LoLA: Low-Rank Linear Attention with Sparse Caching

Selective memory for a fixed-state language model

Paper by Luke McDermott, Robert W. Heath Jr., and Rahul Parhi

Independent paper-reading presentation

arXiv:2505.23666v3 June 13, 2026

Reading date: September 24, 2026

Keep hard-to-reconstruct associations explicit; compress the rest.

Memory allocation changes recall at fixed weights

4K single-needle retrieval, same nominal 512 KV slots

Extended sliding window: 8.8% $\longrightarrow$ LoLA: 97.4%

- LoLCATs+: $(\eta, \lambda) = (512, 0)$; LoLA: $(256, 256)$.
- +88.6 percentage points with a different allocation of historical KV pairs.
- The headline 0.6% baseline uses only a 64-token window, so that comparison also changes capacity.

Sec. 4.1, Table 1, PDF p. 7. All models here use the same distilled 8B base weights.

A fixed matrix can mix incompatible associations

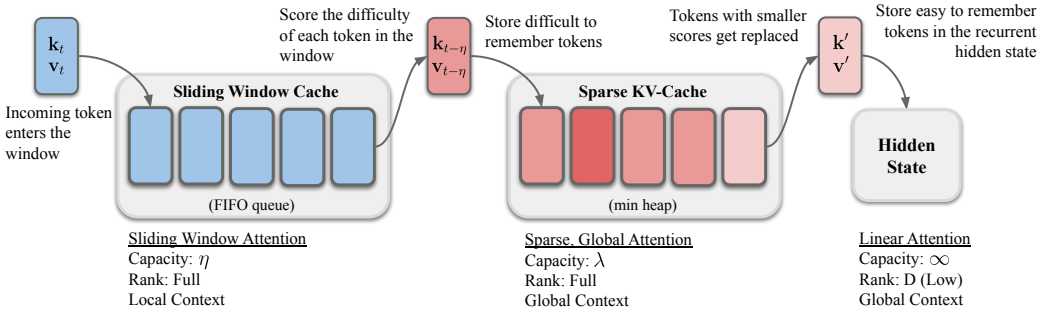
$$H_t = H_{t-1} + \phi(k_t) v_t^\top, \qquad s_t = s_{t-1} + \phi(k_t)$$

$$m_t(q) = \frac{H_t^\top \phi(q)}{\phi(q)^\top s_t}$$

- $H_t \in \mathbb{R}^{D \times d_v}$; $s_t \in \mathbb{R}^D$; $\phi : \mathbb{R}^{d_k} \to \mathbb{R}^D$.
- Non-orthogonal feature keys can interfere: a new write changes the readout for an old key.
- A sliding window preserves recent detail; old associations still enter the compressed state.

Secs. 2.2–2.4, Eqs. (3)–(5), PDF pp. 3–4. Equations transposed to column-vector outputs.

LoLA assigns each past KV pair to one memory



Local window
Recent exact KV
Capacity η

Sparse global cache
Hard associations
Capacity λ

Linear state
Compressed history
Fixed dimensions

Sec. 3, Fig. 1, PDF p. 5. "Infinite capacity" in the original schematic means unbounded input count, not exact recall.

SRE asks whether a key retrieves its own value

Self-recall error

$$\mathrm{SRE}(k, v \mid H, s) = \left\| \frac{H^\top \phi(k)}{\phi(k)^\top s} - v \right\|_2$$

- Large error: the pair does not fit the current associative map well.
- Selection is query-agnostic; reading the selected KV is query-dependent.
- SRE measures present reconstruction difficulty, not proven future importance.

Sec. 3, Eq. (10), PDF p. 4. L2 distance, not squared loss; no accuracy units.

Re-score explicit candidates; compress the easiest

$$\mathcal{E}_t = G_{t-1} \cup \{\text{KV leaving the local window}\}$$

1. Score candidates against the current, pre-update state.
2. Keep the λ highest errors in G_t.
3. Add the remaining outer products to H and their feature keys to s.

A greedy, one-way decision

Cached pairs are re-scored as the state changes. Pairs already compressed into H cannot re-enter the candidate set as raw KV.

Sec. 3, Eqs. (11)–(13), PDF p. 5; Appendix H, pp. 27–28 clarifies score-before-write order.

All three branches share one normalizer

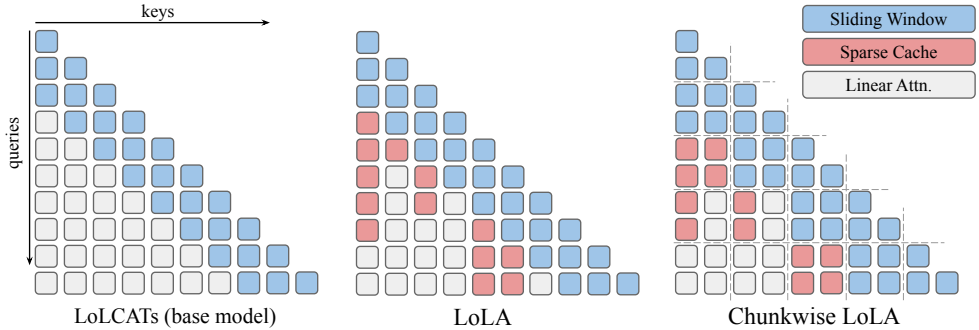
Let $a_{ti} = \exp(q_t^\top k_i/\sqrt{d_k})$ and W_t be the local window.

$$y_t = \frac{H_t^\top \phi(q_t) + \sum_{i\in G_t} a_{ti} v_i + \sum_{j\in W_t} a_{tj} v_j}{\phi(q_t)^\top s_t + \sum_{i\in G_t} a_{ti} + \sum_{j\in W_t} a_{tj}}$$

- The state contains only KV assigned to compressed memory.
- Each historical pair contributes through its assigned branch.
- This is a joint normalized mixture, not a sum of three independently normalized outputs.

Sec. 3, Eq. (14), PDF pp. 5–6. Column-vector convention used throughout this deck.

Chunking reduces the number of serial cache updates



- Queries in a chunk share a state; local and sparse KV use one explicit attention pass.
- Re-score after a chunk: fewer updates, more local parallel work.
- Cache selection remains state-dependent across chunks.

Sec. 3, Fig. 2 and Eq. (15), PDF p. 6. The paper mixes chunk/window notation; implementation must resolve it.

Training-free augmentation still needs a trained base

LoLCATs base	LoLA at inference
Llama-3.1-8B or Llama-3.2-1B	Reuse the base weights
Window + linear attention, $D = 2d_k$	Route KV using SRE
Feature-map distillation, then LoRA	Choose window and sparse budgets
40M Alpaca tokens; length 1024	No additional LoLA training

The 40M-token figure excludes the teacher's original pretraining.

Secs. 2.3 and 4, PDF pp. 3–4, 6–7. Base training window: $\eta = 64$; two epochs for each stage.

Harder needles require more sparse capacity

Model (η, λ)	SN-1 / 4K	SN-2 / 2K	SN-3 / 2K
LoLCATs+ $(128, 0)$	1.4	7.2	3.2
LoLA $(64, 64)$	69.4	3.0	1.6
LoLCATs+ $(512, 0)$	8.8	21.6	10.6
LoLA $(256, 256)$	97.4	85.4	27.2
LoLCATs+ $(1024, 0)$	30.4	60.0	27.6
LoLA $(512, 512)$	100	100	100

- Accuracy in percent; pairs of rows match nominal explicit KV slots.
- SN-3 uses a 32-digit UUID. At a small budget, LoLA can lose to a larger local window.

Table 1, PDF p. 7; needle definitions: Appendix C, p. 20. Llama scores 100 throughout this table.

Hard RULER tasks improve, but the gap remains large

4K task	LoLCATs+	LoLA	Mamba2	Llama
Multi-query	3.3	67.6	49.1	90.5
Multi-value	3.6	65.0	35.0	98.5
Variable tracking	0.7	85.2	76.5	99.6
Frequent words	13.6	51.3	76.6	91.7
HotpotQA	14.2	24.2	31.8	62.0
All 10 tasks: average	6.7	45.2	39.7	91.7

LoLA: $(128, 768)$; LoLCATs+: $(896, 0)$.

Average gain over the matched base: 38.5 points. Gap to Llama: 46.5 points.

Table 2, PDF p. 7. Selected task rows; average uses all 10. Other architectures have different training histories.

Lambada gains are clearer than broad superiority

Scale	Metric	LoLCATs	LoLA	Teacher
8B	Lambada accuracy	67.6	74.9	73.0
8B	MMLU, 5-shot	54.9	57.6	68.0
8B	WinoGrande	73.6	73.6	73.9
1B	Lambada accuracy	53.4	61.9	60.1
1B	Lambada perplexity ↓	9.3	5.3	–
1B	WinoGrande	61.5	60.9	60.5

- 8B uses $(64, 64)$; the 1B table does not separately state cache settings.
- MMLU is an explicit exception to the "zero-shot" description.

Tables 3 and 5, PDF pp. 8, 16. 1B WinoGrande base uses a cited score; authors' reproduced base was also 60.9.

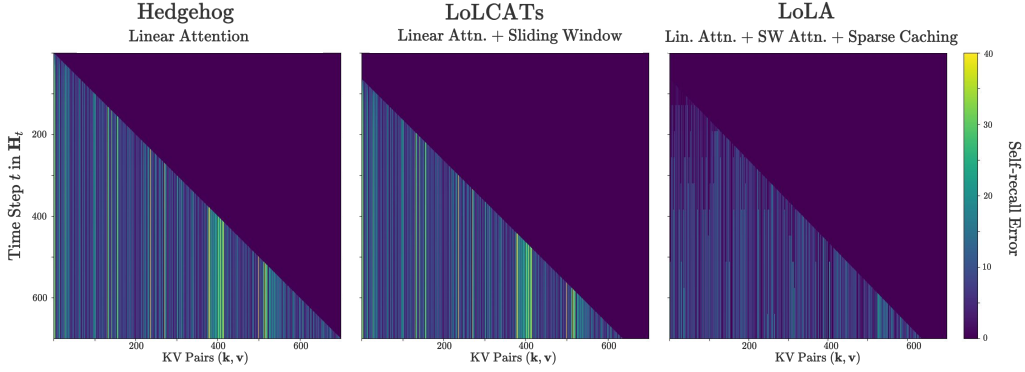
SRE outperforms the tested selection rules

Selection rule	SN-1 / 512 tokens
Self-recall error	99.0%
Squared kernel approximation error	11.4%
Absolute kernel approximation error	20.0%
Linear/exponential kernel ratio	52.0%
Accumulated exponential query–key similarity	10.6%
Extended local window, no sparse cache	29.4%

All scoring variants use $(64, 64)$; the local-only baseline uses 128 slots.
Query-dependent alternatives accumulate scores during the local window.

Table 4, PDF p. 10; Appendix F, p. 26. Evidence is from one synthetic task and one context length.

The visualizations support reduced memory interference



- Horizontal: KV position. Vertical: state time. Brighter: larger SRE.
- A local window delays interference; LoLA reduces it in this example.
- Zero columns for cached pairs are not proof of zero reconstruction error inside the linear state.

Sec. 4.3, Fig. 3, PDF p. 9. Appendix G uses an error difference from initial insertion, not a ratio.

Sparse and compressed memories work together

Readable branches	SN-1 2K	SN-1 4K	SN-2 2K	SN-3 2K
State + sparse + window	100	100	100	100
Sparse + window	14	6	2	0
State + window	26	14	24	10
Window only	18	8	0	0

What this ablation actually changes

All caches are still updated normally at $(512, 512)$. Only query access is restricted; hidden KV are not re-routed into another branch.

Appendix I, Table 12, PDF pp. 28–29. Ablation runs: 50 samples each. This does not test removing the window.

Natural long text improves without closing the gap

LongBench task	LoLCATs+	LoLA	Mamba2
MultiFieldQA	6.1	30.8	29.3
HotpotQA	4.4	11.8	30.0
MuSiQue	1.5	3.4	11.1
TREC	37.5	57.5	54.0
TriviaQA	5.8	50.2	77.77
All 7 tasks: average	9.0	27.8	36.1

LoLA $(512, 1024)$ and LoLCATs+ $(1536, 0)$: same nominal slot count.
+18.8 average points over the base; still 8.3 points behind Mamba2.

Appendix C, Table 11, PDF p. 21. Task-specific benchmark scores, not uniformly accuracy. Full evaluation settings are not reported.

Bounded state does not ensure long-range accuracy

Large-cache LoLA ("4+")	2K	4K	8K	16K
SN-1 accuracy	100%	99.0%	92.2%	13.6%
Reported compression	1.2×	2.4×	4.9×	9.8×

- Distillation sequences were length 1024; 8K performance is useful extrapolation.
- Compression improves with length while recall can collapse.
- The paper proposes a direction toward lifelong context, not a demonstration of lifelong reliable memory.

Appendix C, Table 10, PDF p. 21. 500 samples per length, except 16K: 250. Main Table 1 reports 100 at 4K for its largest cache.

Prefill overhead is visible; decode changes little

Comparison: $\lambda = 0 \rightarrow 512$	Base	Sparse	Ratio
TTFT, $\eta = 512$	0.57 s	0.63 s	1.11×
TTFT, $\eta = 64$	0.99 s	1.46 s	1.47×
Layer forward, $\eta = 512$	4.2 ms	9.7 ms	2.31×
Layer forward, $\eta = 64$	18.3 ms	35.5 ms	1.94×

- RTX 4090; 4K prefill. TTFT excludes input loading; 100 trials.
- Layer: batch 2, width 4096, 32 heads; 250 trials.
- Decode after 1K prefill: 27.9–28.2 tok/s across tested settings.

Appendix B, Fig. 4 and Tables 7–8, PDF pp. 16–18. Ratios recalculated here; these are not speedups over Llama.

Window and sparse budgets trade speed for recall

Window η	Sparse λ	TTFT ↓	VRAM ↓	VT accuracy ↑
64	0	0.99 s	18.1 GB	0.0%
64	512	1.46 s	18.6 GB	78.3%
256	512	0.71 s	18.7 GB	86.9%
512	0	0.57 s	18.5 GB	0.0%
512	512	0.63 s	19.0 GB	72.2%

Larger local windows reduce cache-update frequency.
Accuracy need not increase monotonically with the window size.

Budget selection depends on the workload

Sparse capacity helps recall; local capacity enables faster prefill. Both contribute to memory use.

Selected cells transcribed from Appendix B, Figs. 4–6, PDF pp. 17, 19. LoLA-8B, RTX 4090, 4K context.

Persistent attention state is bounded by the budget

$$M_{\text{full}}(t) = t(d_k + d_v)$$

$$M_{\text{LoLA}}(t) = \min\{t(d_k + d_v),\ (\eta + \lambda)(d_k + d_v) + Dd_v + D\}$$

- Counts elements per head: explicit KV, hidden matrix, normalization vector.
- With fixed budgets, the state saturates; total model VRAM is a different quantity.
- Reported 4K total VRAM: about 18.1–19.0 GB across tested settings.

Appendix B, Figs. 5 and 7, PDF pp. 19–20. The headline 4.6× ratio is author-reported; counting ambiguity is discussed in backup.

The strongest conclusion is narrower than a universal fix

1. **Architecture scope:** only two LoLCATs-derived bases are tested. Extension to nonlinear test-time memory remains future work.
2. **Capacity scope:** cached pairs can be reconsidered; already compressed pairs cannot be recovered as raw KV.
3. **Evidence scope:** limited statistical reporting, mixed-source baselines, and missing details for some evaluations.

Practical reading

Use the same-weight, matched-budget comparisons to assess the mechanism. Treat cross-model rankings and precise compression claims more cautiously.

Sec. 5, PDF p. 10; Appendices A–C and H–I. Scope assessment by this presentation.

A useful rule: reserve exact memory for poor fits

Take-home message

LoLA improves recall by protecting a compressed associative map from difficult KV pairs while keeping those pairs explicitly accessible.

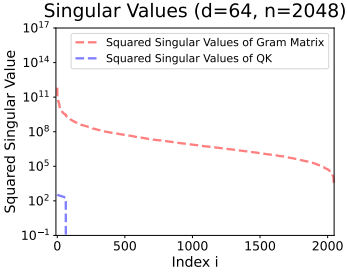
- The allocation rule matters beyond the number of KV slots.
- Sparse and compressed branches jointly support retrieval.
- Bounded storage does not remove accuracy and latency trade-offs.

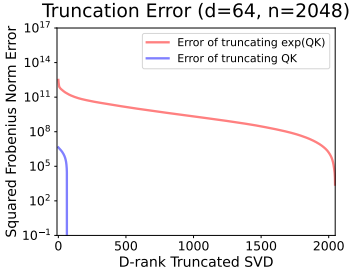
Luke McDermott, Robert W. Heath Jr., and Rahul Parhi.
LoLA: Low-Rank Linear Attention With Sparse Caching.
TMLR, June 2026. arXiv:2505.23666v3.

Main presentation: approximately 25 minutes. Backup: theory, metrics, interpretation, and reproducibility.

Backup: low-rank error is about the kernel matrix

$$G_{ij} = e^{x_i^\top x_j}, \quad \operatorname{rank}(\widehat{G}) \le D \quad \Longrightarrow \quad \|G - \widehat{G}\|_F^2 \ge \sum_{i>D} \sigma_i^2.$$





Scaled Gaussian inputs; $d_k = 64, n = 2048$. An algebraic SVD bound plus an empirical spectrum, not a universal task-accuracy lower bound.

Appendix E, Eq. (21), PDF p. 24; Fig. 8(c,f), p. 25. Unnormalized kernel; vertical axes are logarithmic.

Backup: the full 4K RULER breakdown

Task	LoLCATs+	LoLA	Mamba2	Llama
MK1	12.8	39.4	40.3	100
MK2	1.4	11.6	13.8	100
MK3	0.4	7.6	5.5	98
MQ	3.3	67.6	49.1	90.5
MV	3.6	65.0	35.0	98.5
VT	0.7	85.2	76.5	99.6
CWE	3.0	45.9	32.9	95.2
FWE	13.6	51.3	76.6	91.7
HotpotQA	14.2	24.2	31.8	62.0
SQuAD QA	14.0	53.9	35.5	81.5
Average	6.7	45.2	39.7	91.7

Table 2, PDF p. 7. LoLA $(128, 768)$; LoLCATs+ $(896, 0)$. Llama: 50 samples/task.

Backup: 8B language scores and metric definitions

Metric	LoLCATs	LoLA	Llama
PIQA accuracy	81.0	81.6	81.1
ARC-e accuracy	82.4	82.5	81.7
ARC-c normalized accuracy	54.4	55.4	55.1
HellaSwag normalized accuracy	79.1	79.8	79.3
WinoGrande accuracy	73.6	73.6	73.9
MMLU, 5-shot	54.9	57.6	68.0
Lambada accuracy	67.6	74.9	73.0

Table 6's six-task average excludes MMLU. Its teacher-relative score does not establish parity across all seven metrics.

Table 3, PDF p. 8; Table 6, p. 17. LoLA uses $(64, 64)$. No uncertainty intervals are reported.

Backup: some v3 reporting details need qualification

- **4K, large cache:** Table 1 gives 100; Tables 9–10 give 99.0. The reason is not stated.
- **Table 6:** several 1B models are labeled 1.3B, unlike the main text and Table 5.
- **Cross-teacher comparison:** Mamba2-Llama3's 43.2 is MMLU in Table 3 but appears under Lambada in Table 6.
- **Window versus chunk:** η and C are used inconsistently across the method and efficiency discussion.

These issues do not erase the matched-budget recall gain; they limit exact reproduction and broader comparisons.

Tables 1, 3, 5, 6, 9–10, PDF pp. 7–8, 16–17, 21; Sec. 3 and Appendix B. Reading audit, not author claims.

Backup: the 4.6× ratio is not independently resolved

- Authors report 4.6× smaller cache at 4K for $(\eta, \lambda) = (256, 256)$.
- Using their per-head formula, $d_k = 128, D = 256$, and *assuming* $d_v = 128$ gives:

$$\frac{4096(128 + 128)}{512(128 + 128) + 256 \cdot 128 + 256} = 6.39.$$

How to report this responsibly

Retain 4.6× as an author-reported number. Resolve physical window length and head-sharing conventions before claiming an exact implementation-level ratio.

Abstract; Sec. 4.1, pp. 7–8; Appendix B, pp. 19–20; head dimensions: Appendix E, p. 24. Conditional calculation by this deck.

Backup: implementation and evidence checklist

For an implementation

- Score before writing to H.
- Handle empty state and zero denominator.
- Enforce exact Top-λ capacity, including ties.
- Preserve causal masks and shared normalization.

Author's minimal code link, cited in Appendix H:
github.com/lukemcdermotttt/LoLA-release

For this reading

- Same-version PDF and author TeX.
- Main text and Appendices A–I covered.
- Original figures, with page locators.
- No model experiments reproduced or repository audited.

Appendix H, PDF pp. 27–28. Complete Chinese notes and a portable build helper accompany this deck.