Local Linear Attention

An Optimal Interpolation of Linear and Softmax Attention for Test-Time Regression

Yifei Zuo, Yutong Yin, Zhichen Zeng, Ang Li,
Banghua Zhu, Zhaoran Wang

arXiv:2510.01450v1 Published at ICLR 2026 Official code

Paper discussion 3 November 2026

A query-specific linear fit improves local prediction

One-sentence takeaway

LLA fits a local slope and intercept around each query; FlashLLA makes the resulting solve memory-feasible.

- Statistics: local linear regression addresses boundary bias of local averaging.
- Systems: pairwise differences and per-query covariance matrices need not be stored.
- Evidence: strong synthetic non-stationary regression and recall; state tracking remains weak.

Paper, Abstract and Secs. 2–5, PDF pp. 1, 3–11.

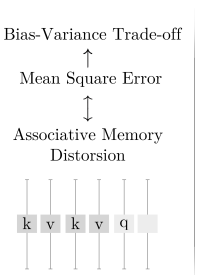
Attention as a regression solver at test time

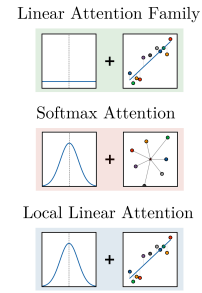
$$\mathcal{D}_i = \{(k_j, v_j) : j \le i\}, \qquad \hat{f}_i \leftarrow \mathrm{Fit}(\mathcal{D}_i), \qquad o_i = \hat{f}_i(q_i).$$

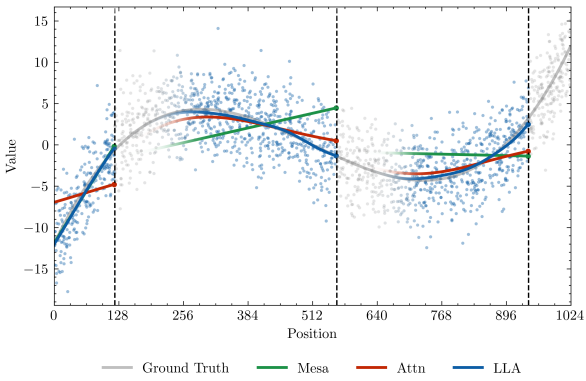
Mechanism	Local hypothesis	Main consequence
Linear / MesaNet	one global linear fit	compact state; model bias
Softmax	constant near each query	local, but boundary bias
LLA	slope + intercept near each query	local and first-order correction

Paper, Sec. 2.1 and Fig. 1, PDF pp. 2–4.

The three estimators make different errors







Global misspecification → local averaging → local slope correction.

Original paper, Fig. 1, PDF p. 2. Illustrative regression diagrams.

Global linear regression can retain non-vanishing bias

- If the true mapping f is not globally affine, every global affine fit has an approximation-error floor.
- A local-constant kernel fit can shrink its neighborhood with sample size.

$$\mathbb{E}\int_D \|\hat{f}_{\mathrm{GL}} - f\|^2 = \Omega(1), \qquad \mathbb{E}\int_D \|\hat{f}_{\mathrm{NW}} - f\|^2 = O\Big(n^{-3/(d+3)}\Big).$$

Interpretation

Locality can eliminate global model misspecification, although local averaging still has a boundary problem.

Paper, Proposition 2.1, PDF p. 4; proof in Appendix A, pp. 15–19. Conditional asymptotic result.

A local slope removes first-order boundary bias

Estimator	Integrated squared bias	Variance
Local constant	$O(h^3)$	$O((nh^d)^{-1})$
Local linear	$O(h^4)$	$O((nh^d)^{-1})$

$$\mathrm{IMSE}_{\mathrm{NW}} = \Omega\Big(n^{-3/(d+3)}\Big), \qquad \mathrm{IMSE}_{\mathrm{LL}} = O\Big(n^{-4/(d+4)}\Big).$$

- The boundary layer has width $O(h)$; its local-constant bias is $O(h)$.
- Integrated MSE over D; requires a sufficiently large boundary-normal gradient and optimal bandwidths.

Paper, Proposition 2.2, PDF p. 5; Appendix B, pp. 20–24.

LLA predicts with the intercept at the query

$$z_{ij} = k_j - q_i, \qquad \min_{b,W} \frac{1}{2} \sum_{j \le i} w_{ij} \| v_j - b - W z_{ij} \|_2^2 + \lambda \| W \|_F^2.$$

Locality

w_{ij} emphasizes keys near the current query q_i.

First-order structure

W captures a local slope; $b = \hat{f}(q_i)$ is the output.

Paper, Sec. 2.3, Eq. (6), PDF p. 5.

The optimal intercept is a corrected weighted sum

$$\omega_i = \sum_j w_{ij}, \qquad \mu_i = \sum_j w_{ij} z_{ij},$$

$$\Sigma_i = \sum_j w_{ij} z_{ij} z_{ij}^\top + \lambda I, \quad \rho_i = \Sigma_i^{-1} \mu_i.$$

$$o_i = \sum_{j \le i} s_{ij} v_j, \qquad s_{ij} = \frac{w_{ij}(1 - z_{ij}^\top \rho_i)}{\omega_i - \mu_i^\top \rho_i}$$

- $\sum_j s_{ij} = 1$, but individual s_{ij} may be negative.
- As $\lambda \to \infty$, $\rho_i \to 0$: recover local-constant weights w_{ij}/ω_i.

Paper, Eqs. (7)–(8), PDF p. 5.

The interpolation is a residual correction

$$o_i = \hat{W}_i q_i + \sum_{j \le i} \frac{w_{ij}}{\omega_i} (v_j - \hat{W}_i k_j)$$

1. Predict from the fitted linear map $\hat{W}_i$.
2. Locally average its residuals to correct misspecification.

Meaning of "optimal interpolation"

The local weighted regression objective determines $\hat{W}_i$ and the correction; this is not a fixed convex mixture of two attention outputs.

Paper, Sec. 2.3, Eq. (9), PDF p. 5.

Two algebraic changes remove the memory bottlenecks

Naive object	Memory	Replacement
All $k_j - q_i$	$\Theta(n^2 d)$	separate key and query terms
All Σ_i	$\Theta(nd^2)$	matrix-free CG products

$$\tilde{\mu}_i = \sum_j w_{ij} k_j, \qquad \mu_i = \tilde{\mu}_i - \omega_i q_i.$$

$$\Sigma_i p = \sum_j w_{ij} (k_j - q_i)\big((k_j - q_i)^\top p\big) + \lambda p.$$

Paper, Sec. 3.1, Eqs. (10)–(12), PDF p. 6.

FlashLLA streams key blocks in three stages

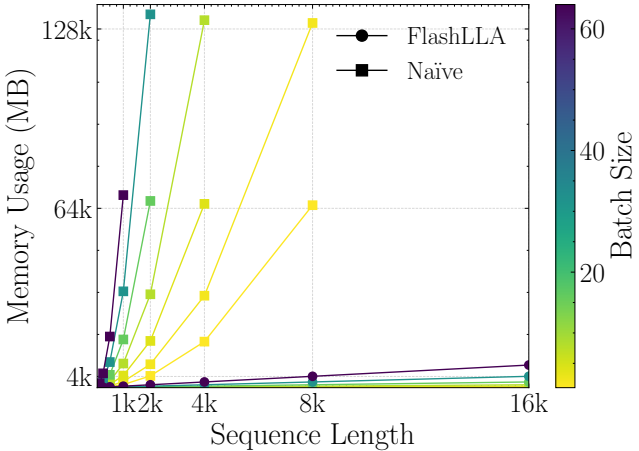
1. **Statistics:** scan key blocks; update ω_i and μ_i with online-max stabilization.
2. **Solve:** run CG for $\Sigma_i \rho_i = \mu_i$ without storing Σ_i.
3. **Output:** rescan key/value blocks and accumulate $\sum_j s_{ij} v_j$.

Trade-off

Working memory becomes $\Theta(nd)$, while each CG iteration streams keys again. Compute and I/O remain substantial.

Paper, Algorithm 1, PDF p. 7; CG details in Appendix C, pp. 24–25.

The measured gain is peak memory, not runtime



- One H200 GPU
- Head dimension $d = 128$
- Several batch sizes
- Naive method reaches memory limits quickly

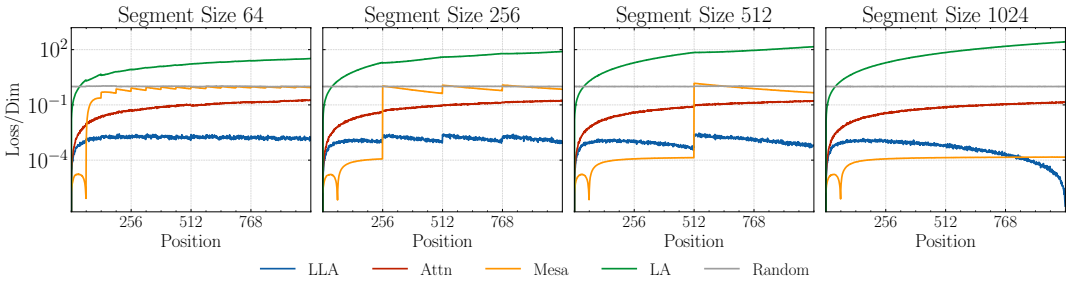
Original paper, Fig. 2, PDF p. 7; profiling description, p. 8.

Four tasks separate adaptation, recall, and state updates

Task	Question	Metric
Test-time regression	Adapt without trained projections?	position MSE $\downarrow$
In-context regression	Learn from changing linear segments?	test MSE $\downarrow$
MQAR recall	Retrieve discrete key–value pairs?	accuracy $\uparrow$
State tracking	Apply sequences of swaps?	accuracy $\uparrow$

Paper, Sec. 4, PDF pp. 8–10; protocols in Appendix E, pp. 27–29.

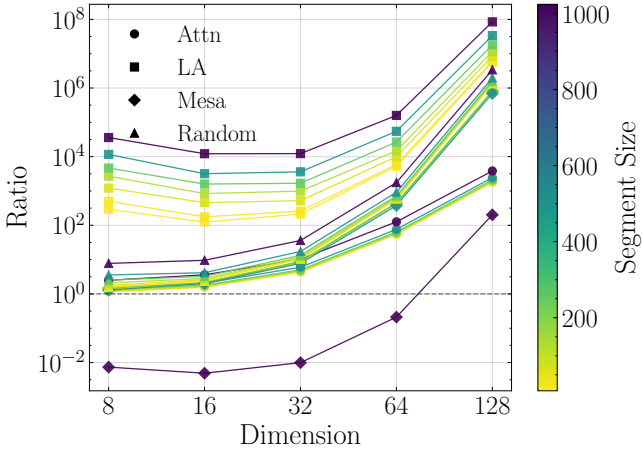
LLA adapts after piecewise-linear shifts



- $L = 1024$, $d = 64$; 10,000 sampled sequences; lower Loss/Dim is better.
- MesaNet can win before the first shift; LLA stays low after shifts.

Original paper, Fig. 3, PDF p. 8; configuration in Appendix E.2, p. 28.

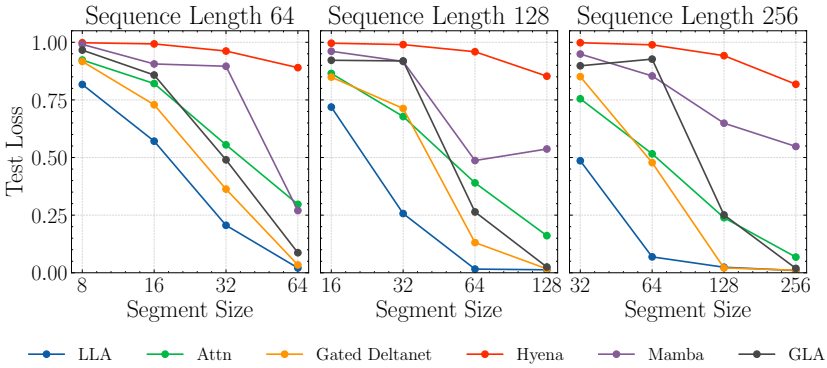
The benefit grows with dimension, with exceptions



Original paper, Fig. 4 and Sec. 4, PDF p. 8. Log axes.

- Ratio = comparator total MSE / LLA total MSE.
- Above 1 favors LLA.
- MesaNet beats LLA for some low-dimensional, long-segment settings.

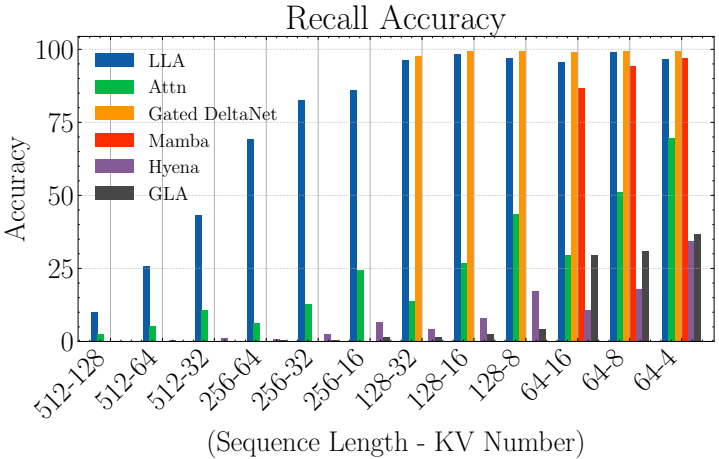
Trained LLA leads on non-stationary in-context regression



- Two-layer models without MLPs; $d_x = d_y = 32$, model width 128, two heads.
- At $L = 256, S = 64$: test loss ≈ 0.07 for LLA versus ≈ 0.52 for Softmax (read from plot).

Original paper, Fig. 5(a), PDF p. 9; three-seed best-hyperparameter scores.

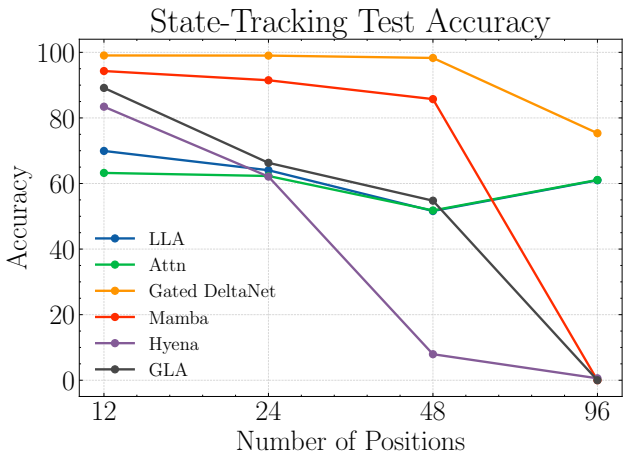
LLA's largest recall gains appear on hard long contexts



Original paper, Fig. 5(b), PDF p. 9; approximate plot readings.

- MQAR vocabulary: 8,192 tokens.
- At 512–32, LLA $\approx$ 43% vs Softmax $\approx$ 11%.
- Gated DeltaNet matches or exceeds LLA in some easier cases.

LLA does not solve permutation state tracking



- LLA tracks Softmax closely.
- At $N = 48$: $\approx 52\%$ vs Gated DeltaNet $\approx 98\%$.
- Sequential swaps favor a state update mechanism.

Original paper, Fig. 6, PDF p. 10. First x-axis label is 12; Appendix E.2 lists 16.

The evidence supports a focused conclusion

- The rate comparison assumes smooth f, a smooth domain, a compact radial kernel, and tuned bandwidths; the experiments use piecewise data and exponentiated dot-product weights.
- FlashLLA demonstrates memory scaling, but CG adds computation and repeated I/O.
- Large-language-model training and real-language benchmark gains remain untested in this paper.

Practical question

Does a cheaper approximate slope preserve the regression/recall advantage at equal end-to-end compute?

Paper, Sec. 5, PDF pp. 10–11; assumptions in Appendices A–B, pp. 15–24.

Take-home messages

1. **Statistical design:** fit a query-specific local line rather than a local constant.
2. **Implementation:** algebraic separation plus matrix-free CG avoids quadratic and nd^2 intermediate storage.
3. **Observed trade-off:** strong adaptation and recall on tested synthetic tasks; weak sequential state tracking and higher compute cost.

$$\text{locality} + \text{first-order fit} \longrightarrow \text{better local prediction}$$

Synthesis of the paper's Secs. 2–5, PDF pp. 3–11.

Sources

- Y. Zuo et al., *Local Linear Attention: An Optimal Interpolation of Linear and Softmax Attention For Test-Time Regression*, arXiv:2510.01450v1 (2025).
- Official ICLR 2026 proceedings record.
- Author's FlashLLA repository (accessed 3 November 2026).

All plotted figures in this deck are from the arXiv v1 author source and are identified by figure number and PDF page on the corresponding slides.

Backup: where the rate exponents come from

$$\text{NW}: \quad h^3 + (nh^d)^{-1} \quad \Rightarrow h \asymp n^{-1/(d+3)}, \quad \text{MSE} \asymp n^{-3/(d+3)},$$

$$\text{LL}: \quad h^4 + (nh^d)^{-1} \quad \Rightarrow h \asymp n^{-1/(d+4)}, \quad \text{MSE} \asymp n^{-4/(d+4)}.$$

- The local-constant boundary layer has width $O(h)$ and pointwise squared bias $O(h^2)$.
- A local line removes the first-order term, leaving $O(h^4)$ squared bias.

Paper, Appendices A–B, PDF pp. 15–24.

Backup: experimental configurations that matter

Task	Configuration
Test-time regression	$L = 1024$; $d = 8, 16, 32, 64, 128$; 10,000 sequences
In-context regression	$L = 64, 128, 256, 512$; 100k train and 1k test per seed; 16 queries; train noise 0.1, test noise 0
MQAR	$L = 64, 128, 256, 512$; KV count $L/16, L/8, L/4$; short convolution disabled
State tracking	8 queries; swaps $S \sim \mathrm{Uniform}(N/6, N/3)$

Paper, Appendix E.2, PDF pp. 28–29.