Understanding Transformer from the Perspective of Associative Memory

Capacity, memory updates, and DeltaFormer

Paper authors: Shu Zhong, Mingyu Xu, Tenglong Ao, Guang Shi

arXiv:2505.19488v1 | Paper reading, 6 Nov 2026

Versioned arXiv paper

The paper connects retrieval capacity to write rules

One-sentence takeaway

Attention and FFNs can be read as associative memories; kernel choice controls retrieval interference, while the delta rule controls how old values are revised.

- **Capacity:** inverse retrieval SNR compares target signal with cross-key interference.
- **Update:** DeltaFormer combines kernel matching with residual writing.
- **Evidence:** a constructive swap argument and finite synthetic learning curves.

Paper, Secs. 1–3, PDF pp. 1–21.

Paper identity and code status

- **Paper read:** arXiv:2505.19488v1, submitted 26 May 2025.
- **Publication:** no formal publication record found for *this article* as of 6 Nov 2026.
- A distinct expanded DeltaFormer paper by overlapping authors appeared at NeurIPS 2025.
- **Code:** author toy code in Appendix E.4; third-party FLA implementation. No separate author repository located.

Paper, arXiv abstract and Appendix E.4; NeurIPS proceedings; FLA repository.

A memory has a write operation and a read operation

$$S_t = \sum_{i \le t} v_i \phi(k_i)^\top, \qquad o(q) = S_t \phi(q) = \sum_{i \le t} \kappa(k_i, q) v_i,$$

$$\kappa(x, y) = \phi(x)^\top \phi(y), \qquad S_t \in \mathbb{R}^{d_v \times d_\phi}.$$

- k_i indexes a value v_i; q retrieves by similarity.
- Context attention writes transient token associations.
- FFN weights hold persistent associations learned during training.

Paper, Sec. 2, PDF pp. 3–4. Softmax normalization is omitted in this algebraic form.

Inverse SNR measures cross-key interference

$$o(k_i) = \underbrace{v_i \kappa(k_i, k_i)}_{\text{signal } cv_i} + \underbrace{\sum_{j \neq i} v_j \kappa(k_j, k_i)}_{\text{noise } r},$$

$$\mathrm{SNR}^{-1} = \mathbb{E}\left[\frac{\|r\|^2}{c^2 \|v_i\|^2}\right] \approx N \frac{\mathbb{E}_{k_j}[\kappa(k_j, k_i)^2]}{\kappa(k_i, k_i)^2}.$$

Interpretation

Lower inverse SNR means cleaner recall for a stored key.

Paper, Sec. 2.1, Eqs. (5)–(10), PDF pp. 4–5; Appendix A, pp. 26–27. Independent Gaussian keys/values and comparable value norms.

The exponential kernel suppresses random-key noise

Kernel $\kappa(x, y)$	Approx. inverse SNR	Condition
$x^\top y$	N/d_k	linear
$\max(0, x^\top y)$	$N/(2d_k)$	ReLU
$\exp(x^\top y/\tau)$	$Ne^{-2(\tau-1)d_k/\tau^2}$	$\tau > 1$

- At $\tau = \sqrt{d_k}$, constant inverse SNR needs d_k on the order of $\log^2 N$, versus order N for a linear kernel.
- This is a Gaussian-key retrieval calculation, not a universal LLM capacity law.

Paper, Sec. 2.1, PDF pp. 5–7; Appendix A, pp. 27–28. N is the number of stored pairs.

An FFN is also a key–value lookup

$$\mathrm{FFN}(x) = W_V^\top \mathrm{ReLU}(W_K x) = \sum_{a=1}^{m} v_a \,\mathrm{ReLU}(k_a^\top x).$$

- Rows of W_K act as keys; rows of W_V provide values.
- The paper hypothesizes that a less selective FFN kernel helps compress many associations into fixed weights.
- Multihead, sparsity and gating are design axes discussed by the authors, not ablated for FFNs here.

Paper, Sec. 2.1.1, Eq. (14) and Table 1, PDF pp. 7–10.

The delta rule writes what the memory still misses

$$\text{Linear:} \quad S_t = S_{t-1} + v_t k_t^\top,$$
$$\text{DeltaNet:} \quad S_t = S_{t-1} + \underbrace{\left(v_t - S_{t-1} k_t\right)}_{\text{residual value}} k_t^\top.$$

- Linear attention keeps adding; related writes can accumulate.
- DeltaNet reads the existing value at k_t, then corrects it.
- Its update is one gradient step on $\frac{1}{2}\|S_{t-1}k_t - v_t\|^2$ (unit step size).

Paper, Sec. 2.2, Table 2, PDF pp. 10–12; Appendix B, pp. 29–30.

DeltaFormer performs residual writing with kernel matches

General recurrence

$$u_t = \alpha_t v_t - \beta_t \sum_{i<t} \kappa_1(k_i, w_t) u_i,$$

$$o_t = \sum_{i \le t} \kappa_2(k_i, q_t) u_i.$$

- κ_1 decides which previous writes to subtract; κ_2 reads the corrected values.
- α_t, β_t gate new information and the subtraction.
- $\beta_t = 0$ gives ordinary attention after output normalization; linear kernels with $w_t = k_t$ recover DeltaNet.

Paper, Sec. 2.2.1, PDF pp. 12–14.

The residual recurrence follows from a kernel delta update

$$
\begin{aligned}
S_t &= S_{t-1} + [v_t - S_{t-1}\phi(k_t)]\phi(k_t)^\top, \\
S_{t-1} &= \sum_{i<t} u_i \phi(k_i)^\top, \\
\Rightarrow \quad u_t &= v_t - \sum_{i<t} \kappa(k_i, k_t) u_i.
\end{aligned}
$$

- Exact algebra requires the same feature map for writing and reading.
- Independent κ_1 and κ_2 in the general model are a further design choice.
- Softmax normalization is restored in practical variants.

Paper, Sec. 2.2.1, PDF p. 13; Appendix C, PDF pp. 31–32.

The write dependency costs parallel depth

$$U = V - AU \quad \Longrightarrow \quad U = (I + A)^{-1} V, \qquad A_{t,i} = \kappa_1(k_i, w_t)\ (i < t).$$

- Naive sequence computation: $O(T^2 d)$ work, $O(T)$ serial steps.
- Invert a strictly lower triangular block of length C in parallel; advance across T/C blocks.
- The appendix gives $O(T^2 d + TCd + TC^2)$ work for this trade-off. No measured speedup is reported in this paper.

Paper, Sec. 2.2.1, PDF pp. 15–16; Appendix D.3, PDF pp. 33–35.

State exchange is a constructive, conditional result

- Initial unit keys are almost orthogonal: $|k_i^\top k_j| \le \epsilon < 1/8$ for $i \neq j$.
- A function f maps neighborhoods of $-1, 0, 1, 2$ *exactly* to those integers.
- For a swap of positions a, b, write $k_t = k_a - k_b$ and $v_t = 0$.
- Then queries k_a, k_b exchange their stored values; other queries stay unchanged.

Scope

The paper relates this construction to NC^1 state tracking; an "above TC^0" interpretation depends on the complexity separation assumption.

Paper, Appendix E, Theorem 1, PDF pp. 35–40.

Infinite context is a model-based extrapolation

$$S^* = (W_v \Sigma W_k^\top)(W_k \Sigma W_k^\top)^{-1},$$

$$W_k \text{ square/invertible} \quad \Rightarrow \quad S^* = W_v W_k^{-1}.$$

- For a stationary, single-head DeltaNet model, the fixed optimum suggests weakening context-specific updates.
- For Softmax, the paper approximates the normalized state update with a $1/t$ step at large t.
- These arguments do not establish degradation for a trained multilayer Transformer; a vanishing step alone does not prove vanishing cumulative adaptation.

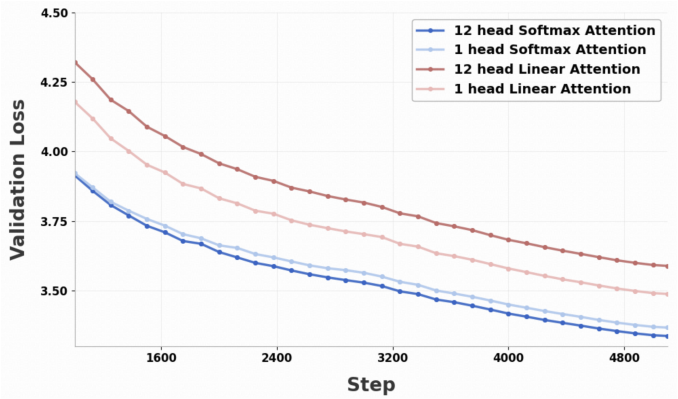
Paper, Sec. 2.2.2, PDF pp. 16–17; Appendices B.2 and F, PDF pp. 30–31, 46.

Evaluation separates retrieval and state tracking

Question	Setup and metric
Head allocation	GPT-2 small: 12 layers, width 768; 12 vs. 1 head; validation loss ↓. Both kernels use per-head RMS normalization.
Swap tracking	Five objects, context length 16; compare 1–8-layer Transformer with DeltaFormer kernels; accuracy ↑.
Scaling/curriculum	Fixed $d = 128$, $n = 128, 256, 512$; direct length 256 versus curriculum 32 to 256; accuracy ↑.
Graph reachability	32-node constructed DAG task; Transformer versus DeltaFormer; accuracy ↑.

Paper, Sec. 3, Figs. 2–7, PDF pp. 17–21.

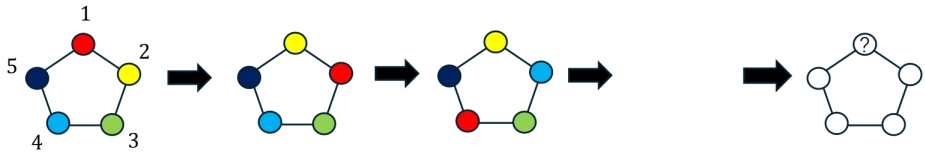
One head helps linear attention but hurts the exp kernel



At fixed width 768, linear attention's one-head curve improves; the exp-kernel curve slightly worsens. Both use per-head RMS normalization.

Paper, Fig. 2, PDF pp. 17–18. Y-axis: validation loss, lower is better.

The swap task asks for the current value at position 1

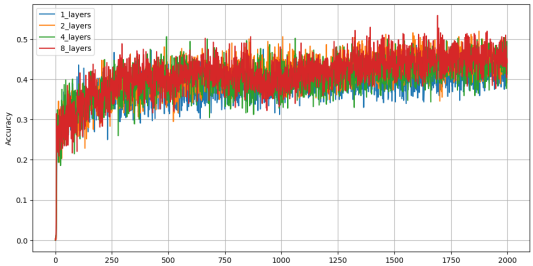


Each input swaps two of five positions; at each step the model predicts the token now at position 1. Default context length is 16.

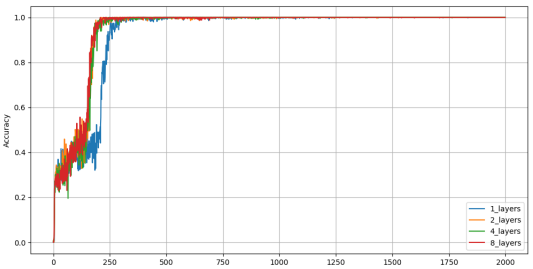
Paper, Sec. 3.2, Fig. 3, PDF p. 18.

Residual writing learns the five-object swap task

Transformer, 1–8 layers



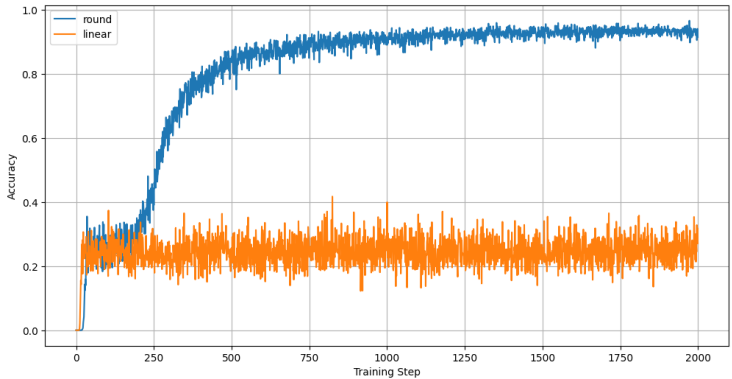
DeltaFormer, rounded κ_1



Transformer plateaus around 0.4–0.45; rounded and linear write kernels approach 1.0 within roughly 300 steps. Read each panel's y-axis separately.

Paper, Fig. 4, PDF pp. 18–19. Training accuracy vs. training step; κ_2 is Softmax.

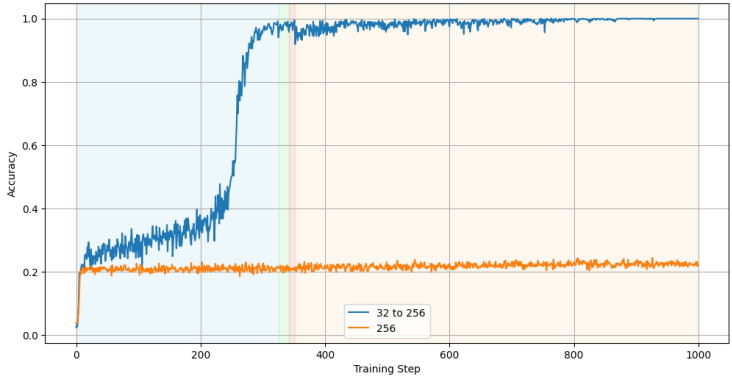
At $d = 128$, a nonlinear write kernel scales better



For $n = 512$, the rounded-kernel curve exceeds 0.9, while the linear curve remains near 0.25. Keys and values were preset near orthogonality; the model learns mainly to read.

Paper, Sec. 3.2, Fig. 5c, PDF p. 19. Accuracy vs. training step; swap length 1024.

Curriculum resolves a hard optimization regime

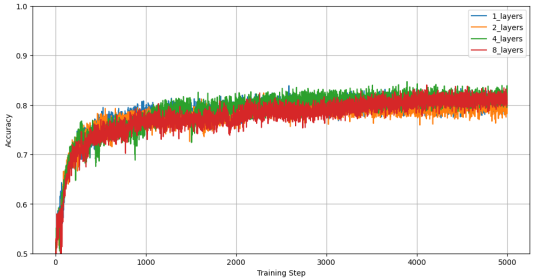


Without position embeddings, direct training at length 256 stays near chance (0.2); increasing length from 32 to 256 approaches full accuracy.

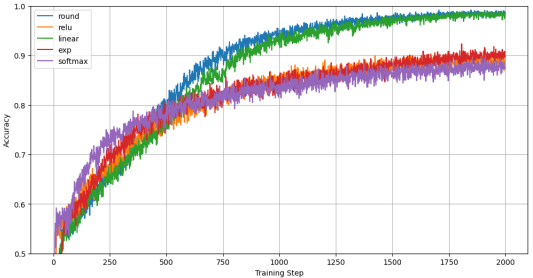
Paper, Sec. 3.2, Fig. 6b, PDF p. 20. Y-axis: accuracy at the current training length; colored bands mark curriculum stages.

DeltaFormer improves the constructed DAG task

Transformer, 1–8 layers



DeltaFormer, 1 layer



On a 32-node reachability task, Transformer curves remain around 0.8; several DeltaFormer write kernels approach 1.0.

Paper, Sec. 3.3, Fig. 7, PDF pp. 20–21. Accuracy vs. training step; κ_2 is Softmax.

What the evidence supports

1. **Mechanism:** kernels control interference; residual writing changes what is stored.
2. **Theory:** an exact-kernel, near-orthogonal construction tracks swaps with logarithmic key dimension.
3. **Experiments:** selected kernels and curricula learn finite swap and reachability tasks more readily than the reported Transformer baselines.

Next question

Does residual writing retain its benefit in end-to-end language modeling under equal compute and realistic context extrapolation?

Paper, Secs. 2–4, PDF pp. 4–22; Appendix E, PDF pp. 35–40.

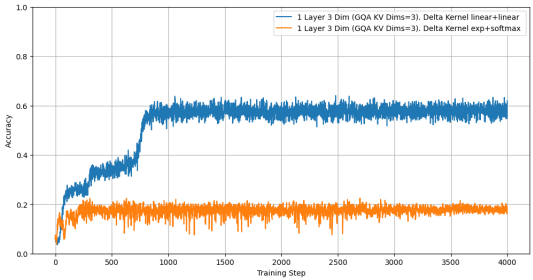
References and links

- Zhong, Xu, Ao, Shi. *Understanding Transformer from the Perspective of Associative Memory*. arXiv:2505.19488v1, 2025.
- Distinct follow-up: Xu et al. *DeltaFormer: Unlock the State Space of Transformer*. NeurIPS 2025 proceedings.
- Flash Linear Attention: third-party DeltaFormer implementation; author toy code is in the original paper's Appendix E.4.

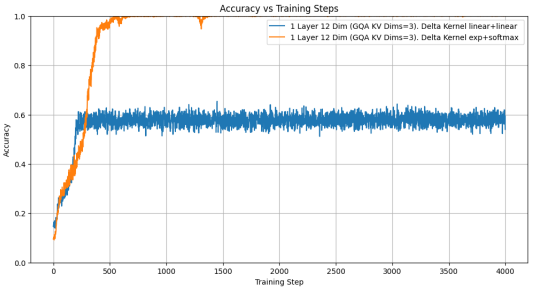
Publication/code links checked 6 Nov 2026. Original arXiv version is the source for every result slide.

Backup: multiple write queries improve a tiny-key setting

One write query



Four write queries



With head dimension 3, the nonlinear configuration rises from about 0.2 to near 1.0 as write-query count increases; the linear configuration remains near 0.6.

Paper, Appendix E.2, Fig. 8, PDF p. 40. Five-object, 16-swap training task.

Backup: the triangular solve exposes the compute trade-off

$$u_t = v_t - \sum_{i<t} A_{t,i} u_i, \qquad U = (I + A)^{-1} V.$$

- A is strictly lower triangular; $(I + A)^{-1}$ exists.
- A chunk can be solved with matrix operations; chunks remain sequential.
- Parallel inversion adds work and does not by itself establish practical speed.

Paper, Appendix D.3, Eqs. (96)–(100), PDF pp. 33–35.