227]$). On single-key retrieval 3, scatter K/V scores 0.0 at all three endpoints while naive K/V scores 1.0 throughout. This crossing localizes that failure to the installed K/V. Endpoint choice matters elsewhere: variable-tracking scatter K/V scores 0.992 at naive's endpoint and 0.512 at its own. Complete tasks and effect families appear in Appendix D; endpoint-cross and interaction contrasts remain unresolved.

Native-reference distances supply a complementary measurement. Closure reduces relative error for selected K/V (from 0.787 to 0.560), recurrent matrices (0.716 to 0.586), and conv buffers (0.730 to 0.458), with adjusted negative error differences. The successful K/V cross retains scatter's higher-error endpoint, separating useful K/V production from improved endpoint fidelity (Appendix D.3).

Replay variants. At 10% payload (Figure 2b), warmup-KV retain scores 0.8031 ($+0.0379$ over closure; 95% CI $[-0.034, +0.110]$) and full-span recompute scores 0.8102 ($+0.0071$ over warmup; 95% CI $[-0.004, +0.018]$). While aggregate scores rise slightly, extra writes harm specific tasks: warmup-KV retain drops single-key retrieval 3 from 0.960 to 0.840 and multi-key retrieval 3 from 0.160 to 0.000. Both aggregate gain intervals include zero (Appendix D). At the matched .2/.4 caps, both variants cost the same as target-only closure within .008 s (Figure 5).

Appendix D.1 tests partial K/V installation; upper closure K/V is still produced through full lower-layer closure execution.

5.3 Inputs and State Errors During Repair

The evidence connects three stages: executed history changes transient computation, that computation changes K/V, and the produced K/V changes query quality at a fixed endpoint. Captured-factor measurements evaluate the first stage by propagating Eq. (3) on a representative sample of 13 requests from the control cohort. At the last selected position, zero-initialized closure reduces layer 12's dense-relative propagated gap norm from 0.459 to 0.272, while the propagated input norm remains 0.659 versus 0.655; layer 24 shows the same pattern. Added execution reduces skipped-transition error while deeper input discrepancies persist (Appendix D.7).

For the second stage, local exchanges of incoming hidden representations, matrices and conv buffers change the next attention layer's K/V, with intervening recurrent pre-states held at closure. These conditional representation changes complement the output crossings; they do not allocate shares of answer quality (Appendix D.8).

A complementary oracle supplies dense context-only conv predecessors on 65 requests, raising zero-initialized scatter from 0.5418 to 0.7562 (Appendix D.5). With zero matrices, closure's added gain falls from $+0.2190$ to $+0.0415$; the interaction is -0.1774 (adjusted CI $[-0.3504, -0.0044]$), and oracle K/V retains $+0.2397$ at ordinary scatter's endpoint (Tables 14–16 in Appendix D.5). The

last-chunk interaction and oracle-K/V intervals include zero. The oracle also changes recurrent inputs; its gain is not a convolution-only share.

Disentangling the exact underlying mechanism remains an open question. A projection-source $\times$ predecessor-position factorial experiment at fixed selected-token execution has all six adjusted intervals containing zero (Appendix D.6, Figure 8). Closure versus gap-start, which matches transition counts before every payload, is also unresolved under both initializers; overlapping predecessor histories limit that contrast (Appendix D.4). These results establish that execution history changes K/V utility at fixed updates. They do not establish which histories consistently help, or separate predecessor proximity, projected values and recurrent effects as explanations of task quality.

5.4 Scope, Update Coverage and Measured Cost

The fixed-update gain does not imply that added history is always beneficial or the best use of an executed-token budget.

Task and response-protocol limits. On the OLMo base checkpoint, closure underperforms scatter on variable tracking (dropping from 0.944 to 0.536) and multivalue retrieval (0.940 to 0.710), while remaining above naive on both. Its 325-request macro gain, $+0.0928$, has an adjusted interval containing zero. Crossings place the multivalue loss in closure K/V and variable tracking in both channels; checkpoint and prompt differences preclude architecture-only attribution (Appendix E.2).

LongBench and LongBench-v2 gains are $+0.0233$ (420 requests) and $+0.0751$ (96 requests); both adjusted intervals include zero. Enforcing a fixed answer prefix in LongBench-v2 also leaves the gain unresolved and raises naive accuracy from 0.0686 to 0.3724 without isolating formatting from semantic recovery (Appendix E.3).

Coverage at fixed executed-token budgets. Algorithm 2 (Appendix B) maximizes the ranked payload prefix whose closure fits $B = \lfloor \rho T \rfloor$; matched controls execute the same realized $|R|$. We extend the sweep to $H \in \{1, 2, 4, 8, 16\}$ at caps $0.1/0.2/0.4$ on the same 325 requests, with a common optimized backend and frozen banks and probe scores (Appendix F.1). Figure 4 shows all 60 settings; history, coverage and selected content change together, so these are allocation comparisons, not fixed-update history effects.

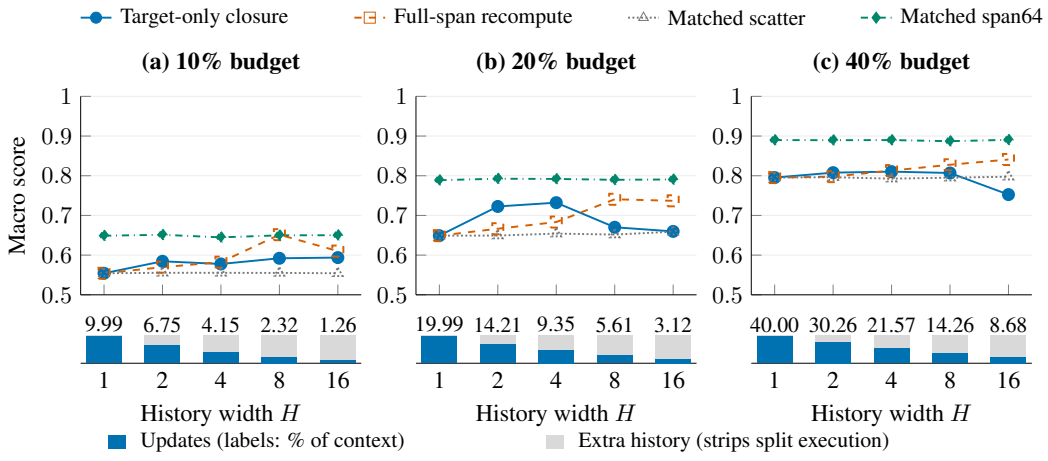
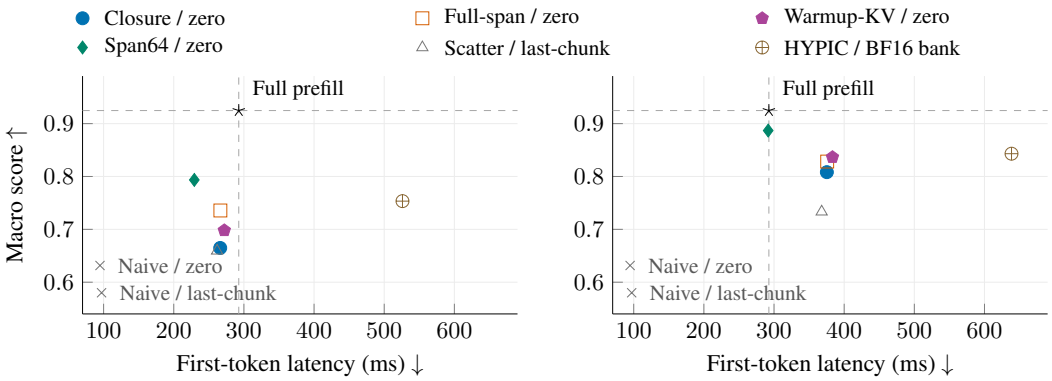


Figure 4: **History trades update coverage for additional execution.** All 60 settings use the same 325 requests and optimized backend. Panels share a score axis; each curve joins the five tested widths for one policy at a fixed cap. Lower strips partition closure's realized execution into updated positions (blue) and extra history (gray); labels give updated context percentages. Increasing H changes both payload size and content. Curves show point estimates; adjusted contrasts remain unresolved (Appendix F.1).

At the 20% cap, increasing H from 4 to 8 reduces updated context from 9.35% to 5.61% and closure score from 0.7322 to 0.6702. Both sweeps leave adjusted policy comparisons unresolved. Target-only writes are a diagnostic, without an established deployment advantage (Appendix F).

Measured request cost. Figure 5 times every replay policy of Figure 2b on the same 312 requests in one campaign (Appendix F.3). At cap .2, target-only closure, full-span recompute and warmup-KV retain take .265–.272 s against .293 s for full prefill, but *Span64*, the full-span contract ($L = W = R$) on contiguous 64-token windows with dense FlashAttention, takes .229 s: the closure family is slower than span64 by +.04 s at cap .2 and +.09 s at cap .4, where it is also slower than full prefill, with all adjusted intervals excluding zero. Full-span recompute differs from closure by at most .0006 s in observed mean latency (adjusted intervals include zero), warmup-KV retain adds +.006–.008 s, and the LinearKV adapter (scatter, last-chunk initializer) is within .007 s of closure. Every closure-family quality contrast against span64 or scatter has an adjusted interval containing zero, though each point estimate favors span64; span20 remains faster than full prefill and span40 unresolved.



(a) Near 20% execution **(b) Near 40% execution**

Figure 5: **Measured quality–latency trade-offs.** Fifteen operating points from one 312-request campaign. Panels group actual execution near 20%/40%; full/naive references repeat in both. Scatter / last-chunk is the LinearKV adapter; HYPIC uses its own initializer and includes transfer/conversion. Other replay points use zero matrices. Times average five-repeat CPU-bank-to-host-ready-first-token request medians, including probe and selection. Points do not establish statistical dominance; all 27 points, actual execution fractions and paired intervals appear in Appendix F.3.

Matching $|R|$ controls executed-token counts, not FLOPs or latency, and sparse replacement limits logical writes while all T context K/V positions remain stored (Appendix B).

6 Discussion and Conclusion

Execution history changes repaired K/V at fixed update locations and initialization. The Qwen/RULER gain persists at a fixed query-start endpoint and replicates on new requests. Measured errors and interventions connect production to utility; predicting when added history helps or harms remains an open question.

OLMo's observed failures are task-specific; natural-language gains remain imprecise, and thirteen RULER clusters limit cross-task precision. Long-context cost and quality, additional architecture families and multi-turn throughput remain unmeasured (measured costs: Appendix F.3).

Implications for evaluating repair. Update locations and initialization alone do not specify how useful K/V is produced. Full/naive references distinguish damage from recovery; matched budgets and measured costs assess whether recovery is worth its work. Fixed-update gains do not choose an allocation policy. Hybrid repair should therefore be evaluated through its executed history, fresh K/V exposure and writes, and query-start state.

AI USE STATEMENT

The authors developed the core research ideas and designed the key experiments. Generative language-model assistants provided limited support with coding and manuscript drafting and editing. The authors take full responsibility for the final code, experiments, analyses, and manuscript.

REPRODUCIBILITY STATEMENT

Appendix A records models, cohorts and provenance; Appendix B specifies replay and exact-budget contracts. Appendices D, E and F retain core controls and all declared budget contrasts. Appendix D.1 adds the descriptive layer-group diagnostic. Appendix G distinguishes checked numerical cases from untested scope. The accompanying supplement contains numerical reports, per-request/task data and payload indices; reduction instructions and checksums are included. Model weights and original benchmark text are separate prerequisites.

REFERENCES

Kirill Afendulev, Alexey Dontsov, Elena Tutubalina, and Anton Korznikov. What attention recalls and recurrence controls in hybrid language models, 2026. URL https://arxiv.org/abs/2609.04434.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, 2023. doi: 10.18653/v1/2023.emnlp-main.298.

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

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3639–3664, 2025. doi: 10.18653/v1/2025.acl-long.183.

Hector Borobia, Elies Seguí-Mas, and Guillermina Tormo-Carbó. Component ablation for efficient hybrid language model architectures: Performance, resilience, and compression implications. *arXiv preprint arXiv:2603.22473v2*, 2026. URL https://arxiv.org/abs/2603.22473v2.

Aleksandar Botev, Soham De, Samuel L Smith, Anushan Fernando, George-Cristian Muraru, Ruba Haroun, Leonard Berrada, Razvan Pascanu, Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot, Johan Ferret, Sertan Girgin, Olivier Bachem, Alek Andreev, Kathleen Kenealy, Thomas Mesnard, Cassidy Hardin, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Armand Joulin, Noah Fiedel, Evan Senter, Yutian Chen, Srivatsan Srinivasan, Guillaume Desjardins, David Budden, Arnaud Doucet, Sharad Vikram, Adam Paszke, Trevor Gale, Sebastian Borgeaud, Charlie Chen, Andy Brock, Antonia Paterson, Jenny Brennan, Meg Risdal, Raj Gundluru, Nesh Devanathan, Paul Mooney, Nilay Chauhan, Phil Culliton, Luiz Gustavo Martins, Elisa Bandy, David Huntsperger, Glenn Cameron, Arthur Zucker, Tris Warkentin, Ludovic Peran, Minh Giang, Zoubin Ghahramani, Clément Farabet, Koray Kavukcuoglu, Demis Hassabis, Raia Hadsell, Yee Whye Teh, and Nando de Frietas. Recurrentgemma: Moving past transformers for efficient open language models, 2024. URL https://arxiv.org/abs/2404.07839.

A. Colin Cameron, Jonah B. Gelbach, and Douglas L. Miller. Bootstrap-based improvements for inference with clustered errors. *The Review of Economics and Statistics*, 90(3):414–427, 2008.

Samuel Cestola, Tianxiang Xia, Zheng Weiyan, Zheng Pengfei, and Diego Didona. An experimental study of kv cache reuse strategies in chunk-level caching systems, 2026. URL https://arxiv.org/abs/2603.20218.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*, volume 2024, pp. 35549–35562, 2024.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 10041–10071, 2024.

In Gim, Guojun Chen, Seung seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. Prompt cache: Modular attention reuse for low-latency inference. In *Proceedings of Machine Learning and Systems*, volume 6, pp. 325–338, 2024.

David Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, 23(1):5–48, 1991. doi: 10.1145/103162.103163.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Proceedings of the First Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Nicholas J. Higham and Theo Mary. Mixed precision algorithms in numerical linear algebra. *Acta Numerica*, 31:347–414, 2022. doi: 10.1017/S0962492922000022.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? In *Proceedings of the First Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Junhao Hu, Wenrui Huang, Weidong Wang, Haoyi Wang, Tiancheng Hu, Zhang Qin, Hao Feng, Xusheng Chen, Yizhou Shan, and Tao Xie. EPIC: Efficient position-independent caching for serving large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 24391–24402, 2025.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 5156–5165, 2020.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP '23)*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165.

Yifei Liu, Juntong Wu, Yang Liu, Junhao Hu, Minghao Li, Xiaoxu Chen, and Weihang Chen. Hypic: Accelerating hybrid-attention llm serving with position-independent caching, 2026a. URL https://arxiv.org/abs/2607.01299.

Yirui Liu, Ruoling Qi, Longwen Wang, Xuaner Wu, Jian Chen, Yuxin Jin, Jiawei Shao, and Xuelong Li. Linearkv: One cached state suffices for position-independent caching in hybrid llms, 2026b. URL https://arxiv.org/abs/2608.11231.

Yirui Liu, Ruoling Qi, Xuaner Wu, Penghang Liu, and Jian Chen. Tail-replay: Escaping the curse of linear attention in prefix caching for hybrid llms, 2026c. URL https://arxiv.org/abs/2608.30310.

Bole Ma, Jan Eitzinger, and Harald Köstler. Irminsul: MLA-native position-independent caching for agentic LLM serving. *arXiv preprint arXiv:2605.05696*, 2026. URL https://arxiv.org/abs/2605.05696.

Rui Pan, Zhuang Wang, Zhen Jia, Can Karakus, Luca Zancato, Tri Dao, Yida Wang, and Ravi Netravali. Marconi: Prefix caching for the era of hybrid llms. In *Proceedings of Machine Learning and Systems*, volume 7, 2025. URL https://proceedings.mlsys.org/paper_files/paper/2025/file/7c180af017258d239bac6248d1eb26ac-Paper-Conference.pdf.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanislaw Wozniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. Rwkv: Reinventing rnns for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14048–14077, 2023. doi: 10.18653/v1/2023.findings-emnlp.936.

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling. In *International Conference on Learning Representations*, volume 2025, pp. 53551–53575, 2025.

Rya Sanovar, Srikant Bharadwaj, Hritvik Taneja, and Moinuddin Qureshi. Sift: Selective-index for fast compute of rag prefill by exploiting attention invariance, 2026. URL https://arxiv.org/abs/2606.09441.

Mikhail Shirokikh and Sergey Nikolenko. Sparse prefix caching for hybrid and recurrent llm serving, 2026. URL https://arxiv.org/abs/2605.05219.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models, 2023. URL https://arxiv.org/abs/2307.08621.

Xin Teng, Canyu Zhang, Shaoyi Zheng, Danyang Zhuo, Tianyi Zhou, and Shenji Wan. Inf oflow kv: Information-flow-aware kv recomputation for long context. In *Proceedings of the 43rd International Conference on Machine Learning (ICML 2026)*, 2026. URL https://openreview.net/forum?id=o8y6CoJsWA.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, Garvit Kulshreshtha, Vartika Singh, Jared Casper, Jan Kautz, Mohammad Shoeybi, and Bryan Catanzaro. An empirical study of mamba-based language models, 2024. URL https://arxiv.org/abs/2406.07887.

Qian Wang, Zahra Yousefijamarani, Morgan Lindsay Heisler, Rongzhi Gu, Bai Xiaolong, Shan Yizhou, Wei Zhang, Wang Lan, Ying Xiong, Yong Zhang, and Zhenan Fan. Mepic: Memory efficient position independent caching for llm serving, 2025. URL https://arxiv.org/abs/2512.16822.

Shihao Wang, Jiahao Chen, Yanqi Pan, Hao Huang, Yichen Hao, Xiangyu Zou, Wen Xia, Chongyang Qiu, Wentao Zhang, and Pengfei Wang. Prophetkv: User-query-driven selective recomputation for efficient kv cache reuse in retrieval-augmented generation. In *Proceedings of the 43rd International Conference on Machine Learning (ICML 2026)*, 2026. URL https://openreview.net/forum?id=SwGwmKIgUb.

Xiabao Wu, Wentao Liu, Yongchao Liu, and Jiajun Zheng. Decoupled attention fusion: Accelerating RAG with efficient KV cache reuse. In *The 21st European Conference on Computer Systems (EuroSys 2026)*, 2026. URL https://arxiv.org/abs/2607.21599. Poster track.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pp. 21875–21895, 2024.

Bin Yang, Qiuyu Leng, Jun Zeng, and Zhenhua Wu. Cacheclip: Accelerating rag with effective kv cache reuse, 2026. URL https://arxiv.org/abs/2510.10129.

Songlin Yang and Yu Zhang. FLA: A triton-based library for hardware-efficient implementations of linear attention mechanism, 2024. URL https://github.com/fla-org/flash-linear-attention.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d13a3eae72366e61dfdc7eea82eeb685-Abstract-Conference.html.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations*, volume 2025, pp. 29687–29707, 2025.

Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. Cacheblend: Fast large language model serving for rag with cached knowledge fusion. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys '25)*, pp. 94–109, 2025. doi: 10.1145/3689031.3696098.

Yanqi Yu, Pingwei Sun, Jianchao Tan, Tao Zhang, Yuchen Xie, Xunliang Cai, and Yao Liu. Dasc: Decay-aware state compression for hybrid linear-attention serving, 2026. URL https://arxiv.org/abs/2608.30386.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/06a52a54c8ee03cd86771136bc91eb1f-Abstract-Conference.html.

Shiju Zhao, Junhao Hu, Jiaqi Zheng, and Guihai Chen. You need an encoder for native position-independent caching. *arXiv preprint arXiv:2602.01519*, 2026. URL https://arxiv.org/abs/2602.01519.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. Sglang: Efficient execution of structured language model programs. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, volume 37, pp. 62557–62583, 2024. doi: 10.52202/079017-2000.

APPENDIX

A SETUP AND PROVENANCE

The supplement contains complete numerical reports, per-request and per-task score files, frozen payload indices, and the previous extended manuscript with its historical tables. This appendix follows the current controlled-execution, output-channel and resource questions; earlier timing and duplicate condition grids remain in the numerical supplement. Historical FAC/BFR, quantization and routing experiments are archived rather than mixed with the current controls. The supplement's manifest gives checksums and the reduction entry points; model weights and the original benchmark text are separate inputs. Displayed means and paired contrasts are rounded independently from full-precision results; subtracting displayed means can therefore differ in the last decimal place.

Models and runtime. The primary checkpoint is Qwen3.5-9B revision `c202236235762e1c871ad0ccb60c8ee5ba337b9a`, with BF16 weights, 24 GDN layers and eight full-attention layers. GDN has 128-dimensional key/value heads, 16 key heads and 32 value heads; full attention has 16 query and four K/V heads. The recorded runtime is `torch 2.13.0+cu130` (Paszke et al., 2019) and `transformers 5.14.1`. Analysis runs use torch recurrent reference kernels, native SDPA attention and explicit query-chunked sparse attention. The timed serving path adds fused kernels (Appendix F.3). Shards run independently on a shared four-GPU H100 80GB HBM3 node, one model replica per GPU. Numerical comparisons are in Appendix G.

Qwen prompts use the benchmark renderer and chat template with thinking disabled. The OLMo-Hybrid-7B base checkpoint uses plain prompts and NoPE attention; Qwen3-8B is the pure-attention control. Model and adapter differences are not architecture-only interventions. The original manifests retain run-specific model revisions and source hashes.

Table 3: Request cohorts used by the retained evidence. Sharing IDs does not imply independent replication or identical tokenized inputs across models.

Study	Requests	Tasks	Role
RULER discovery	1,300	13	Fixed-U history expansion
Qwen same-bank controls	325	13	Output/read-write controls
Replay-budget grid	325	13	Same IDs, 38 conditions
OLMo / pure-attention / chunk heads	325 each	13	Supporting controls
LongBench fixed-payload study	420	21	Uncapped paired/factorial tests
Damage / conv-initialization	325	13	Same original IDs, 12 conditions
New RULER / LongBench-v2 subset	650 / 96	13 / 10	Disjoint requests, five conditions
Replay precision / local S/B	65 / 13	13 / 13	Fixed original-cohort subsets
Initialization / history location	325	13	Same IDs, eleven conditions
Conv-history oracle	65	13	Fixed ranks, fifteen conditions
Source / position factorial	65	13	Same oracle IDs, nine conditions
Aligned baselines / policy remeasurement	312	13	Thirteen / 27 conditions, same IDs
Layer-group K/V crossing	13	13	Reused rank-zero IDs; seven conditions
Expanded budget sweep	325	13	Same IDs, 62 optimized-backend conditions

Bold study labels identify the control and disjoint-request cohorts; shading identifies the expanded budget sweep. Numerical-table emphasis is defined in each caption. Point extrema use displayed precision, retaining ties; they do not imply statistical superiority. Confidence-interval tables bold only effects whose displayed adjusted interval excludes zero.

The discovery cohort has 100 requests per task; the 325-request cohorts have 25 and overlap discovery. LongBench has 20 per task. RULER is nominally 8K, with actual contexts up to 11.3K tokens. The retained LongBench comparison is uncapped; older 2,100-request truncated and 331-request cohorts remain separately identified in the historical supplement.

The layer-group diagnostic reuses the rank-zero requests already used for local mechanism measurements.

Inputs and identity. Chunks are independently prefilled in nominal 512-token segments; a final segment shorter than the four-token conv kernel is merged into its predecessor. The fixed 325-ID selection is `configs/selections/ruler-8k-325-v1.json`; each producing manifest also enumerates its IDs. The common last-chunk-state probe operates on a private cache. Original controls start from zero matrices and the cached final-chunk conv tail; the initialization/history study also tests last-chunk matrices. The probe's mutations never become the replay initializer.

Query score and payload selection. Section 4 gives the primary query score and the shared private-probe contract. Probabilities normalize over the causally available context and query keys; slicing to context does not renormalize them. Aggregation uses float32, with no extra recurrent-score term. No sink or boundary mask is added. Original selected-only runs take $\lceil \rho T \rceil$ global top-score positions via `torch.topk` and sort them for execution. Their exact frozen indices resolve implementation-dependent ties. The hard-cap study instead uses Algorithm 2's explicit stable tie rule and fits expanded execution inside $\lfloor \rho T \rfloor$. The query probe is discarded before replay.

For discovery, all 1,300 paired payloads and closure counts agree, but historical artifact tensor hashes were not retained. The newer controls verify the starting bank, payloads, endpoints and query cache receipts within each study. These checks establish the stated identity contracts; their relation to independent execution validation is discussed in Appendix G.

Baseline implementation lineage. Our single-state replay is a port of an existing paper-derived LinearKV implementation. The original "LinearKV" shorthand used our span64 selector; last-chunk/scatter denotes the global top-k variant. On the current 325-ID subset, the latter's positions, scores and complete answers all reproduce exactly, with macro .5843077. Historical bank tensors lack the newer receipts, so output agreement does not certify bank identity. Earlier HYPIC-style runs and a cached-T/S HYPIC prototype remain in the supplement as existing records, not new replications.

Numerical reproduction. `scripts/generate_compact_appendix.py` validates source score-tree hashes and task means before generating the merged tables and supplement. `scripts/build_paper_v4.sh` regenerates the active evidence and compiles the manuscript; `scripts/check_paper_v4.py` checks the retained structure. The numerical bundle is sufficient to inspect the full result grids; new model execution additionally requires the pinned checkpoints, datasets and execution environment documented in `docs/REPRODUCTION.md`.

Recurrent and hybrid architectures. Linear-attention Transformers admit recurrent evaluation (Katharopoulos et al., 2020). RetNet, RWKV and Mamba use different recurrent constructions (Sun et al., 2023; Peng et al., 2023; Gu & Dao, 2024); Mamba-2 develops structured state-space duality (Dao & Gu, 2024). Gated DeltaNet combines decay with a directional delta rule (Yang et al., 2025). Mamba studies, RecurrentGemma and Samba explore hybrid mixtures (Waleffe et al., 2024; Botev et al., 2024; Ren et al., 2025). Component ablation also reveals model/task-dependent roles (Borobia et al., 2026). These transitions and convolution policies are not interchangeable.

Table 4: **Which family answers which question.** Commas separate families. Roles organize the current account: P = primary mechanism/replication question, S = supporting control, E = exploratory, V = implementation validation, B = aligned baseline comparison. Shared IDs and controls are not independent replications. Archived historical studies retain their original protocols.

Study / cohort	Family sizes and roles	Comparisons and interpretation
Original fixed-U replication	3 (P)	One expansion gain each for Qwen, OLMo and LongBench.
Qwen / OLMo output channels	3, 3 (S)	K/V cross, endpoint cross and their interaction, separately per model.
Qwen / OLMo allocation	4, 3 (S)	Matched execution/write controls; separate model families.
Original Qwen naive reference	9 (E)	Every non-naive condition versus naive reuse; post-hoc reference checks.
History / initializer, 325	**6 (P)**, 2 (S), 4 (E)	Gains/locality and initializer interactions; K/V channels; initializer-matched naive comparisons.
Conv-history oracle, 65	4 (P), 2, 4 (S)	Oracle effects and remaining replay gains; attenuation interactions; K/V-channel effects.
Projection source / position, 65	4 (P), 2 (S)	Within-source position effects, source effect at original positions and interaction; preselected local/original utility and K/V cross.
Naive-source / initial-conv, 325	5, 3 (S)	Damage/recovery output crosses; initial-tail gains and interaction.
New RULER / natural requests	**4 (P)**	Closure and K/V cross on 650 RULER and 96 LongBench-v2 requests.
Replay precision, 65	3 (S)	BF16/FP32 replay gains and their interaction.
LongBench / chunk heads	5, 5 (E)	Separate initializer/selector factorials, retaining their own exploratory correction.
Budget grid, 325	36, 9 (S)	Four cost contrasts at nine settings; full-span recompute minus target-only at every setting.
Expanded budget sweep, 325	60, 15 (S); 12 (E)	Optimized-backend cost/write families; exploratory adjacent-width contrasts. Original families are unchanged.
Aligned baselines, 312	5, 5, 2, 2, 10, 1, 1 (B)	Baseline quality/latency; initializer quality/latency; native latency; HYPIC precision quality/latency.
Replay policies on one clock, 312	12, 4, 12, 2, 8 (B) latency; 4, 12, 2, 8 (B) quality	Closure family versus full prefill, scatter and span64; scatter initializer; variants versus closure.
OLMo full/naive references, 325	3 (E)	Previously observed requests; descriptive failure cases and exploratory aggregate reference contrasts.
Natural fixed-prefix choice, 96	3 (S)	Closure versus scatter/naive and K/V cross versus scatter under the fixed response protocol.

Bold identifies the main history and disjoint-request families; shading identifies the expanded sweep. The six-comparison history family is the main initializer/history test. A shared zero contrast also appears under its original three-comparison replication correction; we retain both provenance and roles rather than choosing the narrower interval. New requests strengthen within-benchmark replication without adding task clusters. Geometry summaries and parsing partitions are descriptive, not additional significance tests. The predeclared layer-group grid adds descriptive development evidence only; it does not alter these families, add significance tests or provide independent replication.

B Replay and Budget Algorithms

Policy names follow Table 1. In historical numerical exports, `live_write_r`, "Live-R" and "in-flight exposure" all denote full-span recompute ($L = W = R$).

Let C_0 contain the assembled context K/V at global positions. Qwen artifact keys are stored before RoPE; assembly applies the model's rotary transform once at chunk offset plus local token index, leaving values unchanged. Cached keys are not unrotated and re-rotated. Replay uses the same original positions, and query positions begin at T (zero-based); OLMo uses NoPE attention.

Recurrent state and conv registers are initialized once, and original tokens at sorted R traverse the full stack. The block operations below include normalization, residual and feed-forward paths. A gap skips transitions and conv inputs; it does not reset state. Standard replay shifts conv registers once per executed token: after the initial tail is displaced, lag j at replay step m uses the live projection from r_{m-j}, even across an original-coordinate gap. Original-coordinate bank lookup is a separate oracle intervention (Appendix C.3). Contiguous predecessors restore local coordinates but need not reproduce dense projected values. Every query still runs recurrence.

Algorithm 1 Full-stack replay with separate full-span recompute and final cache update

Require: Artifacts $\mathcal{A}$, context x, query q, selector π, initializer f, width H, policy p from Table 1

$C_0 \leftarrow \mathrm{Assemble}(\mathcal{A})$; $s \leftarrow \mathrm{Probe}(q, \mathrm{Clone}(C_0))$ with the common last-chunk matrix/conv state

$U \leftarrow \pi(s)$ or the frozen payload; matched policies use P_m or Q_m from Algorithm 2

$R \leftarrow U$ for scatter/matched policies; otherwise $R \leftarrow R_H(U)$

$L \leftarrow R$ if $p =$ full-span, otherwise $L \leftarrow U$

$(S, B) \leftarrow f(\mathcal{A})$; $h \leftarrow \mathrm{Embed}(x_R)$; $C_q \leftarrow \mathrm{Clone}(C_0)$

for layer ℓ in model order **do**

  if layer ℓ is recurrent **then**

    $(h, S^\ell, B^\ell) \leftarrow \mathrm{RecurrentBlock}_\ell(h, S^\ell, B^\ell)$

  else

    $(Q, K', V') \leftarrow \mathrm{Project}_\ell(h, R)$

    $C^{\mathrm{live}} \leftarrow \mathrm{Clone}(C_0^\ell)$; $C_L^{\mathrm{live}} \leftarrow (K', V')_{\mathrm{slots}(L,R)}$

    $h \leftarrow \mathrm{AttentionBlock}_\ell(Q, C^{\mathrm{live}};$ global causal positions $R)$

    $C_q^\ell \leftarrow C^{\mathrm{live}}$; retain $F^\ell \leftarrow (K', V')$ for warmup-KV retain

  end if

end for

if $p =$ warmup-retain **then**

  For each full-attention layer ℓ, update C_q^ℓ with F^ℓ at R after replay

end if

return $\mathrm{QueryAndDecode}(q, C_q, S, B)$

The complete attention cache always retains T positions. Under target-only closure, freshly projected values outside U are neither exposed nor written to cache. Warmup-KV retain captures those values but writes them to the cache only after the unchanged target-only execution, preserving its final (S, B). Full-span recompute exposes them inside each attention layer and can change downstream K/V and state. Output crossing instead combines the complete K/V of one finished preparation with the joint (S, B) of another in a fresh query cache.

Matched-count history location. For one-based payload positions $U = \{u_1 < \cdots < u_m\}$, set $u_0 = 0$ and $k_j = \min(H - 1, u_j - u_{j-1} - 1)$. Closure executes the last k_j positions in each gap before u_j; gap-start executes its first k_j. Both also execute U. Thus they preserve $|R|$ and the number of transitions before every payload, while moving the extra positions. Convolution slots need not all change: overlapping local histories and unchanged payload predecessors limit the intervention coverage measured in Appendix D.4.

Hard-cap selection. Given finite probe scores s, rank positions stably by descending score, breaking ties by earlier position. Let P_k be the first k positions and $B_R = \lfloor \rho T \rfloor$. The selected payload is the largest ranked prefix satisfying $|R_H(P_k)| \le B_R$, not an optimal subset. Nested prefixes have nested closures, so the feasibility predicate is monotone. Infeasible configurations are rejected.

Algorithm 2 Rank-prefix closure and exact replay-count matching

Require: Scores $s_{0:T-1}$, integer $H \geq 1$, $B_R \in [1, T]$
Rank by descending score, then ascending index; $l \leftarrow 1$; $h \leftarrow B_R$; solution $\leftarrow \varnothing$
while $l \leq h$ **do**
  $k \leftarrow \lfloor (l+h)/2 \rfloor$; $C \leftarrow \bigcup_{u \in P_k} [\max(0, u - H + 1), u + 1)$
  if $|C| \leq B_R$ **then**
    solution $\leftarrow (P_k, C)$; $l \leftarrow k + 1$
  else
    $h \leftarrow k - 1$
  end if
end while
Reject if solution $= \varnothing$; otherwise $(U, R) \leftarrow$ solution; $m \leftarrow |R|$
$P_m \leftarrow$ first m positions of the same stable score ranking
$Q_m \leftarrow \mathrm{Span64}(s, m)$ using the procedure below
Verify $|P_m| = |Q_m| = m$; retain unused allowance $B_R - m$
return Sorted U, R, P_m, Q_m and actual count m

Span64 and completion. Visit uncovered peaks in the same stable score order. With $w = \min(64, T)$, the zero-based window for peak p is $[a_p, a_p + w)$, where $a_p = \max(0, \min(p - \lfloor w/2 \rfloor, T - w))$. This shifts the window at context edges to retain its width. Add it if the union fits m; otherwise stop the peak loop. If none fits, select $[0, m)$. Otherwise, scan indices from one past the minimum selected position through $T - 1$, filling uncovered positions (including internal gaps) until the count reaches m. If still short, repeatedly add the highest-scoring unselected neighbour on either side of the selected set, breaking ties by earlier position. This exact-count completion preserves the initial anchors. Algorithm 2 uses zero-based, half-open intervals, equivalent to the main text's one-based closure.

The controls share the bank and score vector but can select different content. Exact request-level m excludes probe/query work and is not latency or FLOP matching. Logical K/V replacement bytes are $2L_A H_{\mathrm{KV}} d_{\mathrm{head}} b|W|$, with element size b and final write set W; this omits full-cache clones, gathers and temporaries. Recurrent work depends on head dimensions and every executed token, while resident attention storage remains proportional to T.

C Mathematical Foundations and Complete Proofs

The analysis separates three objects: the state entering replay, the transitions executed during replay, and the representations supplied to those transitions. This section gives the algebra underlying the controlled interventions. The identities apply to a head or convolution operation; the task-quality consequences are measured by the full-model experiments.

C.1 Gated DeltaNet and initialization propagation

For one head, write the recurrent matrix as $S_t \in \mathbb{R}^{d_k \times d_v}$ and the update as

$$S_t = A_t S_{t-1} + P_t, \qquad A_t = \alpha_t (I - \beta_t k_t k_t^\top), \qquad P_t = \beta_t k_t v_t^\top. \tag{8}$$

Keys are normalized, $\alpha_t = \exp(-\theta_t)$, and $\theta_t \geq 0$. The tested Qwen implementation uses $\beta_t = \mathrm{sigmoid}(b_t)$ and $\theta_{h,t} = \exp(A_h^{\log}) \, \mathrm{softplus}(a_{h,t} + b_h^{\mathrm{decay}})$. It normalizes keys inside the delta-rule kernel. The matrix form in Eq. (8) uses key-by-value orientation; a transposed storage layout represents the same operation.

Proposition 3 (Propagation of an initialization discrepancy)**.** *Let two trajectories share the same coefficients* $(\alpha_t, \beta_t, k_t, v_t)$*, with* $0 \leq \beta_t \|k_t\|_2^2 \leq 2$*. If* $\Delta S_t = S_t - \widehat{S}_t$*, then*

$$\Delta S_T = A_T \cdots A_1 \Delta S_0, \qquad \|\Delta S_T\|_F \leq \Big(\prod_{t=1}^{T} \alpha_t \Big) \|\Delta S_0\|_F = e^{-\sum_{t=1}^{T} \theta_t} \|\Delta S_0\|_F. \tag{9}$$

Proof. Subtracting the two updates cancels P_t, and induction gives the product identity. On the span of k_t, the eigenvalue of $I - \beta_t k_t k_t^\top$ is $1 - \beta_t \|k_t\|_2^2 \in [-1, 1]$; on its orthogonal complement it is one.

The same norm bound holds in one dimension and when $k_t = 0$. Hence $\|A_t\|_2 \le \alpha_t$. Repeatedly applying $\|MX\|_F \le \|M\|_2 \|X\|_F$ proves the result. The empty product is the identity. □

The familiar sufficient conditions $\|k_t\|_2 \le 1$ and $\beta_t \in [0, 2]$ imply the stated spectral condition. We use the product condition directly because finite-precision normalization need not return a key with norm at most one. This result bounds the discrepancy inherited from matrix initialization under shared inputs. Changed lower-layer outputs can change upper-layer projections and gates; the next identity includes these effects.

C.2 Sparse execution and input perturbations

Let $S_t^\star$ be a dense reference trajectory and let $\widehat{S}_m$ be the replay state after original position r_m, with $0 = r_0 < r_1 < \cdots < r_M$. Set $e_m = S^\star_{r_m} - \widehat{S}_m$. Dense and replay coefficients carry $\star$ and $\widehat{\ }$, respectively. The index $r_m - 1$ is the preceding context position; r_{m-1} is the preceding executed position.

Proof of Proposition 1. The reference and replay updates give

$$e_m = A^\star_{r_m} S^\star_{r_m - 1} + P^\star_{r_m} - \widehat{A}_{r_m} \widehat{S}_{m-1} - \widehat{P}_{r_m}.$$

Add and subtract $A^\star_{r_m} S^\star_{r_{m-1}}$ and $A^\star_{r_m} \widehat{S}_{m-1}$. Collecting terms gives Eq. (3), with the explicit gap and input perturbations in Eqs. (4)–(5). The gap term is zero when $r_m = r_{m-1} + 1$. Under shared current coefficients the input term is zero. The identity imposes no sign or alignment on either term. □

For any two trajectories indexed at a common step, subtraction likewise gives

$$\Delta S_t = A_t \Delta S_{t-1} + \xi_t, \qquad \xi_t = (A_t - \widehat{A}_t)\widehat{S}_{t-1} + P_t - \widehat{P}_t. \tag{10}$$

For a contiguous replay run $[a, b]$ beginning at $a = r_k$, reindex replay states by context position within the run. Define its entry state as $\widehat{S}_{a-1} := \widehat{S}_{k-1}$ from the step-indexed notation above, even if $a-1$ was skipped. Taking $S_t = S_t^\star$ gives $\Delta S_{a-1} = S^\star_{a-1} - \widehat{S}_{k-1}$. Consequently, $A^\star_a \Delta S_{a-1} = A^\star_a e_{k-1} + \zeta^{\mathrm{gap}}_k$, including the gap immediately before the run. Let $\Gamma_{p:q} = \sum_{t=p}^{q} \theta_t$, with an empty sum equal to zero.

Corollary 1 (Propagation within a contiguous run)**.** *For $a \le u \le b$, suppose the reference transitions satisfy $\alpha_t = e^{-\theta_t}$, $\theta_t \ge 0$ and $0 \le \beta_t \|k_t\|_2^2 \le 2$, so that $\|A_t\|_2 \le \alpha_t$. The replay coefficients may differ. Then*

$$\|\Delta S_u\|_F \le e^{-\Gamma_{a:u}} \|\Delta S_{a-1}\|_F + \sum_{\tau=a}^{u} e^{-\Gamma_{\tau+1:u}} \|\xi_\tau\|_F. \tag{11}$$

Proof. Unroll Eq. (10) to obtain

$$\Delta S_u = A_u \cdots A_a \Delta S_{a-1} + \sum_{\tau=a}^{u} A_u \cdots A_{\tau+1} \xi_\tau. \tag{12}$$

Apply the triangle inequality and the product norm bound to each term. Signed matrix terms in Eq. (12) can cancel; the upper bound in Eq. (11) discards that cancellation. The entry discrepancy already includes initialization and earlier gaps. □

Thus contiguous execution both advances the inherited state and removes explicit skipped-transition terms inside that run. Sparse replay carries its state across gaps: the initializer's propagation horizon counts every executed transition, while run depth measures the local suffix. Cancellation and input perturbations determine the realized discrepancy. The experiments therefore test quality and conditional output utility directly.

C.3 CONVOLUTION HISTORY: SOURCE AND COORDINATE IDENTITIES

Use lag-indexed channelwise weights W_j, where $j = 0$ is the current projection. At replay token r_m, let z^a be a fixed pre-convolution source bank, $a \in \{\text{local}, \text{dense}\}$, and let $\widetilde{z}_m^{a,p}$ be the current live projection in cell (a, p). The two predecessor maps are

$$\pi_{\text{original}}(m, j) = r_m - j, \qquad \pi_{\text{packed}}(m, j) = r_{m-j}.$$

An unavailable context/stream predecessor is denoted $\perp$ and has $z_\perp^a = 0$. With kernel width K_c, the pre-activation is

$$c_m^{a,p} = W_0 \widetilde{z}_m^{a,p} + \sum_{j=1}^{K_c-1} W_j z^a_{\pi_p(m,j)} + b_{\text{conv}}. \tag{13}$$

The convolution nonlinearity is applied afterwards. The lag indexing reverses the stored PyTorch convolution-weight order.

Lemma 1 (Source and position discrepancy)**.** *For any two cells (a, p) and (b, q) with the same weights,*

$$\begin{aligned} c_m^{a,p} - c_m^{b,q} = W_0(\widetilde{z}_m^{a,p} - \widetilde{z}_m^{b,q}) \\ + \sum_{j=1}^{K_c-1} W_j \big(z^a_{\pi_p(m,j)} - z^b_{\pi_p(m,j)}\big) \\ + \sum_{j=1}^{K_c-1} W_j \big(z^b_{\pi_p(m,j)} - z^b_{\pi_q(m,j)}\big). \end{aligned} \tag{14}$$

Proof. Subtract Eq. (13) for the two cells and add and subtract $\sum_{j=1}^{K_c-1} W_j z^b_{\pi_p(m,j)}$. The bias cancels. The three terms retain current-input, source and coordinate discrepancies, respectively. □

The decomposition fixes an ordering of the two substitutions; it is not a unique allocation of effect shares. In a full hybrid run, lower-layer interventions can change $\widetilde{z}_m$ at later layers. Crossing the two controllable axes and measuring their interaction evaluates this coupled computation without assuming the live projection stays equal across cells.

The earlier cached-tail identity is recovered by distinguishing replay-generated downstream projections $\widetilde{z}$ from cached upstream projections $\bar{z}$. Here t is the local one-based offset after loading a tail at an arbitrary segment boundary; $t \le 0$ indexes that tail, not the global document origin. For $1 \le t < K_c$,

$$c_t^\star - c_t^{\text{tail}} = \sum_{j=0}^{t-1} W_j(z^\star_{t-j} - \widetilde{z}_{t-j}) + \sum_{j=t}^{K_c-1} W_j(z^\star_{t-j} - \bar{z}_{t-j}). \tag{15}$$

Lemma 2 (Cached-history substitution)**.** *Replacing zero history by a cached tail adds $\sum_{j=t}^{K_c-1} W_j \bar{z}_{t-j}$ to the pre-activation. Full-context pre-activation equality holds precisely when the weighted residual in Eq. (15) vanishes.*

Proof. The substituted pre-activation is $\sum_{j=0}^{t-1} W_j \widetilde{z}_{t-j} + \sum_{j=t}^{K_c-1} W_j \bar{z}_{t-j} + b_{\text{conv}}$. The common bias cancels upon subtraction, giving both identities. For example, $K_c = 2$, $t = 1$, $W_0 = 0$, $W_1 = 1$, $z_0^\star = 2$ and $\bar{z}_0 = 1$ leave residual one. The cached representation's provenance therefore remains relevant after its position is specified. Once $t \ge K_c$, only the most recent K_c live projections remain and there is no cached-tail term. □

C.4 ATTENTION-ONLY INVARIANCE

Proof of Proposition 2. Embeddings at U agree. Inductively, equal hidden inputs at U give equal queries and fresh K/V there. Every attention operation reads these same fresh entries and the same fixed artifact entries elsewhere, under identical positional transformations and causal masks. Its outputs at U agree, and tokenwise normalization, residual and MLP maps preserve equality. The final installed K/V and subsequent query outputs consequently agree in exact arithmetic. Extra positions cannot alter U under the stated fresh-read restriction. Recurrent or convolution mixing introduces additional cross-token paths, which are precisely what the intervention studies. □

D PRIMARY MECHANISM EVIDENCE

All ten conditions use the same 325-request Qwen cohort, frozen 10% payload, zero replay matrices and cached terminal conv tail. Starting-bank identity, complete indices, restored query-state receipts and canonical scatter/closure query parity are checked before reduction. A crossed row's replay count describes only its K/V source; its state/conv requires the other preparation, so it is not a single-replay deployment policy.

Table 5: Every completed primary condition. Full prefill and naive reuse remain explicit references. Shading identifies the fixed-update scatter/closure/K/V-cross comparison; no cross-budget ranking is implied.

Condition	Mean r_R	Mean write fraction	Macro
Full prefill	1.0000	1.0000	0.9284
Naive reuse	0.0000	0.0000	0.6333
Scatter	0.1000	0.1000	0.5561
Closure	0.3135	0.1000	0.7651
Closure K/V + baseline state/conv	0.3135	0.1000	0.7382
Baseline K/V + closure state/conv	0.1000	0.1000	0.5763
Warmup-KV retain	0.3135	0.3135	0.8031
Full-span recompute	0.3135	0.3135	0.8102
Matched scatter	0.3135	0.3135	0.7383
Matched span64	0.3135	0.3135	0.8095

Table 6: All primary output-channel, cost and naive-reference contrasts. Nominal 95% and original familywise intervals are distinct. The naive-reference family is exploratory. Bold marks effects and familywise intervals that exclude zero.

Contrast	Δ	95% CI	Familywise CI
Closure vs. scatter (primary family: 3)			
Closure	$\mathbf{+0.2090}$	$[+0.081, +0.337]$	$\mathbf{[+0.056, +0.362]}$
Output channels (mechanism family: 3)			
K/V cross	$\mathbf{+0.1821}$	$[+0.057, +0.307]$	$\mathbf{[+0.035, +0.329]}$
State/conv cross	$+0.0202$	$[-0.002, +0.043]$	$[-0.005, +0.045]$
Interaction	$+0.0067$	$[-0.030, +0.044]$	$[-0.036, +0.049]$
Replay-count controls (cost family: 4)			
Matched scatter − closure	-0.0268	$[-0.128, +0.075]$	$[-0.154, +0.100]$
Matched span64 − closure	$+0.0444$	$[-0.089, +0.178]$	$[-0.099, +0.188]$
Warmup-KV retain − closure	$+0.0379$	$[-0.034, +0.110]$	$[-0.051, +0.127]$
Full-span recompute − warmup-KV retain	$+0.0071$	$[-0.004, +0.018]$	$[-0.006, +0.020]$
Condition vs. naive reuse (exploratory family: 9)			
Full prefill	$\mathbf{+0.2950}$	$[+0.129, +0.461]$	$\mathbf{[+0.031, +0.559]}$
Scatter	-0.0772	$[-0.321, +0.167]$	$[-0.428, +0.274]$
Closure	$+0.1318$	$[-0.066, +0.330]$	$[-0.163, +0.427]$
K/V cross	$+0.1049$	$[-0.104, +0.314]$	$[-0.196, +0.405]$
State/conv cross	-0.0570	$[-0.303, +0.189]$	$[-0.409, +0.295]$
Warmup-KV retain	$+0.1697$	$[-0.040, +0.380]$	$[-0.163, +0.502]$
Full-span recompute	$+0.1768$	$[-0.041, +0.394]$	$[-0.161, +0.515]$
Matched scatter	$+0.1050$	$[-0.127, +0.337]$	$[-0.272, +0.482]$
Matched span64	$+0.1762$	$[+0.013, +0.339]$	$[-0.071, +0.424]$

The original K/V cross improves quality conditional on scatter's endpoint; unresolved state/conv and interaction estimates do not establish state dispensability or complete mediation. The subsequent naive-source extension below distinguishes initial damage from recovery.

	Closure	K/V cross	State/conv	Interaction
cwe	+0.400	+0.324	+0.172	-0.096
fwe	+0.013	+0.013	+0.027	-0.027
niah_multikey_1	+0.200	+0.200	+0.000	+0.000
niah_multikey_2	+0.160	+0.040	+0.000	+0.120
niah_multikey_3	+0.160	+0.080	+0.000	+0.080
niah_multiquery	+0.240	+0.230	+0.000	+0.010
niah_multivalue	+0.160	+0.120	+0.000	+0.040
niah_single_1	+0.000	+0.000	+0.000	+0.000
niah_single_2	+0.000	+0.000	+0.000	+0.000
niah_single_3	+0.960	+0.960	+0.000	+0.000
qa_1	+0.080	+0.120	+0.040	-0.080
qa_2	+0.120	+0.080	+0.000	+0.040
vt	+0.224	+0.200	+0.024	+0.000

Red: negative. White: zero. Blue: positive.
Linear color scale: -1 to $+1$; cells give task means.

Figure 6: **Task heterogeneity of output-channel effects.** The first three columns give task-level gains over scatter; the fourth gives their interaction. Color uses the same symmetric scale throughout. The state/conv cross retains scatter K/V. Interaction is closure minus the two single-output crosses plus scatter. The large single-key retrieval 3 gain and the mostly small state/conv effects are descriptive patterns; cells have no task-level significance interpretation. Paired task-cluster intervals remain in Table 6, and the original numeric grid remains in the supplement.

Single-key retrieval 3 moves from naive 1.000 to scatter .000 and closure/K/V-cross .960, whereas variable tracking improves from naive .024 to scatter .512 and closure .736. Removing any one task leaves a closure macro point difference between +.1464 and +.2264; removing single-key retrieval 3 leaves +.1464 for closure and +.1173 for the K/V cross. These are descriptive task deletions, not new significance tests or held-out replications. The complete deletion table is in the supplement.

Additional writes are not uniformly helpful: warmup-KV retain lowers single-key retrieval 3 from .960 to .840 and multi-key retrieval 3 from .160 to .000 while improving other tasks. Its aggregate interval includes zero. Likewise, equal write sets and a small full-span recompute minus warmup-KV retain macro do not imply equal K/V values or task-wise equivalence.

D.1 Layer-group K/V installation

A predeclared diagnostic uses one previously evaluated rank-zero request per RULER task (13 total). It retains the reference engine, source banks, frozen 10% payload U, $H = 8$, zero replay matrices and common terminal conv tail. The lower/upper attention groups are layers $[3, 7, 11, 15]$/$[19, 23, 27, 31]$ (zero-based). Writing S/C for scatter/closure K/V, the four cells are $\mathsf{B} = (S, S)$, $\mathsf{D} = (C, S)$, $\mathsf{U} = (S, C)$ and $\mathsf{A} = (C, C)$ in lower/upper order. All four use scatter's complete matrix/conv endpoint; only source values at the fixed payload change. Full prefill, naive reuse and closure at its own endpoint provide three additional anchors. All five historical prediction/score controls reproduce, and installed sources and repeated query caches/logits pass identity checks.

The sans-serif cell label U denotes upper-only installation and is distinct from the italic payload set U. Every closure-source tensor is generated by full-stack closure before installation is mixed; this experiment does not execute closure only at upper layers. Layer-wise cache-removal controls also appear in Afendulev et al. (2026). Here all context entries remain resident at every attention layer; the intervention varies their repair-production source at U.

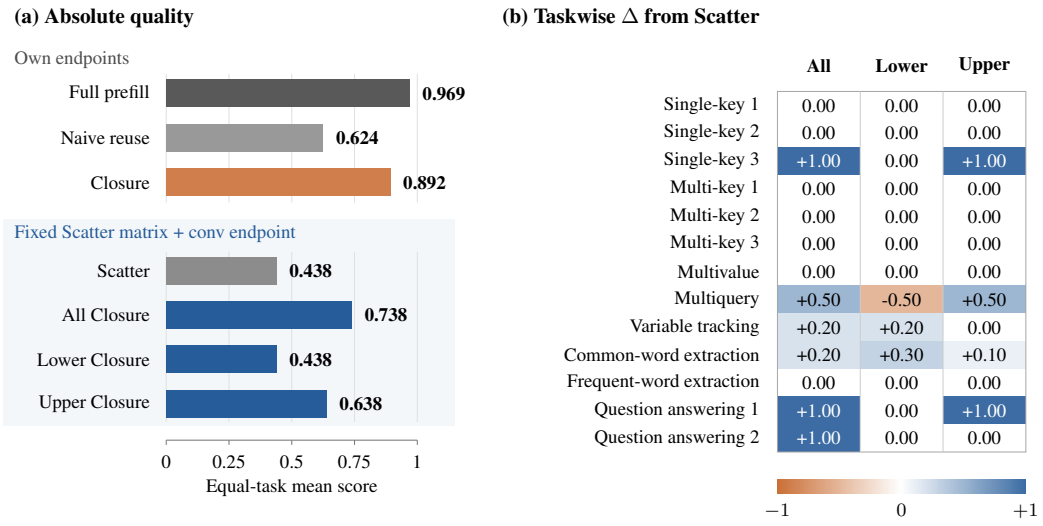


Figure 7: **Whole-bank utility need not survive upper-only installation.** All seven mean scores and all thirteen requests are retained. Each heatmap row is one request; colored columns compare against scatter at the same endpoint, on a common $[-1, 1]$ scale. Colors indicate score differences, not significance. The three reference anchors use their own endpoints.

Mean scores are .438/.438/.638/.738 for B/D/U/A. Lower-only's zero mean effect hides positive and negative request effects. For the QA2 request, B, D, U score zero while A scores one. The mean score interaction $s_A - s_U - s_D + s_B$ is +.100. These observations limit upper-only sufficiency on this set; a thresholded score interaction alone does not establish a representation-compatibility mechanism.

These are descriptive development diagnostics, not new-request replication or new confidence families. Upper-only retains lower scatter K/V and scatter's endpoint, while its upper closure K/V was produced through full lower closure execution. The experiment therefore does not establish a suffix-repair algorithm or a speedup. All 91 scores, nine contrasts and exact source/code receipts remain in `supplement/v19_evidence/`.

D.2 Damage-source and conv-initialization controls

The follow-up reconstructs all 325 original banks and verifies existing prepared endpoints, predictions and scores exactly before evaluating all nine K/V-by-E cells, full prefill and two zero-conv arms (12 conditions). Original endpoint tensors were not retained, so these are regenerated from verified sources rather than assumed interchangeable. The five damage contrasts were fixed before the new outcomes. Table 7 anchors the task examples; Table 2 retains the complete macro crossing.

Table 7: **Repair can damage accurate reuse or improve inadequate reuse.** Two illustrative tasks from the same control. The K/V cross retains scatter's final state/conv. Shading links naive, scatter and closure within each task; bold marks the best repair score in each row.

Task	Full	Naive	Scatter	Closure	K/V cross
Single-key retrieval 3	1.000	1.000	0.000	**0.960**	**0.960**
Variable tracking	0.776	0.024	0.512	**0.736**	0.712

Table 8: All damage-source contrasts. The first two compare a single scatter output against naive reuse; the last two compare closure with scatter at a fixed naive partner. Bold marks effects and adjusted intervals that exclude zero.

Contrast	Δ	Nominal 95% CI	Adjusted CI (5)
Scatter K/V at naive E	-0.0444	$[-0.3274, +0.2386]$	$[-0.4037, +0.3149]$
Scatter E at naive K/V	$+0.0055$	$[-0.0414, +0.0525]$	$[-0.0627, +0.0738]$
Naive–scatter interaction	-0.0384	$[-0.1144, +0.0377]$	$[-0.1300, +0.0533]$
Closure–scatter K/V at naive E	$\mathbf{+0.1740}$	$[+0.0519, +0.2961]$	$\mathbf{[+0.0253, +0.3227]}$
Closure–scatter E at naive K/V	$+0.0133$	$[-0.0149, +0.0415]$	$[-0.0216, +0.0483]$

Closure K/V improves on scatter K/V at naive E, but the other four adjusted intervals include zero. Task-specific failures therefore do not establish general macro harm. At naive E, either scatter or closure K/V gives .992 on variable tracking, whereas their own endpoints give .512/.736. This conditional sensitivity rules out interpreting similar aggregate crosses as endpoint irrelevance.

Table 9: Replay conv-initialization factorial, all 325 requests. Matrix initialization stays zero; the last-chunk probe and its frozen payload are unchanged. The two gains and their interaction form one three-comparison family. Bold marks effects and adjusted intervals that exclude zero.

Conv initialization	Scatter	Closure	Δ	Adjusted CI (3)
Cached tail	0.5561	0.7651	$\mathbf{+0.2090}$	$\mathbf{[+0.0559, +0.3622]}$
Zero	0.5708	0.7618	$\mathbf{+0.1911}$	$\mathbf{[+0.0329, +0.3493]}$
Zero–cached-tail interaction	—	—	-0.0179	$[-0.0657, +0.0298]$

The zero-conv gain remains positive after adjustment, but its difference from the terminal-conv gain is unresolved, not equivalent. This changes the initial tail only: original-position gaps still compress conv adjacency during scatter. It is not a conv-history-controlled or matrix-initialization experiment.

D.3 Reference fidelity

The diagnostic reproduces the original bank and endpoint hashes, retaining their quality scores. For endpoint $X^a_{\ell,c}$ and native dense context-only reference $X^*_{\ell,c}$ rounded to the compared dtype by Π_a, relative error is

$$e^a_{\ell,c} = \frac{\|X^a_{\ell,c} - \Pi_a X^*_{\ell,c}\|_F}{\|\Pi_a X^*_{\ell,c}\|_F}.$$

Norms use float64. K/V error uses common U; matrix/conv error uses final pre-query buffers. Values average layers then tasks equally; full data are in the supplement.

Table 10: Reference-distance summary on the verified source endpoints. State fidelity and conditional task utility are different measurements. Bold marks column minima for errors; shading identifies the scatter/closure error comparison.

Condition	Mean r_R	Source quality	K/V error	Matrix error	Conv error
Naive ($S = 0$)	0.0000	0.6333	0.7913	1.0000	0.5556
Scatter	0.1000	0.5561	0.7869	0.7156	0.7295
Closure8	0.3135	0.7651	**0.5601**	**0.5856**	**0.4576**

All nine original fidelity contrasts and the complete layer curves remain in the numerical supplement, with their original familywise adjustment.

Closure reduces all three mean errors, yet its K/V cross remains useful with scatter's less faithful endpoint. Scatter also has smaller matrix error but lower quality than naive reuse. Final distances do not measure the transient matrix/conv contributions during replay or predict a task-quality bound.

D.4 MATRIX INITIALIZATION AND HISTORY LOCATION

The eleven-condition extension uses the original 325 requests and frozen U. Zero/last-chunk matrix initialization is crossed with scatter, near and gap-start replay; the conv tail, query and fresh-U exposure/writes are fixed. Near and gap-start execute equal transition counts before every payload, although short gaps force shared positions. All original matching banks, endpoints and predictions reproduce exactly. Tables 11 and 12 retain all scores and six primary contrasts. All four exploratory comparisons against initializer-matched naive reuse remain unresolved after adjustment; their exact intervals are in `supplement/history/tables/history_damage.tex`.

Table 11: **Initialization and history on the same 325 requests.** Near and gap-start match execution counts before every payload and write the same U. All cells share the bank, terminal conv tail and selector. The K/V cross uses near K/V and its initializer-matched scatter endpoint; its r_R describes the K/V source, not a deployable single preparation. Bold marks the best repair score within each initializer; shading highlights their common update fraction.

Execution	r_U	r_R	Zero	Last-chunk
Full prefill	1	1	0.9284	0.9284
Naive reuse	0.000	0.000	0.6333	0.5790
Scatter	0.100	0.100	0.5561	0.5843
Closure	0.100	0.313	**0.7651**	**0.7734**
Gap-start replay	0.100	0.313	0.7563	0.7475
Closure K/V, scatter E	0.100	0.313	0.7382	0.7675

Table 12: **All six declared history/initializer contrasts.** Interactions are last-chunk minus zero. Shared zero controls anchor this extension; they are not an independent replication. Bold marks effects and familywise intervals that exclude zero.

Contrast	Δ	Familywise 95% CI
Closure − scatter, zero	$\mathbf{+.2090}$	$\mathbf{[+.0445, +.3735]}$
Closure − scatter, last-chunk	$\mathbf{+.1891}$	$\mathbf{[+.0201, +.3580]}$
Closure − gap-start, zero	$+.0088$	$[-.0366, +.0543]$
Closure − gap-start, last-chunk	$+.0258$	$[-.0106, +.0623]$
Initializer interaction: near − scatter	$-.0199$	$[-.0959, +.0560]$
Initializer interaction: near − gap-start	$+.0170$	$[-.0332, +.0673]$

An unweighted positional audit of the same 325 requests finds that near-to-gap-start changes 27.3% of conv-history slots at payloads and affects 33.2% of payloads. Slot mismatch against original predecessors is 57.96% for scatter, 0.25% for near and 27.57% for gap-start. The initial terminal conv tail is included. These post-hoc descriptors track positions, not activation identity or quality mediation. Probe vectors were not retained, so importance-weighted coverage is unavailable.

Table 13: The separate two-comparison K/V-channel family. Each cross retains its initializer-matched scatter endpoint. Bold marks effects and familywise intervals that exclude zero.

Contrast	Δ	Nominal 95% CI	Familywise 95% CI
K/V cross − scatter, zero	$\mathbf{+.1821}$	$[+.0569, +.3073]$	$\mathbf{[+.0438, +.3204]}$
K/V cross − scatter, last-chunk	$\mathbf{+.1832}$	$[+.0557, +.3107]$	$\mathbf{[+.0427, +.3238]}$

D.5 ORIGINAL-POSITION CONV-HISTORY ORACLE

The 65-request subset and four primary, two interaction and four channel contrasts were fixed before examining new history-study outcomes. Every shared bank, endpoint and prediction reproduces the 325-request study. At each replay position and layer, the oracle substitutes the preceding conv inputs

with dense context-only projections; the current input remains sparse. Its final conv buffer follows the last replay position. Query/decode receive no oracle inputs. Dense projections can encode missing contextual information, and transient matrix states still change, so this is not an isolated adjacency or matrix-only test.

Table 14: **Four declared conv-history contrasts, 65 requests.** Each initializer compares ordinary and oracle execution at the same U. Near/scatter execute .313/.100 of the context on average. Intervals retain the four-comparison adjustment; complete scores and supporting families are in Appendix D.5. Bold marks effects and familywise intervals that exclude zero.

Contrast	Δ	Familywise 95% CI
Oracle − ordinary scatter, zero	$\mathbf{+.2144}$	$\mathbf{[+.0178, +.4109]}$
Oracle − ordinary scatter, last-chunk	$+.1662$	$[-.0047, +.3370]$
Closure − scatter with oracle, zero	$+.0415$	$[-.0248, +.1079]$
Closure − scatter with oracle, last-chunk	$+.0385$	$[-.0219, +.0988]$

Table 15: Both declared conv-history interactions: the oracle near-minus-scatter effect minus the ordinary effect. Each retains the two-comparison adjustment. Bold marks effects and familywise intervals that exclude zero, including negative interactions.

Contrast	Δ	Nominal 95% CI	Familywise 95% CI
Conv-history interaction, zero	$\mathbf{-.1774}$	$[-.3313, -.0236]$	$\mathbf{[-.3504, -.0044]}$
Conv-history interaction, last-chunk	$-.1208$	$[-.2900, +.0485]$	$[-.3087, +.0672]$

Table 16: All four declared oracle K/V-channel contrasts. The query starts from the same ordinary scatter endpoint within each initializer. Bold marks effects and familywise intervals that exclude zero.

Contrast	Δ	Nominal 95% CI	Familywise 95% CI
Oracle-scatter K/V gain, zero	$\mathbf{+.2397}$	$[+.0749, +.4046]$	$\mathbf{[+.0356, +.4438]}$
Oracle-scatter K/V gain, last-chunk	$+.1769$	$[+.0252, +.3287]$	$[-.0138, +.3677]$
Closure − scatter oracle K/V, zero	$+.0238$	$[-.0223, +.0699]$	$[-.0284, +.0761]$
Closure − scatter oracle K/V, last-chunk	$-.0115$	$[-.0592, +.0362]$	$[-.0620, +.0389]$

The full/naive references and all 15 condition scores are retained in `supplement/conv_oracle/tables/oracle_scores.tex`; all ten declared contrasts remain here. Crosses use the corresponding ordinary scatter endpoint, and dense activation construction is additional oracle work.

The source–position study in Appendix D.6 separates the two controllable inputs to the oracle. Retaining the required BF16 pre-conv projections costs about 3 GiB per 8,192-token request; chunk-boundary tails alone cannot supply arbitrary selected-token predecessors.

D.6 Crossing projection source and predecessor position

The fixed protocol uses the same 65 original ranks 0–4 per task, with zero matrices, the original 10% payload and four cells: local/packed, local/original, dense/packed and dense/original. The local bank captures complete pre-conv Q/K/V projections during independent chunk construction; the dense bank comes from native context-only execution. Both exclude the query and answer.

All four cells execute $R = L = W = U$. Original lookup uses each token's preceding $K_c - 1$ context positions; packed lookup uses the preceding selected positions. Missing predecessors in either coordinate system are zero, and the current projection remains the live sparse value. Under the oracle's tail rule, dense/original reproduces the existing zero/oracle/scatter endpoint and answer exactly. Ordinary live scatter, with its terminal warm-start tail and live previous projections, is a separate reference rather than the local/packed cell.

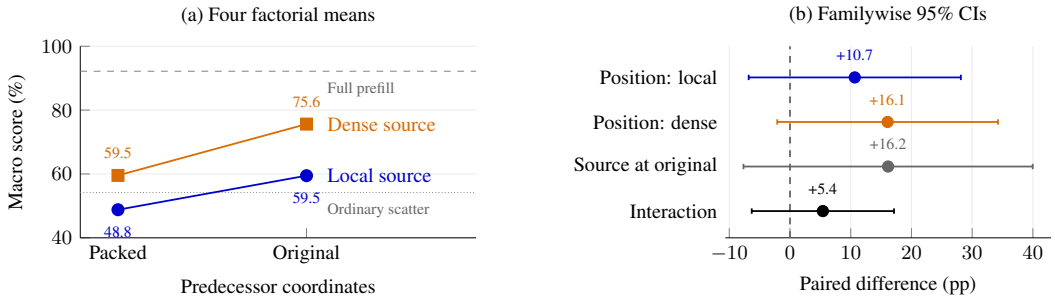


Figure 8: **Projection source and predecessor position at fixed selected-token execution.** All 65 requests, zero matrices. (a) Orange squares use dense-context projections; blue circles use independent-chunk projections (local source). Each line joins packed and original predecessor lookup. The gray dashed line marks full prefill (92.2%); the dotted line marks ordinary scatter (54.2%). Both are separate references from the four cells. (b) Four primary paired contrasts in percentage points (pp), with familywise 95% intervals. Source is dense minus local at original positions; interaction is the dense position effect minus the local position effect. All four intervals include zero, leaving source and position contributions unresolved. Full results are in Tables 17–18.

The four-comparison primary family contains the position effect within each source, the dense-minus-local effect at original positions, and their interaction. A separate two-comparison family tests local/original versus ordinary scatter and its preselected K/V cross at scatter's matrix/conv endpoint. The cross was specified before new outcomes. The current live projection can change across full-model cells because lower-layer interventions propagate; Eq. (14) retains this term explicitly.

Local-source scores are 0.4879/0.5946 under packed/original lookup; dense-source scores are 0.5954/0.7562. The local position effect is +.1067 (familywise CI $[-.0681, +.2814]$); the source–position interaction is +.0541. All effects are paired on this same cohort.

Table 17: Complete source–position operating points on the fixed 65 requests. All four factorial cells and the preselected K/V cross use the same selected-token execution count; near is a separately labelled history reference. Shading isolates the four source/position cells; bold marks their largest point score.

Condition	Macro score
Full prefill	0.9218
Naive reuse	0.5726
Ordinary scatter	0.5418
Near replay	0.7608
Local / packed	0.4879
Local / original	0.5946
Dense / packed	0.5954
Dense / original	**0.7562**
Local/original K/V, scatter E	0.6815

Table 18: Every frozen source–position contrast. The four primary and two supporting comparisons retain separate familywise adjustments. Bold, if present, marks effects and adjusted intervals that exclude zero.

Contrast	Δ	Nominal 95% CI	Familywise 95% CI
Source/position (four comparisons)			
Position: local	$+.1067$	$[-.0418, +.2552]$	$[-.0681, +.2814]$
Position: dense	$+.1608$	$[+.0064, +.3152]$	$[-.0210, +.3425]$
Source at original	$+.1615$	$[-.0260, +.3491]$	$[-.0767, +.3997]$
Interaction	$+.0541$	$[-.0356, +.1438]$	$[-.0631, +.1713]$
Local utility (two comparisons)			
Local/original − scatter	$+.0528$	$[-.1814, +.2870]$	$[-.2330, +.3386]$
Local/original K/V − scatter	$+.1397$	$[-.1094, +.3889]$	$[-.1391, +.4186]$

The complete source–position study reproduces 260 historical reference outputs and all 65 dense/original endpoints and answers. Original-position lookup raises both source-specific point estimates, and local/original K/V at scatter's endpoint scores .6815. All four primary and two supporting adjusted intervals include zero, leaving source and position effects unresolved as explanations of task quality. Both interventions can also alter transient recurrent computation. The exact source banks, predecessor maps, task effects and comparison families are retained in the numerical supplement.

D.7 Measuring initialization, gap and input terms

We instrument one existing rank-zero request per RULER task, at Gated DeltaNet layers 0/12/24 and the first, middle and last selected positions. The four native preparations cross zero/last-chunk matrices with scatter/closure, preserving the historical payload and artifact bank. Native executions supply normalized keys, values, gates and actual selected K/V at the next full-attention layers 3/15/27. This 13-request diagnostic measures state errors rather than task-quality effects.

For each captured trajectory, a separate FP64 scan constructs the terms of Proposition 1. Set $I_0 = S_0^\star - \widehat{S}_0$ and $G_0 = J_0 = 0$, then propagate

$$\begin{aligned} I_m &= A_{r_m}^\star I_{m-1}, \\ G_m &= A_{r_m}^\star G_{m-1} + \zeta_m^{\mathrm{gap}}, \\ J_m &= A_{r_m}^\star J_{m-1} + \zeta_m^{\mathrm{input}}, \qquad e_m = I_m + G_m + J_m. \end{aligned} \tag{16}$$

The initialization term concerns the starting *matrix* and propagates through dense-reference coefficients. A zero matrix term does not eliminate conv-initialization effects, and initializer-induced changes to later coefficients enter J_m. These terms are an algebraic attribution of captured errors, not the total effects of independent model interventions.

For $e_m \neq 0$, we report signed alignment $q_X = \langle X, e_m\rangle_F / \|e_m\|_F^2$, $X \in \{I_m, G_m, J_m\}$, and $\kappa = (\|I_m\|_F + \|G_m\|_F + \|J_m\|_F)/\|e_m\|_F$. Signed alignments sum to one but can be negative or exceed one; $\kappa \geq 1$ measures how much component norms exceed their resultant. Opposing and nonparallel directions both raise this ratio. Undefined ratios remain missing. We first average positions within a request, then weight the thirteen requests equally.

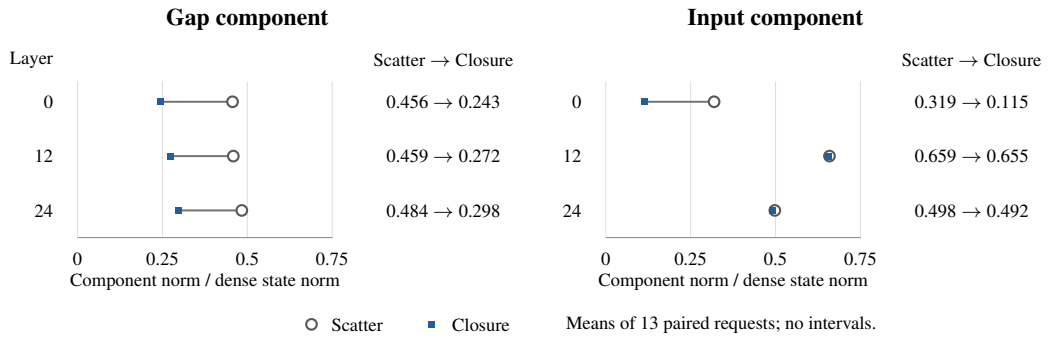


Figure 9: **History expansion reduces gap error but leaves deeper input error.** Mean propagated component norms relative to the dense state, under zero matrix initialization at the *final selected position*, on thirteen requests. Each segment joins scatter to closure at the same layer. These measurements are a subset of the diagnostic, with a different position average from the full four-arm table; they do not measure task-quality mediation or choose a history width.

The complete four-arm grid, averaged over first/middle/last selected positions, remains in `supplement/v12_mechanism/state_terms.tex`. It retains q_I, q_G, q_J, κ, relative FP64 matrix error e_S against the dense scan, and combined selected-key/value relative L2 error e_{KV} against actual dense K/V.

At the final selected position, zero-initialized closure reduces the propagated gap norm relative to dense state from .4564 to .2434 at layer 0, .4586 to .2723 at layer 12, and .4837 to .2978 at layer 24. The input term falls from .3191 to .1148 at layer 0 but remains .6589/.6546 at layer 12 and .4980/.4916 at layer 24. Thus reducing skipped-transition discrepancy leaves substantial deeper-layer input discrepancy. For per-request means over all three sampled positions, state- and K/V-error reductions have mixed descriptive Pearson correlations across the thirteen requests ($r = .096/.313/-.186$ at layers 0/12/24 under zero initialization); these data do not establish a scalar quality predictor.

Across 195 layer/source groups, $\max \beta \|k\|_2^2 = 1.011401 < 2$ satisfies Proposition 3's effective spectral condition. The captured factors are reconstructions, not observed interior full-kernel states; reconstruction residuals and index checks remain in the supplement.

Actual selected K/V comes from unchanged model preparations: its association with state error describes coupled computation, not a proven cause of task failure. The supplement retains every position, including the last-chunk/layer-12 request where closure slightly increases state error.

D.8 Signed local hidden/matrix/conv effects

On the same thirteen rank-zero requests, first, middle-by-position and last payloads at layers 0, 12 and 24 define 234 local intervention points. At each point, the incoming hidden representation H, matrix S and conv register B are each taken from scatter or closure (eight cells); captured projected factors reconstruct the preceding states with the production update, and reconstructed endpoints match captured endpoints exactly. Anchored inclusion–exclusion gives seven vector differences (H, S, B, HS, HB, SB, HSB) whose sum is the all-closure-source minus all-scatter-source local output. Propagation to the next full-attention K/V holds the intervening recurrent layers' pre-states at closure. Thus this is a local source intervention, not a whole-stack difference or the initialization/gap/input decomposition above.

Table 19: **Signed local alignment and norm inflation.** q_A projects a local component onto the total vector difference; C is the sum of its seven component norms divided by the total norm. The 78 all-zero first-position differences are excluded from ratio averages and retained as undefined in the records. Bold marks the largest absolute single-component alignment per row; shading marks negative alignments, not significance.

Layer	Output	q_H	q_S	q_B	q_{HS}	q_{HB}	q_{SB}	q_{HSB}	C
0	next K/V	0.000	**0.540**	0.440	0.000	0.000	0.020	0.000	1.549
0	mixed	0.000	**0.559**	0.431	0.000	0.000	0.010	0.000	1.518
12	next K/V	**0.914**	0.059	0.038	−0.013	0.002	0.003	−0.003	1.851
12	mixed	**0.528**	0.364	0.118	0.002	−0.006	−0.004	−0.003	3.199
24	next K/V	**0.993**	−0.000	0.006	0.001	0.000	0.000	−0.001	1.181
24	mixed	**0.656**	0.152	0.136	0.027	0.024	0.002	0.002	2.231

For each layer and output, the two nonzero positions are averaged within each request, then equally across requests. At layers 12/24, hidden-input alignment rises from .528/.656 in the mixed output to .914/.993 in the next K/V; norm inflation falls from 3.199/2.231 to 1.851/1.181. All 26 nonzero next-K/V points at each of those layers contain a negatively aligned component. Component norms therefore cannot be added as effect shares. These local alignments neither predict held-out task quality nor measure global causal mediation.

Larger discovery cohort. On 1,300 requests, fixed-U gains are +.192/+.237/+.261 for widths 4/8/16. All eleven non-tied task clusters favour each expansion. Complete depth estimates and intervals remain in the supplement; these historical runs lack artifact tensor hashes. Increasing width beyond a single convolution kernel changes recurrent transitions, lower-layer hidden inputs and the effective history of stacked convolutions together. The curve therefore measures history-width sensitivity, without isolating a recurrent-only contribution or selecting a budget-optimal width.

E Robustness and Scope

Supporting controls vary model, workload, selector or matrix initialization. Overlapping requests are not independent replications. Complete condition scores remain in `supplement/full_tables/olmo_controlled_v4.tex`, `supplement/full_tables/longbench_factorial_scores.tex` and `supplement/full_tables/chunk_head_scores.tex`; all effect families, including negative results, remain below.

E.1 Disjoint, outcome-blind request cohorts

The five-arm protocol, $H = 8$, selections and four primary contrasts (two per workload) were frozen before new model outcomes. An audit resolves 186,669 prior sample records, including full-prefill evaluations; no unused LongBench-v1 requests remain. The natural workload therefore uses LongBench-v2 (Bai et al., 2025), not a relabelled v1 replication. New cohorts have zero exact or whitespace-normalized request overlap with audited prior requests and no within-cohort duplicates; document overlap and model-training contamination are not ruled out.

RULER uses seeds 20260909/20260910, 25 requests per task/seed, 13 tasks and an 8,192-token generator cap including its decode reservation (actual context 5,790–8,123). The official generator is pinned at `ab17b7853df4e0a30b78cd5d2b463ac7dff6ee13`; QA question offsets 1000/1025 avoid previously used questions. The 96 natural requests cover ten subdomains in four language domains, excluding code and structured-data domains; complete rendered inputs must fit 32,768 tokens, with no truncation. Subdomains require at least five eligible requests and contribute at most 20, selected by a fixed hash order. The pinned data/code revisions, salt and token hashes are in the frozen selection manifest. This is a length/domain subset, not whole-LongBench-v2 performance.

Both workloads use zero matrices and common terminal conv tails. Last-chunk initialization was outside this frozen five-arm protocol; its control-study effects have not been replicated on disjoint requests. Payload rates are .1/.2; mean closure replay fractions are .3137/.5391. Each request runs full, naive, scatter, closure and closure-K/V/scatter-E. RULER seeds are combined within each of 13

task clusters, not treated as 26 independent clusters. Natural subdomains receive equal weight despite unequal counts. All five scores and every task are in the supplement.

Table 20: **Frozen independent requests.** Equal task/subdomain weighting; K/V cross uses closure K/V and scatter E. Payload is .1 for RULER, .2 for the natural subset. Full and naive references are retained. Bold marks the best repair score per cohort; shading identifies the closure and K/V-cross cells.

Cohort	Full	Naive	Scatter	Closure	K/V cross
New RULER (650)	0.8828	0.6839	0.5358	**0.7595**	0.7285
LongBench-v2 subset (96)	0.4337	0.0686	0.2917	**0.3668**	0.3486

Table 21: Complete held-out primary family. Both natural-language adjusted intervals include zero. These four comparisons were not replaced after observing results. Bold marks effects and adjusted intervals that exclude zero.

Condition minus scatter	Δ	Nominal 95% CI	Adjusted CI (4)
New RULER, closure	$\mathbf{+0.2237}$	$[+0.0715, +0.3759]$	$\mathbf{[+0.0346, +0.4128]}$
New RULER, K/V cross	$\mathbf{+0.1927}$	$[+0.0631, +0.3223]$	$\mathbf{[+0.0330, +0.3524]}$
LongBench-v2 subset, closure	$+0.0751$	$[+0.0032, +0.1470]$	$[-0.0108, +0.1610]$
LongBench-v2 subset, K/V cross	$+0.0569$	$[+0.0059, +0.1079]$	$[-0.0103, +0.1242]$

Natural evaluation uses the official direct-answer prompt/extractor and greedy decoding capped at 128 tokens, not the upstream temperature-.1 protocol. Complementing the naive/closure extraction counts in Appendix E.3, unparsed full/scatter/K/V-cross responses number 16/20/16 out of 96. A post-hoc arithmetic partition with unchanged task denominators attributes +.042821 of each improvement to requests where both answers parse, and +.032273/+.014091 to newly parsed answers for closure/K/V-cross. These sum to the full macro differences; they are not causal strata, semantic-accuracy estimates or new tests. Preserve the official scores and format uncertainty.

Table 22: Complete controlled effect families for the supporting studies. OLMo's closure row belongs to the across-study primary family of three; its next three rows form a separate mechanism family. LongBench's separately designated primary pair is reported below with its three-comparison adjustment. Bold, if present, marks effects and familywise intervals that exclude zero.

Contrast	Δ	95% CI	Familywise CI
OLMo: separate primary / mechanism families of 3			
Closure	$+0.0928$	$[-0.070, +0.255]$	$[-0.098, +0.284]$
K/V cross	$+0.0924$	$[-0.037, +0.222]$	$[-0.065, +0.249]$
State/conv cross	-0.0098	$[-0.059, +0.039]$	$[-0.067, +0.047]$
Interaction	$+0.0103$	$[-0.028, +0.049]$	$[-0.042, +0.062]$
OLMo: cost family of 3			
Matched scatter − closure	-0.0127	$[-0.082, +0.057]$	$[-0.099, +0.073]$
Matched span64 − closure	$+0.0857$	$[-0.034, +0.206]$	$[-0.061, +0.233]$
Full-span recompute − closure	$+0.0619$	$[-0.051, +0.175]$	$[-0.071, +0.195]$
LongBench: exploratory family of 5			
Closure gain, zero	$+0.0233$	$[+0.002, +0.044]$	$[-0.005, +0.052]$
Closure gain, last-chunk	$+0.0270$	$[+0.002, +0.052]$	$[-0.005, +0.059]$
Zero − last-chunk, base	$+0.0015$	$[-0.018, +0.021]$	$[-0.025, +0.028]$
Zero − last-chunk, closure	-0.0022	$[-0.020, +0.015]$	$[-0.025, +0.021]$
Interaction	-0.0038	$[-0.030, +0.023]$	$[-0.039, +0.032]$
Chunk heads: exploratory family of 5			
Closure gain, zero	$+0.0350$	$[+0.004, +0.066]$	$[-0.004, +0.074]$
Closure gain, last-chunk	$+0.0206$	$[-0.000, +0.041]$	$[-0.005, +0.047]$
Zero − last-chunk, base	$+0.0137$	$[-0.013, +0.040]$	$[-0.020, +0.048]$
Zero − last-chunk, closure	$+0.0281$	$[-0.018, +0.074]$	$[-0.028, +0.084]$
Interaction	$+0.0144$	$[-0.012, +0.041]$	$[-0.018, +0.047]$

OLMo. One bank, last-chunk probe, frozen 20% payload and terminal conv tail are shared across all seven conditions on 325 requests. Closure8 executes .5675 of the context on average. Its +.0928 macro gain has nominal 95% interval $[-.0698, +.2554]$ and adjusted interval $[-.0982, +.2838]$. The near-equal closure and K/V-cross means are not evidence of equivalence: task effects differ in sign and magnitude. Appendix E.2 adds matched full-prefill and naive-reuse references to this cohort, preserving the original seven repair arms and diagnosing both repair-induced losses and recovery relative to unrepaired reuse.

	Absolute score			Paired Δ from Scatter		
	Full	Naive	Scatter	Closure	K/V cross	Endpoint
cwe	0.456	0.284	0.080	+0.288	+0.256	+0.132
fwe	0.893	0.693	0.587	+0.227	+0.187	+0.120
niah_multikey_1	1.000	0.720	1.000	+0.000	+0.000	-0.080
niah_multikey_2	1.000	0.520	0.120	+0.200	+0.200	+0.000
niah_multikey_3	0.960	0.400	0.000	+0.240	+0.120	+0.000
niah_multiquery	1.000	0.810	0.990	+0.010	+0.000	+0.010
niah_multivalue	0.980	0.420	0.940	-0.230	-0.210	-0.030
niah_single_1	1.000	1.000	1.000	+0.000	+0.000	+0.000
niah_single_2	1.000	1.000	1.000	+0.000	+0.000	+0.000
niah_single_3	1.000	1.000	0.120	+0.760	+0.720	+0.000
qa_1	0.760	0.360	0.640	+0.120	+0.120	-0.040
qa_2	0.600	0.280	0.520	+0.000	-0.040	-0.040
vt	1.000	0.008	0.944	-0.408	-0.152	-0.200
Macro	**0.8961**	**0.5766**	**0.6108**	**+0.0928**	**+0.0924**	**-0.0098**

0 1 −1 0 +1

Color shows magnitude only; taskwise differences are descriptive.

Figure 10: **OLMo history effects across all thirteen tasks.** Full/naive/scatter references place each task's starting quality beside its closure, K/V-cross and endpoint-cross gains over scatter. The same diverging scale applies to all three effect columns; colors are descriptive, not task-level significance tests. Macro weights tasks equally. Variable tracking and multivalue retrieval retain adverse effects; Table 23 adds their fixed-endpoint effects and interactions. All absolute scores remain in the numerical supplement.

LongBench. The uncapped study uses 420 requests, 20 per task, identical payloads and a common conv policy. Mean payload fraction is .2001; closure8 expands replay to .5250. The designated zero-matrix closure-minus-scatter gain is +.0233 with nominal 95% interval $[+.0024, +.0441]$ and the original three-replication-family interval $[-.0030, +.0495]$. The same contrast's exploratory five-comparison interval in Table 22 is different by design, not a numerical inconsistency. Neither family establishes adjusted superiority. Every task's primary and factorial effects are included in the supplement.

Query-independent chunk heads. The four-cell Qwen study freezes contiguous chunk-prefix payloads. Under zero initialization, closure raises mean replay from .1000 to .1132 and macro from .6946 to .7296. Its five-comparison interval $[-.0044, +.0745]$ contains zero. It supplies far less extra history than ProphetKV closure and is not a reproduction of full EPIC serving.

Pure-attention control. The 325-request Qwen3-8B run scores .78359 (full), .68031 (scatter), .67513 (span64) and .68277 (closure8). Closure-minus-scatter is +.00246 with 90% interval $[-.00758, +.01250]$. This older run lacks payload/source fingerprints; execution shapes can change floating-point results. It does not isolate architecture alone.

E.2 OLMo reference points and adverse task effects

Full-prefill and naive-reuse references supplement the same 325 OLMo requests and original seven repair arms. Tokens, checkpoint and independent chunk banks match the earlier control; naive reuse shares its zero-matrix/terminal-conv initialization. This is OLMo-Hybrid-7B base with plain prompts, whereas LinearKV evaluates OLMo-Hybrid-7B-Instruct (Liu et al., 2026b). We have no matched base/Instruct intervention and cannot attribute these failures to architecture or instruction tuning separately. One scatter request per task checks the original output. Dense prefill uses 1,024-token

persistent-cache streaming. These are quality anchors on previously evaluated requests, not new latency measurements or held-out replication.

Table 23: **Conditional channels in two adverse OLMo tasks.** Entries are mean score changes when replacing scatter by closure K/V or endpoint at the stated fixed source of the other output, on 25 requests per task. Interaction is the difference between the two conditional effects. These tasks were selected from earlier results; the entries are descriptive effects, not mediation shares. Figure 10 retains every task, and the complete absolute-score table remains in the supplement. Bold marks the largest absolute conditional channel effect per task, not significance.

	K/V effect		Endpoint effect		
Task	Scatter E	Closure E	Scatter K/V	Closure K/V	Interaction
Multivalue	$\mathbf{-0.210}$	-0.200	-0.030	-0.020	$+0.010$
Variable tracking	-0.152	-0.208	-0.200	$\mathbf{-0.256}$	-0.056

Macro full/naive/scatter/closure scores are .8961/.5766/.6108/.7036. The exploratory closure–naive contrast is +.1271, with nominal 95% interval $[-.0071, +.2613]$ and three-comparison adjusted interval $[-.0479, +.3020]$. Scatter–naive is +.0343 with adjusted interval $[-.3115, +.3800]$. These intervals do not establish repair superiority over naive reuse; the taskwise references below explain the heterogeneity behind the means.

The references distinguish damage to accurate reuse from damage to successful repair. Variable tracking and multivalue retrieval remain above naive reuse even when closure degrades scatter. Conversely, single-key retrieval 3 scores 1.000 for full and naive, .120 for scatter and .880 for closure. OLMo lacks naive-source output crossings, so that reference comparison alone cannot assign the latter damage to K/V. Table 23 shows greater K/V sensitivity in multivalue retrieval and sensitivity to both outputs in variable tracking; queries continue ordinary recurrence.

Post-hoc response descriptors place the failures in context. In variable tracking, outputs begin with the prompt's fixed worked-example answer in 0/25 full, 24/25 naive, 1/25 scatter and 11/25 closure responses; the K/V and endpoint crosses do so in 4/25 and 6/25. These example-copying responses score zero. Example copying is already prevalent under naive reuse and is less completely suppressed by closure than by scatter. These counts identify a recurring failure pattern without establishing why the repairs differ.

Multivalue closure contains exactly one distinct seven-digit number, repeated, in 5/25 responses, versus 0/25 for scatter; all five retain partial credit. These descriptions are not prospective mechanism tests. The supplement retains all nine arm scores, output hashes, descriptors and the complete exploratory three-contrast anchor family in `supplement/v12_olmo/`.

E.3 Natural-language evaluation with a fixed answer prefix

On the same 96 LongBench-v2 requests and five cache conditions, a companion protocol fixes the teacher-forced assistant prefix `The correct answer is (` before evaluation and selects the largest next-token logit among single-token A/B/C/D labels; ties use alphabetical order. The rendered query, payload, replay, bank and endpoint stay fixed. Tokenization checks preserve the original token boundary, and each condition reproduces its earlier first-token argmax. Label probabilities normalize only over those four logits.

Table 24: **Official generation and the fixed-prefix companion metric.** Equal weight across the same ten subdomains, 96 requests. Both columns use a fixed $[0, 1]$ gray scale; shading indicates score, not significance. Fixed-prefix choice changes the response protocol and removes the requirement to generate an extractable answer within 128 tokens; it is not a semantic rescore of the original responses. Bold marks the best repair score within each response protocol.

Condition	Official generation	Fixed-prefix choice
Full prefill	0.4337	0.5199
Naive reuse	0.0686	0.3724
Scatter	0.2917	0.3891
Closure	**0.3668**	**0.4480**
Closure K/V, Scatter endpoint	0.3486	0.4278

Table 25: Complete three-comparison family for fixed-prefix choice accuracy. The metric, contrasts and ten subdomain clusters were fixed before this new run. The cohort itself had already been evaluated; this is a supporting diagnostic, not a new-request replication. Bold, if present, marks effects and adjusted intervals that exclude zero.

Fixed-prefix contrast	Δ	Adjusted interval
Closure − scatter	$+0.0588$	$[-0.0116, +0.1293]$
Closure − naive	$+0.0756$	$[-0.1151, +0.2663]$
Closure K/V − scatter	$+0.0387$	$[-0.0368, +0.1141]$

All three adjusted fixed-prefix intervals contain zero despite positive point differences. Naive's absolute score changes from .0686 to .3724, showing response-protocol sensitivity without establishing semantic-recovery superiority. The fixed prefix prevents free-form deliberation before choice and can change label biases and recurrent computation; it does not assign a causal fraction of the original gain to formatting.

An audit of the original responses finds that every repair-arm extraction failure, and 78 of naive's 81, reaches the 128-token cap without an answer declaration; repair-arm extraction failure is thus associated with the output budget, without predicting longer-decode answers. Official scores remain unchanged; full examples and lexical rules are in `supplement/v12_natural/`.

F Allocation and Measured Request Cost

All 325 canonical requests complete all 38 conditions: nine cap/width settings with four policies each, plus full/naive references. The bank and last-chunk probe are shared within each request; every matched policy executes exactly that request's realized $|R|$. Canonical-query, read/write and source-immutability checks precede aggregation. Full task grids, actual geometry ranges and unused-cap counts remain machine-readable in the supplement.

In this original reference-engine grid, span64 has the highest point score in all nine settings, but all 36 cost-family and nine full-span-recompute-versus-target-only adjusted intervals include zero. Quality is not monotone in H; target-only's updated fraction r_U falls as H increases while realized r_R stays near the cap, and matched controls execute and write that same realized count. At cap .2 and $H = 8$, full-span recompute and target-only closure score .7466/.6677. Withholding already computed K/V serves the fixed-update diagnostic; it is not established as a deployment advantage. Exact values, geometry ranges and every nominal/adjusted interval remain in the supplement; the expanded sweep below repeats this grid at more widths on the optimized backend.

F.1 EXPANDED HISTORY-WIDTH SWEEP WITH OPTIMIZED EXECUTION

We extend the original control cohort to execution caps $\rho \in \{.1, .2, .4\}$ and widths $H \in \{1, 2, 4, 8, 16\}$, retaining all 325 requests and four policies at every setting. Zero matrices and the common terminal conv tail are fixed. $H = 1$ adds no predecessors, so target-only closure, full-span recompute and matched scatter are identical at that width. The original query-probe score vector is frozen; the historical bank is reconstructed once per request and checked against its tensor hashes. All $H = 4, 8, 16$ payloads and matched execution sets reproduce the earlier geometry exactly.

All 60 repair conditions and full/naive references are evaluated on a common FLA/paged-attention backend. The backend uses an output-preserving layerwise schedule that executes context repair immediately before each native query layer and omits the final context layer's unused attention/MLP outputs; on thirteen development requests this schedule preserves complete answers, query logits and full query caches exactly while reducing request time by about 20%. A preflight on thirteen rank-zero requests checks the optimized and separate repair/query schedules for exact logits and whole-query-cache agreement at every unique execution. Each worker loads one model, reuses each request's positioned bank through private cache copies, and shares results only when executed and written positions are identical. Unselected K/V and the shared bank are checked for mutation. Elapsed execution time includes diagnostic work and reuses frozen probe scores; it is not serving TTFT.

Table 26: **All fifteen settings of the expanded optimized-backend sweep.** Scores average the same thirteen task means. r_U is closure's payload fraction; each matched control executes and writes the corresponding realized r_R. Earlier reference-engine scores are retained separately. Bold marks column maxima within each budget; shading marks the best policy in each row. These are point estimates, not significant differences.

Cap (%)	H	r_U (%)	r_R (%)	Closure	Full-span	Scatter	Span64
10	1	9.99	9.99	0.5543	0.5543	0.5543	0.6489
10	2	6.75	9.99	0.5844	0.5696	**0.5555**	**0.6512**
10	4	4.15	9.98	0.5777	0.5824	0.5554	0.6448
10	8	2.32	9.96	0.5921	**0.6515**	0.5552	0.6498
10	16	1.26	9.91	**0.5938**	0.6102	0.5543	0.6502
20	1	19.99	19.99	0.6491	0.6491	0.6491	0.7888
20	2	14.21	19.99	0.7226	0.6665	0.6493	**0.7925**
20	4	9.35	19.98	**0.7322**	0.6833	0.6542	0.7919
20	8	5.61	19.96	0.6702	**0.7407**	0.6517	0.7897
20	16	3.12	19.92	0.6597	0.7368	**0.6590**	0.7907
40	1	40.00	40.00	0.7954	0.7954	0.7954	0.8898
40	2	30.26	39.99	0.8078	0.7977	0.7963	0.8895
40	4	21.57	39.98	**0.8104**	0.8130	0.7926	0.8898
40	8	14.26	39.96	0.8069	0.8277	0.7951	0.8871
40	16	8.68	39.93	0.7525	**0.8409**	**0.7975**	**0.8905**

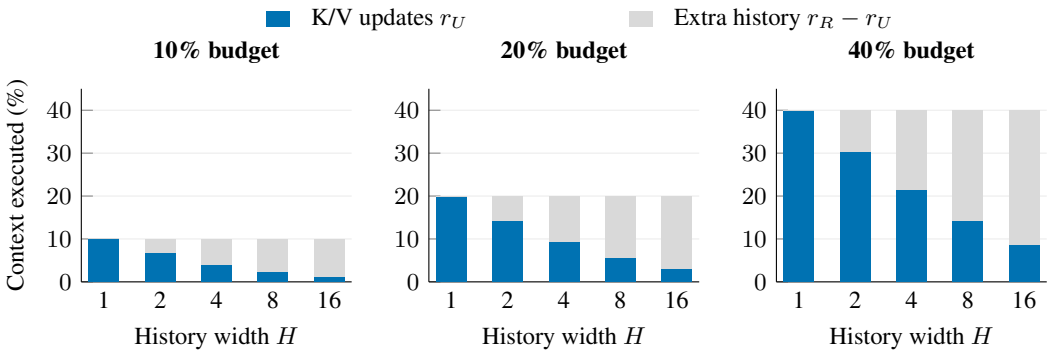


Figure 11: **Allocation at every tested budget and history width.** Closure's actual executed fraction is split into updated positions (blue) and extra predecessor execution (gray), on a shared context-percentage scale. Each bar averages the same 325 requests. At $H = 1$ there is no extra history. The ranked payload prefix is selected separately for each cap/width, so its locations as well as its size can change. Exact means are in Table 26.

The expanded run has a separately declared 60-member cost family (target-only and full-span recompute versus each matched control), a 15-member full-span recompute minus target-only family, and twelve exploratory adjacent-width closure contrasts. The original 36/nine-member families are unchanged. All intervals use the same enumerated task-cluster procedure; neither repeated conditions nor worker shards add task clusters. Full prefill and naive reuse score 0.9265/0.6313 in this backend. Of the 60 repair-minus-matched-control contrasts, 0/0 adjusted intervals lie entirely above/below zero; the other 60 contain zero. Of the 15 full-span recompute minus target-only contrasts, 0/0 lie entirely above/below zero. The optimized path changes 299 of the 11,700 overlapping per-request repair scores relative to the historical reference-engine grid, despite identical banks and geometry. These are descriptive backend differences, not additional significance tests.

Table 27: **Adjacent history-width changes at each fixed cap.** These twelve closure contrasts retain one exploratory family. Increasing H also changes payload coverage and selected content, so they are allocation contrasts, not fixed-U history effects. Budget blocks are separated; no adjusted interval excludes zero.

Cap (%)	History comparison	Δ (pp)	Familywise 95% CI (pp)
10	$H = 2$ minus 1	$+3.02$	$[-5.76, +11.79]$
10	$H = 4$ minus 2	-0.67	$[-15.34, +14.00]$
10	$H = 8$ minus 4	$+1.44$	$[-14.70, +17.57]$
10	$H = 16$ minus 8	$+0.17$	$[-14.65, +14.99]$
20	$H = 2$ minus 1	$+7.35$	$[-3.99, +18.70]$
20	$H = 4$ minus 2	$+0.95$	$[-5.34, +7.25]$
20	$H = 8$ minus 4	-6.19	$[-16.51, +4.12]$
20	$H = 16$ minus 8	-1.05	$[-8.55, +6.46]$
40	$H = 2$ minus 1	$+1.24$	$[-3.16, +5.64]$
40	$H = 4$ minus 2	$+0.26$	$[-3.63, +4.14]$
40	$H = 8$ minus 4	-0.34	$[-5.45, +4.77]$
40	$H = 16$ minus 8	-5.44	$[-13.54, +2.66]$

All 20,150 logical arm scores, the complete task-level grid, and all 87 nominal/familywise contrasts are retained with the source and execution receipts. Duplicate geometries are explicit aliases, not extra independent observations. The point-estimate frontier retained in the numerical export consists of naive reuse, matched span64 and full-span recompute, with no target-only closure point. This describes observed means, without establishing statistical dominance or selecting a deployable policy. This is a remeasurement on existing requests, not a new-request replication.

F.2 ALIGNED PAPER-DERIVED BASELINE COMPARISON

The aligned comparison evaluates LinearKV-derived selected-token repair and HYPIC seam repair under common inputs and a common request clock. Its final costs are reported in Appendix F.3, which remeasures these baselines together with the closure-family policies; earlier runtime campaigns remain in the numerical supplement.

Operating paths and cross-paper scope. Six selected-token repair points use caps .2/.4 at the budget study's realized request counts; native full prefill and zero/last-chunk naive reuse supply three references. HYPIC reuses the first chunk, completely replays later chunks shorter than its seam width, and performs three GDN seam-transition passes plus cached matrix composition inside the clock. Widths and comparisons were fixed before outcomes.

Figure 5 uses the common executed fraction r_R, keeping method-specific width parameters out of its labels. Span uses the fixed 64-token selector window (span64); its cap determines the final selected count. Scatter selects ranked positions without a predecessor-window parameter. HYPIC's w counts leading seam tokens in each subsequent chunk, not a global percentage: $w = 8, 102, 205$ yield mean $r_R = 1.51\%, 19.04\%, 38.15\%$ here. The first chunk is reused without seam repair, and short later chunks can be replayed in full. Additional recurrent seam/transition passes and matrix composition contribute to latency but not to the count of distinct replayed positions. Thus the near-20% and near-40% groups are not exact token-count or FLOP matches.

This latency campaign has no 10% selected-token operating point. The expanded 10% quality sweep uses a different request set/backend and reuses frozen probe scores; its instrumented durations are not interchangeable with this end-to-end request clock.

Liu et al. (2026b)'s up-to-92% recovery uses OLMo-Hybrid-7B-Instruct, approximately 20% recomputation and an eight-category RULER aggregate; their Qwen checkpoint is Qwen3.6-27B. Our mechanism control uses Qwen3.5-9B, 10% fixed updates and thirteen-task macro; supporting OLMo uses the base checkpoint. These setting differences prevent a direct cross-paper score comparison. Initializer sufficiency and selected-only execution sufficiency are distinct questions. This study evaluates paper-derived constructions, excluding the authors' SGLang/NIXL serving runtimes.

Storage and numerical alignment. Each method builds its own offline bank from identical chunk tokens. PIC retains BF16 K/V, matrices and conv artifacts. Primary HYPIC retains BF16 transitions/states (T/S) and K/V/conv; T/S expand to FP32 after GPU transfer. Both transfer and conversion are timed. This PyTorch adapter does not reproduce HYPIC's engine-integrated routing or overlapped NIXL transfers (Liu et al., 2026a). These request times do not isolate intrinsic seam-composition cost; a transfer/conversion/kernel breakdown remains unmeasured. A fourth HYPIC point retains the recovered prototype's FP32 T/S at seam width 8. The adapter shares native fresh-segment attention, convolution, scoped FLA recurrence and an HF greedy decoder. Dense native/prototype last logits must match byte-for-byte on every request, and banks must remain unchanged; this does not establish equality of approximate repairs.

Table 28: **HYPIC bank and resource scope.** Copied bytes average the CPU-bank tensors requested for GPU materialization; GPU arithmetic can expand T/S further. Peak is the maximum allocated memory, not a mean or total device usage. Offline seconds average the recorded bank-construction path and are excluded from first-token timing; they are not an amortized serving cost. Decimal GB are used. Shading isolates the matched seam-width precision pair; bold marks its lower copied-byte count.

HYPIC bank	r_R	Copied GB	Peak GB	Offline s
HYPIC $w = 8$, BF16 bank	0.015	**1.121**	21.778	4.598
HYPIC $w = 102$, BF16 bank	0.190	1.073	25.137	4.482
HYPIC $w = 205$, BF16 bank	0.382	1.022	28.819	4.430
HYPIC $w = 8$, FP32 bank	0.015	1.956	21.778	4.150

Clock, identity checks and families. The 312 requests are control-study ranks 1–24 per task; their earlier quality outcomes were known. Thirteen rank-zero requests validate the paths, and one

evaluation request diagnoses final-normalization shape: all layer outputs agree, but final-token-only normalization changes three BF16 values. Full-length normalization restores exact logits. The final complete runs use this aligned shape. A bounded host allocator and between-request heap trimming operate outside the timers while live banks remain CPU-resident.

The clock begins with the CPU bank and ends at a host-ready first token. It includes materialization, selection, initialization/composition, replay, query and HYPIC conversion; offline bank construction, warmup, tokenization, queueing, network and subsequent decode are excluded. One decoded warmup precedes five randomized paired repeats. Repeat zero decodes the full answer; later repeats verify query-end K/V, matrix/conv, logits and cache metadata, triggering full decoding on mismatch. Checks and hashing are outside timing. Workers share an H100 node; these are single-request costs, not concurrent-serving throughput. An earlier campaign's lower-right causal-bias constructor inherited unused NaN-filled FP32 storage; the metadata-only constructor used in every final timing reproduces all outputs exactly while lowering span latency (records in `supplement/v13_runtime/` and `supplement/causal_bias/`).

The frozen families comprise five baseline contrasts separately for quality and latency, two initializer contrasts separately for each metric, ten repair-versus-native latency contrasts, and singleton HYPIC BF16-minus-FP32 quality and latency contrasts: 26 in total; Appendix F.3 retains them.

F.3 Every replay policy on one request clock

The main-text cost comparison (Figure 5) comes from a single campaign that remeasures the thirteen aligned-baseline operating points of Appendix F.2 together with the three closure-family policies of Figure 2b. Target-only closure, full-span recompute and warmup-KV retain each run under zero and last-chunk initialization at the matched .2/.4 caps with $H = 8$; zero-initialized matched scatter completes the policy/initializer grid. All 27 arms share the same 312 requests (ranks 1–24 in each RULER task), the same CPU-resident bank, the same bounded host allocator and the same CPU-bank-to-host-ready-first-token clock; one decoded warmup and five randomized paired repeats per arm give 42,120 timed measurements. No timing row is imported from an earlier campaign.

Policies on the serving path. All policies share positional work, avoid unused cache/probe copies and use FLA recurrent kernels (Yang & Zhang, 2024) (fla-core 0.5.2, Triton 3.7.1); native full prefill uses the same recurrence. Target-only closure and full-span recompute use the serving path's existing budgeted ranked-prefix selection under global-mask SDPA attention; span64 uses Flash attention (Dao, 2024) on contiguous runs. Warmup-KV retain runs the target-only replay, buffers the post-rotary K/V that every replayed position already projects, and writes those entries after replay; its recurrent endpoint equals target-only closure by construction. The capture audit that compares buffered entries with the engine's installed entries runs on the untimed warmup execution only. The timed request path includes its own probe, selection and closure construction. After the clock stops, a separate audit recomputes the expected geometry from independently obtained native probe scores and checks it against the executed indices; this audit supplies no positions to the timed closure path. Matched scatter/span64 receive the reference execution count as an experimental input and time their own selection.

Bank rebuild and identity gates. The original bank tensors were removed from disk after the earlier campaign and were rebuilt with the recorded producer contract. Each rebuilt bank is accepted only with the recorded chunk order and the quality study's serialized tensor hashes; the rebuilt manifest digests differ in serialization bytes and are recorded alongside the original digests. The thirteen carried-over arms must reproduce their earlier predictions, scores, positions, query logits and post-query cache signatures exactly on every request, and native/prototype dense full-prefill logits must agree; closure-family quality is decoded fresh in this campaign and is a quality–cost point, not new held-out task-quality validation.

Policy / initializer	r_R (%)	Score ↑	First-token time (ms) ↓
Full prefill	100.0	0.9247	292.7
Near 20% execution			
Closure / zero	20.0	0.6648	266.2
Closure / last-chunk	20.0	0.6960	265.1
Full-span / zero	20.0	0.7357	266.5
Full-span / last-chunk	20.0	0.7381	265.2
Warmup-KV / zero	20.0	0.6981	272.0
Warmup-KV / last-chunk	20.0	0.7128	271.0
Span64 / zero	20.0	0.7935	229.3
Span64 / last-chunk	20.0	0.7900	228.1
Scatter / zero	20.0	0.6530	263.2
Scatter / last-chunk	20.0	0.6593	261.8
HYPIC / BF16 bank	19.0	0.7534	526.0
Near 40% execution			
Closure / zero	40.0	0.8082	375.3
Closure / last-chunk	40.0	0.7937	374.3
Full-span / zero	40.0	0.8282	375.8
Full-span / last-chunk	40.0	0.8045	374.9
Warmup-KV / zero	40.0	0.8365	383.3
Warmup-KV / last-chunk	40.0	0.7965	382.2
Span64 / zero	40.0	0.8869	291.7
Span64 / last-chunk	40.0	0.8769	290.8
Scatter / zero	40.0	0.7944	369.1
Scatter / last-chunk	40.0	0.7333	367.9
HYPIC / BF16 bank	38.2	0.8431	638.5
Zero / low execution			
Naive / zero	0.0	0.6316	94.9
Naive / last-chunk	0.0	0.5800	97.1
HYPIC / BF16 bank	1.5	0.6908	428.7
HYPIC / FP32 bank	1.5	0.6928	591.3

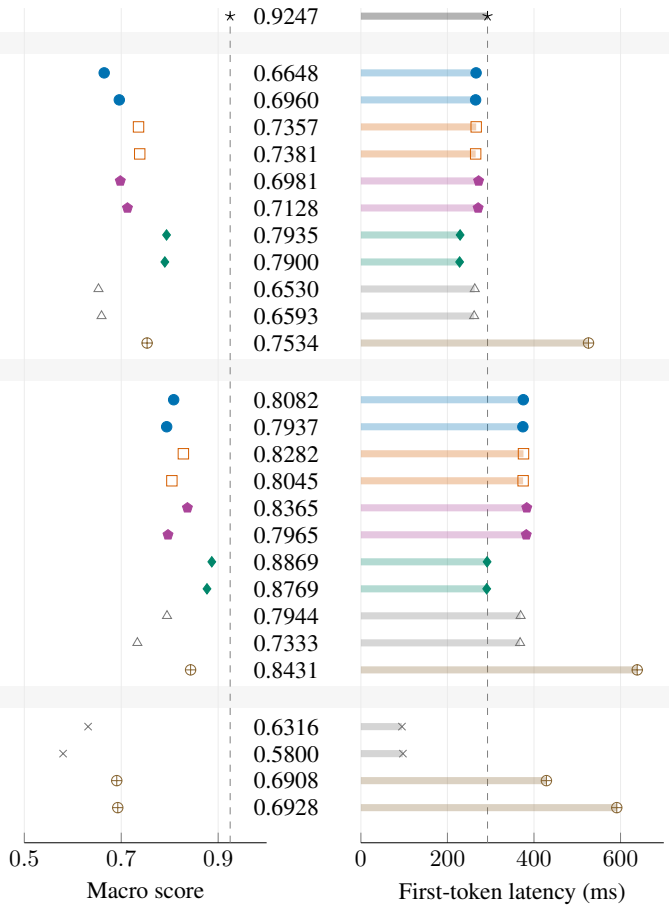


Figure 12: **All 27 operating points on the same 312 requests.** Rows name the replay policy and initializer; r_R gives the actual mean context execution. Times average five-repeat request medians. Dashed lines mark full prefill. The main-text figure shows the zero-initialized rows, the LinearKV adapter and HYPIC.

Table 29: **Paired latency contrasts.** Differences are seconds of per-request median latency on the same requests in the same campaign; task-macro means with nominal and familywise-adjusted exact wild cluster bootstrap-t intervals over thirteen task clusters. Cap identifies the matched executed-token budget; family sizes m set the adjustment across both caps.

Family	Contrast	Cap (%)	Δ (s)	95% CI	Adjusted CI
Closure family vs. full prefill ($m = 12$)	Closure / last-chunk − Full prefill	20	−0.0275	[−0.0360, −0.0190]	[−0.0405, −0.0146]
	Closure / last-chunk − Full prefill	40	+0.0817	[+0.0751, +0.0882]	[+0.0716, +0.0917]
	Closure / zero − Full prefill	20	−0.0264	[−0.0346, −0.0182]	[−0.0389, −0.0140]
	Closure / zero − Full prefill	40	+0.0826	[+0.0760, +0.0893]	[+0.0727, +0.0926]
	Full-span / last-chunk − Full prefill	20	−0.0274	[−0.0358, −0.0191]	[−0.0403, −0.0145]
	Full-span / last-chunk − Full prefill	40	+0.0822	[+0.0756, +0.0888]	[+0.0722, +0.0922]
	Full-span / zero − Full prefill	20	−0.0262	[−0.0344, −0.0180]	[−0.0388, −0.0136]
	Full-span / zero − Full prefill	40	+0.0831	[+0.0765, +0.0897]	[+0.0733, +0.0929]
	Warmup-KV / last-chunk − Full prefill	20	−0.0217	[−0.0300, −0.0134]	[−0.0344, −0.0090]
	Warmup-KV / last-chunk − Full prefill	40	+0.0895	[+0.0826, +0.0964]	[+0.0788, +0.1002]
	Warmup-KV / zero − Full prefill	20	−0.0206	[−0.0290, −0.0122]	[−0.0336, −0.0076]
	Warmup-KV / zero − Full prefill	40	+0.0906	[+0.0841, +0.0972]	[+0.0807, +0.1006]
Closure vs. scatter ($m = 4$)	Closure / last-chunk − Scatter / last-chunk	20	+0.0034	[+0.0030, +0.0037]	[+0.0029, +0.0039]
	Closure / last-chunk − Scatter / last-chunk	40	+0.0064	[+0.0054, +0.0074]	[+0.0051, +0.0077]
	Closure / zero − Scatter / zero	20	+0.0031	[+0.0023, +0.0039]	[+0.0020, +0.0041]
	Closure / zero − Scatter / zero	40	+0.0062	[+0.0053, +0.0070]	[+0.0050, +0.0073]
Closure family vs. span64 ($m = 12$)	Closure / last-chunk − Span64 / last-chunk	20	+0.0370	[+0.0284, +0.0456]	[+0.0242, +0.0498]
	Closure / last-chunk − Span64 / last-chunk	40	+0.0836	[+0.0673, +0.0998]	[+0.0586, +0.1085]
	Closure / zero − Span64 / zero	20	+0.0369	[+0.0281, +0.0458]	[+0.0232, +0.0507]
	Closure / zero − Span64 / zero	40	+0.0836	[+0.0676, +0.0997]	[+0.0590, +0.1083]
	Full-span / last-chunk − Span64 / last-chunk	20	+0.0371	[+0.0284, +0.0458]	[+0.0240, +0.0502]
	Full-span / last-chunk − Span64 / last-chunk	40	+0.0841	[+0.0679, +0.1004]	[+0.0591, +0.1091]
	Full-span / zero − Span64 / zero	20	+0.0372	[+0.0282, +0.0461]	[+0.0236, +0.0507]
	Full-span / zero − Span64 / zero	40	+0.0841	[+0.0680, +0.1002]	[+0.0596, +0.1086]
	Warmup-KV / last-chunk − Span64 / last-chunk	20	+0.0428	[+0.0340, +0.0516]	[+0.0296, +0.0560]
	Warmup-KV / last-chunk − Span64 / last-chunk	40	+0.0914	[+0.0748, +0.1080]	[+0.0661, +0.1167]
	Warmup-KV / zero − Span64 / zero	20	+0.0427	[+0.0340, +0.0514]	[+0.0296, +0.0559]
	Warmup-KV / zero − Span64 / zero	40	+0.0916	[+0.0756, +0.1077]	[+0.0670, +0.1162]
Scatter initializer ($m = 2$)	Scatter / zero − Scatter / last-chunk	20	+0.0014	[+0.0008, +0.0020]	[+0.0007, +0.0021]
	Scatter / zero − Scatter / last-chunk	40	+0.0012	[+0.0004, +0.0020]	[+0.0003, +0.0021]
Variant vs. closure ($m = 8$)	Full-span / last-chunk − Closure / last-chunk	20	+0.0001	[−0.0002, +0.0004]	[−0.0004, +0.0006]
	Full-span / last-chunk − Closure / last-chunk	40	+0.0006	[−0.0002, +0.0013]	[−0.0005, +0.0016]
	Full-span / zero − Closure / zero	20	+0.0002	[−0.0005, +0.0009]	[−0.0008, +0.0013]
	Full-span / zero − Closure / zero	40	+0.0005	[+0.0001, +0.0008]	[−0.0001, +0.0010]
	Warmup-KV / last-chunk − Closure / last-chunk	20	+0.0058	[+0.0053, +0.0064]	[+0.0050, +0.0067]
	Warmup-KV / last-chunk − Closure / last-chunk	40	+0.0078	[+0.0071, +0.0086]	[+0.0066, +0.0091]
	Warmup-KV / zero − Closure / zero	20	+0.0058	[+0.0052, +0.0065]	[+0.0049, +0.0067]
	Warmup-KV / zero − Closure / zero	40	+0.0080	[+0.0076, +0.0084]	[+0.0074, +0.0086]

Table 30: **Paired quality contrasts.** Cap identifies the matched executed-token budget. Differences of macro score on the same requests, with the same interval procedure as Table 29.

Family	Contrast	Cap (%)	Δ	95% CI	Adjusted CI
Closure vs. scatter ($m = 4$)	Closure / last-chunk − Scatter / last-chunk	20	+0.0366	[−0.1285, +0.2018]	[−0.1544, +0.2277]
	Closure / last-chunk − Scatter / last-chunk	40	+0.0605	[+0.0045, +0.1164]	[−0.0112, +0.1322]
	Closure / zero − Scatter / zero	20	+0.0118	[−0.1537, +0.1772]	[−0.1844, +0.2079]
	Closure / zero − Scatter / zero	40	+0.0137	[−0.0584, +0.0858]	[−0.0686, +0.0960]
Closure family vs. span64 ($m = 12$)	Closure / last-chunk − Span64 / last-chunk	20	−0.0940	[−0.2720, +0.0841]	[−0.3446, +0.1567]
	Closure / last-chunk − Span64 / last-chunk	40	−0.0832	[−0.1969, +0.0305]	[−0.2292, +0.0628]
	Closure / zero − Span64 / zero	20	−0.1287	[−0.3100, +0.0526]	[−0.3973, +0.1399]
	Closure / zero − Span64 / zero	40	−0.0787	[−0.1831, +0.0257]	[−0.2136, +0.0563]
	Full-span / last-chunk − Span64 / last-chunk	20	−0.0518	[−0.2377, +0.1341]	[−0.2735, +0.1699]
	Full-span / last-chunk − Span64 / last-chunk	40	−0.0724	[−0.1637, +0.0189]	[−0.2031, +0.0583]
	Full-span / zero − Span64 / zero	20	−0.0578	[−0.2378, +0.1222]	[−0.2756, +0.1600]
	Full-span / zero − Span64 / zero	40	−0.0587	[−0.1615, +0.0442]	[−0.1969, +0.0796]
	Warmup-KV / last-chunk − Span64 / last-chunk	20	−0.0771	[−0.2633, +0.1091]	[−0.3152, +0.1609]
	Warmup-KV / last-chunk − Span64 / last-chunk	40	−0.0804	[−0.1964, +0.0355]	[−0.2495, +0.0886]
	Warmup-KV / zero − Span64 / zero	20	−0.0954	[−0.2567, +0.0659]	[−0.3194, +0.1286]
	Warmup-KV / zero − Span64 / zero	40	−0.0503	[−0.1650, +0.0644]	[−0.2009, +0.1003]
Scatter initializer ($m = 2$)	Scatter / zero − Scatter / last-chunk	20	−0.0063	[−0.0334, +0.0208]	[−0.0350, +0.0224]
	Scatter / zero − Scatter / last-chunk	40	+0.0612	[−0.0136, +0.1359]	[−0.0198, +0.1421]
Variant vs. closure ($m = 8$)	Full-span / last-chunk − Closure / last-chunk	20	+0.0421	[−0.1106, +0.1949]	[−0.1629, +0.2472]
	Full-span / last-chunk − Closure / last-chunk	40	+0.0107	[−0.0778, +0.0993]	[−0.0873, +0.1088]
	Full-span / zero − Closure / zero	20	+0.0709	[−0.0982, +0.2399]	[−0.1642, +0.3060]
	Full-span / zero − Closure / zero	40	+0.0200	[−0.0520, +0.0921]	[−0.0634, +0.1034]
	Warmup-KV / last-chunk − Closure / last-chunk	20	+0.0168	[−0.1346, +0.1683]	[−0.1810, +0.2146]
	Warmup-KV / last-chunk − Closure / last-chunk	40	+0.0027	[−0.0690, +0.0745]	[−0.0719, +0.0774]
	Warmup-KV / zero − Closure / zero	20	+0.0333	[−0.1338, +0.2003]	[−0.1971, +0.2637]
	Warmup-KV / zero − Closure / zero	40	+0.0284	[−0.0468, +0.1035]	[−0.0559, +0.1126]

Results. All 312 requests pass every gate: the thirteen carried-over arms reproduce their earlier identities exactly (4,056 request outputs), and native/prototype logits agree on every request. Their quality is therefore unchanged, and their latencies move by under .03 s except FP32 HYPIC (+.066 s). At cap .2 the closure family runs .265–.272 s: faster than full prefill (.293 s; adjusted intervals within

$[-.041, -.008]$ s), slower than span64 (.229 s) by $+.037$ to $+.043$ s and slower than matched scatter by $+.003$ s. At cap .4 it runs .374–.383 s, slower than full prefill by $+.08$ to $+.09$ s and than span64 (.292 s) by $+.084$ to $+.092$ s. Full-span recompute executes the same set as target-only closure and has an observed mean latency difference of at most 0.6 ms (all four adjusted intervals contain zero; equivalence is not established), whereas warmup-KV retain's buffering and final write add $+.006$–.008 s. Zero and last-chunk initialization differ by under .002 s within every closure-family policy (descriptive). Every closure-family quality contrast against span64, against scatter, and between variants has an adjusted interval containing zero; the point estimates against span64 range from $-.05$ to $-.13$. Zero-initialized closure at cap .2 reproduces the backend-selection candidate's macro score (.6648) exactly and runs .266 s here against .272 s under the earlier allocator. The carried-over baseline, initializer, native-reference and HYPIC-precision families reproduce the earlier campaign's conclusions: both span20 initializers and LinearKV (scatter20) are faster than full prefill, span40 is unresolved, and all baseline-quality intervals contain zero.

G Numerical Validation and Remaining Scope

Hash identity, installed-state receipts and independent numerical agreement test different properties. A historical dense integration check (13 RULER prompts, $U = R = [T]$, zero initialization) gave 13/13 next-token top-1 agreement with worst BF16/FP32 relative K/V errors of .345/.0090; dense agreement does not validate gaps or the main terminal-conv policy.

Independent sparse-policy audit. A separate token-by-token reference checks 13 requests, one per task: 15 sparse geometry/write-policy combinations plus six scan-boundary combinations in FP32/BF16, with one-token and full queries. Cases cover contiguous runs, large gaps, chunk crossings, overlapping windows and single-token replay. Bank assembly is checked exactly; layer outputs, K/V, matrices, conv registers and logits are compared. Native one-shot/chunked and split/joint processing provide numerical baselines.

The reference independently implements ordering, masks, conv advancement, recurrent updates and cache installation; it shares weights and primitive projections/norms. Sequence-parallel delta-rule evaluation (Yang et al., 2024) and floating-point effects (Goldberg, 1991; Higham & Mary, 2022) motivate separating semantic checks from arithmetic agreement, not assuming any observed discrepancy is harmless rounding.

Table 31: Independent audit maxima across the base and scan-boundary cases in each precision. Errors are relative L2 distances; the compared implementations perform the same sparse policy. These are numerical diagnostics, not benchmark-quality gains or statistical equivalence tests. Shading identifies the BF16 matrix and conv errors discussed in the text.

Quantity	FP32 max	BF16 max
Layer output	2.79×10^{-5}	0.305
K/V	3.62×10^{-6}	0.0473
Matrix	0.000614	0.224
Conv	4.5×10^{-5}	0.308
Same-start query logits	1.63×10^{-5}	0.0615
End-to-end query logits	8.97×10^{-5}	0.0694
Native chunked / one-shot query	1.84×10^{-5}	0.0494
Native split / joint query	4.56×10^{-5}	0.0309

The FP32 path still uses the implementation's BF16 installed matrix storage. Same-start query comparison isolates query arithmetic; end-to-end comparison additionally includes replay differences. BF16 scan cases include 5 last-token top-1 disagreements across 156 query comparisons; the 390 base comparisons have 0. These disagreements and the large BF16 layer errors are retained. Separate same-input local base-case checks have maximum mixed-block relative error 0.003383, and the long-scan local checks have 0.002242; they do not establish full-stack bitwise equality.

The audit source commits, complete request IDs, raw-tree checksums and numerical reports are in the supplement. Engineering tolerances were used for review, not as preregistered statistical thresholds.

The audit supports the tested policies/cases, not every input, every implementation or an explanation that all differences are rounding.

Of the five BF16 scan-query top-1 disagreements, four concern one-token query prefixes and one a full 99-token query. They are not five task-quality failures. The base and long-scan same-input local checks cover ten and four cases per request, respectively; full source hashes and per-case errors remain in the supplement.

G.1 Replay-only arithmetic precision

The original ranks 0–4 per task fix 65 requests before outcomes. BF16 scatter/closure reproduce banks, endpoints and predictions. Only replay parameters/operations become FP32; positioned K/V are cast without re-rotation, and unselected K/V remain exact after cast-back. Original parameters are restored; buffers are never promoted in place. Bank/probe/query/decode and installed matrix storage remain BF16. Full-prefill predictions are rechecked after restoration.

Table 32: Complete precision diagnostic and its three-comparison family. Original BF16 query/decode is used in every cell. All per-task scores and nominal intervals accompany the supplement. Bold marks effects and adjusted intervals that exclude zero.

Replay arithmetic	Scatter	Closure	Δ	Adjusted CI (3)
BF16	0.5418	0.7608	$\mathbf{+0.2190}$	$\mathbf{[+0.0479, +0.3901]}$
FP32 replay	0.5418	0.7577	$\mathbf{+0.2159}$	$\mathbf{[+0.0507, +0.3810]}$
Precision interaction	—	—	-0.0031	$[-0.0084, +0.0022]$

The quality gain remains positive in both precisions; their interaction is -.00308 with adjusted CI $[-.00836, +.00221]$. Scatter changes nine predictions but no scores; closure changes twelve predictions and two scores. This is bounded precision robustness, not bitwise equality, an equivalence test or another independent cohort. Large internal relative errors need not change a discrete answer score, but the numerical-audit maxima and this quality comparison use different cases. They do not establish directional robustness of state space. Nor do Frobenius state distances supply a monotone predictor of quality: the native-reference distances in Appendix D.3 and the conditional vector alignments in Appendix D.8 measure different objects.