# Tail-Replay

## Escaping the Curse of Linear Attention in Prefix Caching for Hybrid LLMs

Paper by Yirui Liu, Ruoling Qi, Xuaner Wu,
Penghang Liu, and Jian Chen

arXiv:2608.30310v2 | Revised October 15, 2026
Reading presentation | October 16, 2026

Exact FA caches + short tail replay $\longrightarrow$ flexible prefix reuse

https://arxiv.org/abs/2608.30310v2. Analysis slides; not an author presentation.

# A token match may have no matching recurrent state

## Full attention (FA)

Each token has its own KV entry.

Matching $t_{1:m}$ lets us retrieve $K_{1:m}, V_{1:m}$.

## Linear attention (LA)

Each update overwrites a recurrent state.

The final $S_n$ cannot simply be sliced into an earlier $S_m$.

Checkpoints restore selected boundaries. A hit between them needs extra recomputation.

Tail-Replay reconstructs an approximate state at the token-match boundary.

Paper, Sec. 1, PDF pp. 1–2.

# The design changes what is cached and reconstructed

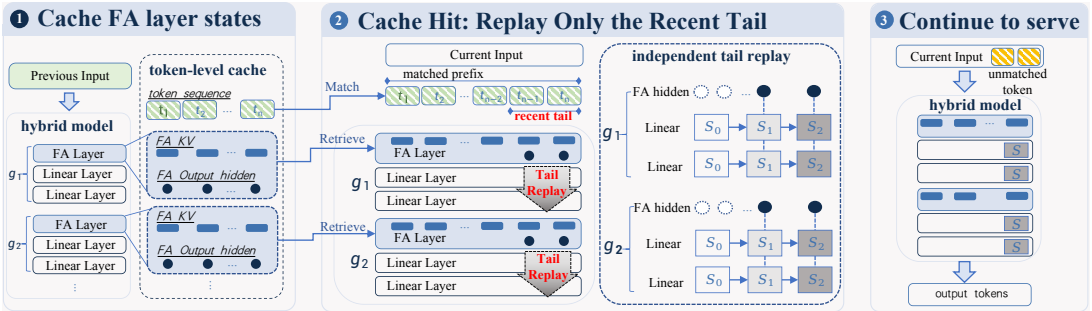
| Approach | Recovery material | Main design choice |
|---|---|---|
| FA prefix caches | Token-indexed KV | Slice the shared prefix |
| Marconi / Sparse Prefix Caching | Recurrent checkpoints | Which states to keep / where to place them |
| LinearKV | Cached local linear states | Initialize from saved local states |
| Tail-Replay | FA KV + FA output hiddens | Rebuild LA states from a recent tail |

Reported speedups compare with **full prefill**.

Checkpoint-system latency is not directly compared.

Paper, Secs. 1–2, PDF p. 2; comparison scope: Table 2, PDF p. 4.

# Exact FA outputs let each LA group replay independently




1. Cache token-level FA KV and FA output hiddens.
2. Replay a recent hidden suffix inside each LA group.
3. Continue with exact FA KV and approximate LA states.

Original Fig. 1, Sec. 3, PDF p. 3. A group is one FA layer plus following consecutive LA layers.

# A cache hit replays only the last $k = \lceil rm \rceil$ positions

For each FA layer, retain

$$\mathcal{C} = \{(K_i^{\mathrm{FA}}, V_i^{\mathrm{FA}}, h_i)\}_{i=1}^{n}.$$

1. Match $m$ prefix tokens and retrieve their FA KV.
2. Fetch $h_{m-k+1:m}$ for each group; zero its LA states.
3. Replay the tail through that group's LA layers.
4. Synchronize, then process the unmatched suffix.

## Illustrative example

$m = 10{,}000$, $r = 5\% \Rightarrow$ replay positions 9,501–10,000.
The ratio refers to the *matched prefix*, not the new suffix.

Paper, Sec. 3, Eq. (2), PDF p. 3. Example is explanatory, not an experiment.

# Gated updates can attenuate the omitted history

The paper writes the Gated DeltaNet recurrence as

$$S_i = T_i S_{i-1} + \beta_i v_i k_i^\top, \qquad T_i = \alpha_i (I - \beta_i k_i k_i^\top).$$

- $S_i$: recurrent state; $k_i, v_i$: key and value vectors.
- $\alpha_i \in (0, 1]$: forgetting gate; $\beta_i$: update strength.
- Old information is transformed repeatedly; recent inputs are written into the state.

**Replay**

Set $\hat{S}_{m-k} = 0$ and apply the recurrence to the last $k$ positions.

Paper, Eqs. (1)–(2), PDF pp. 2–3. Operator notation follows the paper; tensor orientation is discussed in backup.

# A short tail works when old-state influence contracts

**Conditional derivation:** assume identical transitions and writes during replay. Let $a = m - k$.

$$E_m \equiv S_m - \hat{S}_m = (T_m \cdots T_{a+1}) S_a.$$

If $\|T_i\|_2 \le \rho < 1$, then

$$\|E_m\|_F \le \rho^k \|S_a\|_F.$$

- Longer replay can suppress more initial-state error.
- The paper does not establish a uniform contraction rate.
- Later layers in a group can also receive perturbed inputs.

Reader derivation from Eq. (1), PDF p. 2; not a theorem claimed by the paper.

# Every FA boundary provides a fresh exact replay input

## Within a group

Approximate LA states can alter outputs and the inputs of later LA layers.

## At the next group

Start from that FA layer's cached output hidden, taken from full prefill.

- This confines propagation of replay-input error to each group.
- It does not make all reconstructed states exact.
- Exact KV for the matched prefix is retained throughout.

Paper, Sec. 3, "Independent Tail-Replay," PDF p. 3.

# Only the final FFN in each replayed group is unnecessary

$$\underbrace{\mathrm{LA}_1 \to \mathrm{FFN}_1 \to \cdots \to \mathrm{LA}_L}_{\text{needed to reconstruct this group's states}} \qquad \underbrace{\mathrm{FFN}_L}_{\text{skip during replay}}$$

- The last LA block already updates the final recurrent state.
- Its FFN output would feed the next group, whose input is cached.
- Intermediate FFNs remain necessary for later LA layers.
- Normal processing of the unmatched suffix is unchanged.

Paper, Sec. 3, "Replay efficiency," PDF p. 3. Diagram is an explanatory abstraction.

# FA KV transfer can overlap with LA reconstruction

Replay uses tail hiddens; it does not consume cached FA KV.

$$T_{\text{serial}} \approx T_{\text{KV}} + T_{\text{replay}} + T_{\text{other}},$$
$$T_{\text{overlap}} \approx \max(T_{\text{KV}}, T_{\text{replay}}) + T_{\text{other}}.$$

- Copy FA KV on a separate stream; synchronize before suffix forward.
- Tail hiddens must already be available to begin replay.
- Increasing $r$ costs more once replay lies on the critical path.

Paper mechanism: Sec. 3, PDF p. 3. Timing equations are an idealized reader model, ignoring contention.

# Quality and latency use different evaluation protocols

- **Models:** OLMo-Hybrid-7B, Qwen3.5-4B, Qwen3.6-27B.
- **Platform:** NVIDIA H100; PyTorch 2.9.1.
- **Quality:** full prefill, zero-only, 5% replay, 10% replay.
- **LongBench:** five tasks; selected 16K–64K cells; F1 / ROUGE-L.
- **RULER:** four tasks $\times$ two lengths (8K / 16K); recall.
- **TTFT:** shared narrativeqa prefixes at nominal 8K / 16K / 32K; mean of 20 timed repetitions.

Zero-only keeps matched FA KV but resets LA states without replay.

Paper, Sec. 4, Tables 1–2, PDF p. 4; evaluated cells: Table 3, PDF p. 5.

# LongBench: short replay restores most average quality

| Model | Full | Zero | 5% replay | 10% replay |
|---|---|---|---|---|
| Qwen3.6-27B | .426 | .313 | .410 (96.2%) | .417 (97.8%) |
| Qwen3.5-4B | .374 | .271 | .369 (98.9%) | .367 (98.1%) |
| OLMo-Hybrid-7B | .317 | .159 | .294 (92.8%) | .297 (93.9%) |

$$\text{Retention} = 100\% \times \text{Score}_{\text{replay}}/\text{Score}_{\text{Full}}$$

- Replay improves substantially over zero-only.
- Qwen3.5's average is slightly lower at 10% than at 5%.
- Length cells differ by model; use within-model comparisons.

Paper, Table 1, PDF p. 4. Parentheses reproduce reported retention; higher is better.

# RULER: strong recovery, with one reporting discrepancy

| Model | Full | Zero | 5% replay | 10% replay |
|---|---|---|---|---|
| Qwen3.6-27B† | .987 | .299 | .985 (99.8%) | .986 (99.9%) |
| Qwen3.5-4B | .960 | .415 | .959 (99.9%) | .958 (99.7%) |
| OLMo-Hybrid-7B | .812 | .215 | .756 (93.1%) | .785 (96.7%) |

- Qwen3.5 is already close to full prefill at 5%.
- OLMo benefits more from increasing the replay budget.

**† Keep the two reported data sources distinct**

Table 3 gives Qwen3.6 means .97125 / .348 / .970 / .97125.
These do not match Table 1 above; details are in backup.

Paper, Table 1, PDF p. 4; reader arithmetic from Table 3, PDF p. 5. Higher is better.

# Average retention hides larger task-specific losses

| Model | Task / length | Full | 5% | 10% |
|---|---|---|---|---|
| OLMo-Hybrid-7B | qasper / 16K | .394 | .308 | .318 |
| OLMo-Hybrid-7B | musique / 16K | .305 | .282 | .273 |
| Qwen3.5-4B | qasper / 16K | .487 | .482 | .461 |
| Qwen3.6-27B | qa1 / 8K | .895 | .880 | .885 |

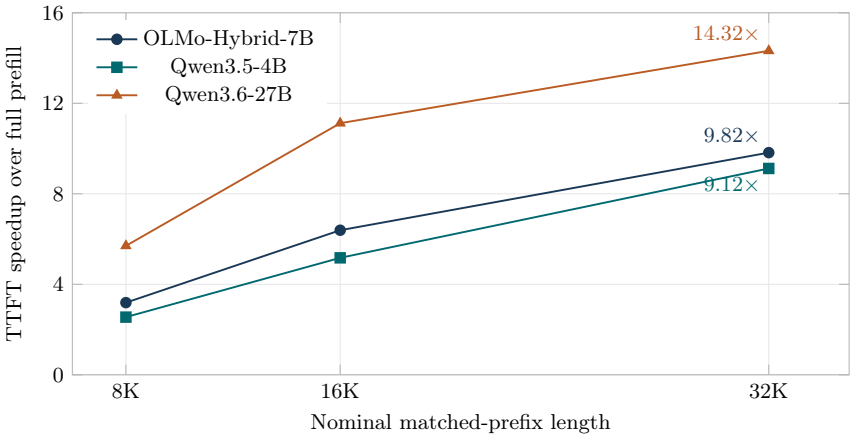
## OLMo on Qasper

5% retains only $0.308/0.394 = 78.2\%$ of the full score.
At 10%, retention rises to 80.7%, still below the benchmark average.

More replay does not monotonically improve every task score.

Paper, Appendix A, Table 3, PDF p. 5. Retention percentages on this slide are reader calculations.

# Longer matched prefixes yield larger TTFT speedups




Redrawn from Table 2, PDF p. 4. 5% replay, H2D-OVL+skip; speedup versus same-row full prefill. Nominal lengths; exact token counts on the next slide / backup.

# At 32K, doubling the budget can raise replay cost sharply

| Model | Matched tokens | Full (ms) | 5% (ms) | 10% (ms) |
|---|---|---|---|---|
| OLMo-Hybrid-7B | 31,366 | 1068.3 | 108.8 | 112.5 |
| Qwen3.5-4B | 31,847 | 786.4 | 86.2 | 108.1 |
| Qwen3.6-27B | 31,847 | 3605.2 | 251.8 | 371.3 |

- Both replay columns use H2D-OVL+skip.
- Qwen3.6: 5% $\rightarrow$ 10% increases TTFT by **47.5%**.
- Its speedup drops from 14.32$\times$ to 9.71$\times$.
- OLMo's corresponding latency increase is only 3.4%.

Paper, Table 2, PDF p. 4. Percentage increases computed from the reported means.

# Overlap and FFN skip jointly cut recovery overhead

| Model, about 32K | SER (ms) | OVL+skip (ms) | Reduction |
|---|---|---|---|
| OLMo-Hybrid-7B, 5% | 158.8 | 108.8 | 31.5% |
| Qwen3.5-4B, 5% | 105.8 | 86.2 | 18.5% |
| Qwen3.6-27B, 5% | 311.9 | 251.8 | 19.3% |
| OLMo-Hybrid-7B, 10% | 191.7 | 112.5 | 41.3% |

- These reductions use serialized Tail-Replay as the denominator.
- Table 2 tests the two optimizations together.
- Individual effects of overlap and FFN skip are not separated.

Paper, Table 2, PDF p. 4. Reductions computed as 1 - (OVL+skip)/SER.

# Checkpoint-free recovery still has a storage cost

## Extra cache content

Store an output hidden for every token at every FA layer.

With uniform layer widths and $b$ bytes per element,

$$M_{\text{cache}} \approx nL_{\text{FA}}b(d_K + d_V + d_h).$$

- $n$: cached tokens; $L_{\text{FA}}$: number of FA layers.
- $d_K, d_V$: total KV widths; $d_h$: hidden width.
- Hidden storage adds $nL_{\text{FA}}bd_h$ bytes.
- The paper gives no equal-memory comparison with checkpoints.

Reader cost model from the cache definition in Sec. 3, PDF p. 3; not a reported memory measurement.

# The measured gain is a cache-hit TTFT gain

- Timing starts at cache transfer or replay and ends at the first token.
- Results include this recovery path's transfer cost.
- Queueing, cache lookup, and concurrent serving throughput are not evaluated by Table 2.
- Quality tasks and latency prefixes use different length sets.
- Quality retention is an aggregate score ratio, not output equivalence.

## The next decisive comparison

Tail-Replay versus checkpoint recovery under the same cache capacity and request distribution.

Measurement definition: Sec. 4 and Table 2, PDF p. 4. Proposed comparison is reader analysis.

# Tail-Replay trades state exactness for flexible reuse

1. **Design:** exact FA KV and hidden inputs; approximate LA recovery from a short tail.
2. **Evidence:** strong recovery over zero-only; 9.12–14.32× TTFT speedup at about 32K with 5% replay.
3. **Tradeoff:** task-dependent losses, hidden storage, and budget-dependent replay latency.

## Discussion

What signal could choose a tail length per group or request, with a checkpoint fallback when recovery is unreliable?

Paper, Secs. 3–5 and Tables 1–3, PDF pp. 3–5. Adaptive replay / fallback is a proposed extension.

# Primary reference and evidence map

**Yirui Liu, Ruoling Qi, Xuaner Wu, Penghang Liu, Jian Chen.**

*Tail-Replay: Escaping the Curse of Linear Attention in Prefix Caching for Hybrid LLMs.*

arXiv:2608.30310v2, revised October 15, 2026.

https://arxiv.org/abs/2608.30310v2

- Mechanism: Secs. 2–3, Eqs. (1)–(2), Fig. 1, pp. 2–3.
- Aggregate quality and TTFT: Tables 1–2, p. 4.
- Per-task quality: Appendix A, Table 3, p. 5.

All paper page references use 1-based PDF page numbers. Following slides are backup material.

# Backup: the Qwen3.6 RULER averages do not reconcile

| Source | Full | Zero | 5% | 10% |
|---|---|---|---|---|
| Table 1 | .987 | .299 | .985 | .986 |
| Mean of Table 3 cells | .97125 | .34800 | .97000 | .97125 |

Example: the eight full-prefill recall values give

$$\frac{1+1+.895+.875+1+1+1+1}{8} = .97125.$$

- Appendix-based retention is 99.87% / 100.00%.
- The differences exceed rounding at the displayed precision.
- No alternative aggregation rule is stated; keep both sources labeled.

Paper, Table 1, PDF p. 4 versus Table 3, PDF p. 5; appendix means computed by the reader.

# Backup: exact 8K and 16K timing rows

| Model | Matched tokens | Full (ms) | 5% (ms) | 10% (ms) |
|---|---|---|---|---|
| OLMo-Hybrid-7B | 8,225 | 264.3 | 82.9 | 82.7 |
| OLMo-Hybrid-7B | 16,417 | 533.7 | 83.5 | 85.6 |
| Qwen3.5-4B | 8,226 | 191.6 | 75.2 | 75.9 |
| Qwen3.5-4B | 16,418 | 387.7 | 75.0 | 76.0 |
| Qwen3.6-27B | 8,226 | 879.6 | 154.3 | 154.7 |
| Qwen3.6-27B | 16,418 | 1797.2 | 161.6 | 218.1 |

Both replay columns use H2D-OVL+skip.

These are means over 20 timed repetitions, not throughput measurements.

Paper, Table 2, PDF p. 4.

# Backup: contraction needs more than a forgetting gate

For the paper's operator form, a sufficient nonexpansion condition is

$$\|k_i\|_2 = 1, \quad \beta_i \in [0, 2] \quad \Rightarrow \quad \|I - \beta_i k_i k_i^\top\|_2 \leq 1.$$

- A uniform $\alpha_i \leq \rho < 1$ then yields a $\rho^k$ bound.
- The paper explicitly states only $\alpha_i \in (0, 1]$.
- With perturbed replay inputs, the general error includes

$$E_i = T_i E_{i-1} + (T_i - \hat{T}_i)\hat{S}_{i-1} + (b_i - \hat{b}_i).$$

Reader derivation from Eq. (1), PDF p. 2. $b_i$ is the additive write; hats denote replay quantities.

# Backup: matrix layout matters when implementing Eq. (1)

The paper writes $T_i S_{i-1} + \beta_i v_i k_i^\top$ without specifying axes.

For a value-by-key state layout,

$$S_i \in \mathbb{R}^{d_v \times d_k}, \quad k_i \in \mathbb{R}^{d_k}, \quad v_i \in \mathbb{R}^{d_v},$$

a key-space operator acts on the *right*:

$$S_i = \alpha_i S_{i-1}(I - \beta_i k_i k_i^\top) + \beta_i v_i k_i^\top.$$

- A transposed state layout requires transposing the write term too.
- The main derivation treats $T_i$ as an abstract state operator.

Reader dimensional clarification of the notation in Eq. (1), PDF p. 2; model head dimensions are not supplied.