Gated Slot Attention-2

Two-Sided Associative Memory Correction in Linear Attention

Research group reading presentation

arXiv:2610.02816v1; 13 November 2026

GSA2 pairs key-side and value-side correction through slots

One association, two complementary editing roles

$$k_t \xleftarrow{\text{OJA2}} w_t \xrightarrow{\text{Delta-2}} v_t$$

Paper authors: Ruijie Li, Shengnan Ding, Weimin Zhang, Derick Tang, Zhanpeng Zeng, Qinsong Zeng, Ming Chen, Jiaxi Hu and Yuxuan Liang.

- Two recurrent memories share a learned slot code w_t.
- Each stage has independent decay, erase and write controls.

arXiv v1, 2 October 2026. Publication unverified as of 2026-11-13.
PDF header says ICLR 2027; author page lists arXiv; no official record found.
Community code: FLA. Author's complete release unverified.
Paper, Abstract and Sec. 1, PDF pp. 1–2. External status checked 2026-11-13.

GSA factorizes key–value memory through slots

$$S_t^{(1)} \in \mathbb{R}^{M \times D_k}, \qquad S_t^{(2)} \in \mathbb{R}^{D_v \times M}$$

$$\begin{aligned} S_t^{(1)} &= \mathrm{Diag}(\alpha_t) S_{t-1}^{(1)} + (\mathbf{1} - \alpha_t) k_t^\top, \\ S_t^{(2)} &= S_{t-1}^{(2)} \mathrm{Diag}(\alpha_t) + v_t (\mathbf{1} - \alpha_t)^\top, \\ o_t &= S_t^{(2)} \,\mathrm{softmax}(S_t^{(1)} q_t). \end{aligned}$$

- Keys map to M slots; slot activations map to values.
- Original GSA updates both memories by gated accumulation.
- GSA2 equips each side with an explicit associative correction.

Paper, Sec. 2.2, Eqs. (4)–(6), PDF p. 3.

Oja correction reverses the prediction direction of Delta

Delta: key predicts value

$$\mathcal{L}_t^{\Delta}(S) = \frac{1}{2}\|v_t - Sk_t\|_2^2$$

$$S_t = S_{t-1} + \beta_t(v_t - S_{t-1}k_t)k_t^\top$$

Correction acts on the right of S.

Oja: value predicts key

$$\mathcal{L}_t^{K}(S) = \frac{1}{2}\|k_t - S^\top v_t\|_2^2$$

$$S_t = S_{t-1} + \beta_t v_t(k_t - S_{t-1}^\top v_t)^\top$$

Correction acts on the left of S.

Transpose duality

With $T = S^\top$, the Oja update is a Delta update with swapped roles. GSA2 places these directions in two semantically different memories.

Paper, Secs. 2.1 and 3.1, Eqs. (1)–(2), (7)–(8), PDF pp. 2–3. Duality: reading analysis.

OJA2 separates what is erased from what is written

$$\widetilde{S}_{t-1} = \mathrm{Diag}(\alpha_t) S_{t-1}$$

$$\boxed{S_t = [I - v_t (b_t \odot v_t)^\top] \widetilde{S}_{t-1} + v_t (c_t \odot k_t)^\top}$$

- $b_t \in \mathbb{R}^{D_v}$ controls the value-side erase direction.
- $c_t \in \mathbb{R}^{D_k}$ controls key-channel writes.
- Setting $b_t = \beta_t \mathbf{1}$ and $c_t = \beta_t \mathbf{1}$ recovers OJA.

Why decouple?

Removing an old association and admitting new content need not share one scalar update strength.

Paper, Sec. 3.1, Eq. (10), PDF p. 4; Appendix A.2, pp. 14–15.

Stage 1 uses the shared slot code to correct keys

$$S_t^{(1)} \in \mathbb{R}^{M \times D_k}, \qquad w_t \in \mathbb{R}^M$$

$$\begin{aligned} S_t^{(1)} = &[I - w_t(b_t^{(1)} \odot w_t)^\top] \operatorname{Diag}(\alpha_t^{(1)}) S_{t-1}^{(1)} \\ &+ w_t(c_t^{(1)} \odot k_t)^\top. \end{aligned}$$

- The associated key is predicted as $S^{(1)\top} w_t$.
- w_t plays the value-side role of the OJA2 update.
- Erase and decay gates act in slot space; writes select key channels.

The slot code is projected from the current input.

Paper, Sec. 3.2, Eq. (11), PDF p. 4; Sec. 3.4, p. 6.

Stage 2 corrects values; a query then reads both memories

$$S_t^{(2)} = S_{t-1}^{(2)} \operatorname{Diag}(\alpha_t^{(2)})[I - (b_t^{(2)} \odot w_t) w_t^\top] + (c_t^{(2)} \odot v_t) w_t^\top, \qquad S_t^{(2)} \in \mathbb{R}^{D_v \times M}.$$

$$h_t = \operatorname{SiLU}(S_t^{(1)} q_t), \qquad o_t = S_t^{(2)} h_t$$

- The same w_t is now the addressing key of a Delta-2 memory.
- h_t is a read activation; it is different from the write code w_t.
- SiLU replaces GSA's normalized competition between slots.

Paper, Sec. 3.2, Eqs. (12)–(13), PDF pp. 4–5. Compact readout notation from the paper.

A one-hot slot illustrates the two editing roles

Teaching example: two slots, $w_t = (0, 1)^\top$

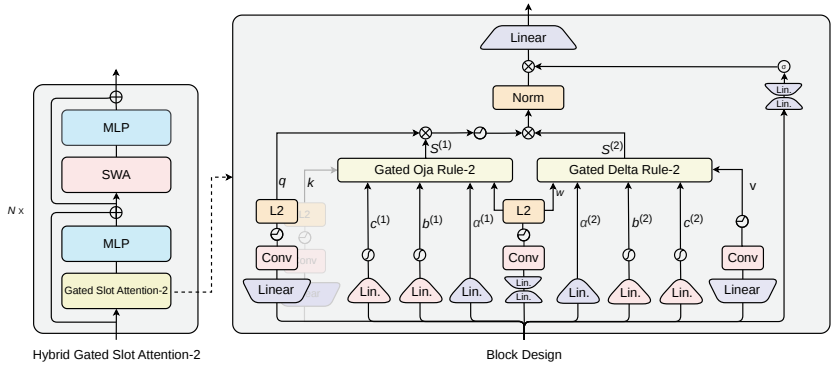
Set erase/write gates to 1 and decay to the identity.

Memory	Active portion	Effect of the write
$S^{(1)}$	Second row	Replace by $k_t^\top$
$S^{(2)}$	Second column	Replace by v_t

- Learned dense slot codes distribute edits across slot directions.
- Total state size is $M(D_k + D_v)$ elements per head.
- Fixed with context length; equal parameter count need not imply equal state capacity.

Teaching example and state count derived from Eqs. (11)–(12), PDF p. 4.

The block supports recurrent and SWA hybrid backbones



q, k, w are L2-normalized; output uses RMSNorm and a sigmoid gate.

Hybrid cell: GSA2–MLP–SWA–MLP, with a fixed 2K SWA window.

Original Fig. 1, PDF p. 5; Sec. 3.4, p. 6. Complete architecture figure crop.

Chunkwise training absorbs edits into a triangular solve

- Accumulated rank-one transitions admit a compact WY form.
- The OJA2 stage derives new gate-aware UT factors.
- The value stage reuses the GDN2 implementation.

OJA2 chunk-boundary state

$$S_{\text{out}} = \text{Diag}(p_C) S_{\text{in}} + (A_{\text{suffix}} \odot V)^{\top} (U - W S_{\text{in}})$$

- U, W: auxiliaries from unit lower-triangular systems.
- W is not the token-level slot code w_t.
- Fixed chunk length gives linear sequence scaling; recurrent state stays bounded.

Paper, Sec. 3.3, Eqs. (14)–(15), PDF pp. 5–6; Appendix A.1–A.2, pp. 13–15.

The study trains 1.3B models on 100B tokens

Data / tokenizer	FineWeb-Edu; TinyLlama v1.1, 32K vocabulary
Training length	4096; hybrid SWA window 2048
Global batch	128 sequences, approximately 0.5M tokens
AdamW	$(0.9, 0.95)$; peak LR 4×10^{-4}; weight decay 0.1
Schedule	Warm up 1B tokens; cosine decay to 10% peak LR
Precision / GPUs	BF16; 32 NVIDIA H20 GPUs across four nodes

Starred recurrent baselines in Table 2 are taken from GDN2.
OJA, OJA2, GSA and GSA2 are evaluated in this work.

Paper, Sec. 4 and Table 2, PDF pp. 6–7; Appendix A.3.1, p. 15.

Recurrent GSA2 improves both PPL and the accuracy average

Recurrent model	Wiki. PPL ↓	LMB. PPL ↓	Accuracy avg. ↑
GDN2	15.90	11.41	53.11
Mamba-3 MIMO	16.17	11.10	53.31
OJA2	18.28	14.84	50.15
GSA	16.52	11.22	53.28
GSA2	15.69	10.68	54.08

- Versus GDN2: -0.21 / -0.73 PPL and $+0.97$ accuracy points.
- OJA2 alone is weaker; its value is in the complementary pairing.

Paper, Table 2, PDF p. 6. Accuracy average: LAMBADA plus eight commonsense tasks (%).

Hybrid GSA2 gains accuracy while WikiText PPL worsens

Hybrid model	Wiki. PPL ↓	LMB. PPL ↓	Accuracy avg. ↑
GDN2 + SWA	16.26	11.93	52.65
Mamba-3 MIMO + SWA	15.94	11.75	53.63
GSA + SWA	15.98	11.36	52.67
GSA2 + SWA	17.23	10.80	53.96

A concrete trade-off

WikiText PPL is 0.97 higher than GDN2, about 5.97% worse. The paper does not explain this degradation.

The average gain does not mean every task improves.

Paper, Table 2, PDF p. 6. Accuracy average uses nine reported accuracy columns.

At 2K inputs, local attention changes the retrieval ranking

Model	Recurrent avg. ↑	Hybrid avg. ↑
GDN2	29.88	35.41
OJA	25.95	**37.49**
OJA2	26.24	37.19
GSA	26.71	32.89
GSA2	31.26	36.70

- Six real-world retrieval tasks; all inputs truncated to 2K.
- Recurrent GSA2 beats GDN2 by 1.38 percentage points.
- With a 2K SWA window, OJA/OJA2 hybrids become strongest on average.

Paper, Table 3 and discussion, PDF p. 7. Six-task accuracy average (%).

RULER averages favor GSA2 in both model families

Family	Model	NIAH avg. ↑	Other tasks avg. ↑
Recurrent	GSA	25.2	17.9
	GDN2	39.7	26.0
	GSA2	42.1	31.8
Hybrid	GSA	44.7	33.0
	GDN2	47.7	28.5
	GSA2	50.6	36.2

NIAH: 6 task variants $\times$ 4 lengths $=$ 24 settings.
Other tasks: 7 tasks $\times$ 4 lengths $=$ 28 settings, from 1K to 8K.

Paper, Fig. 2, PDF p. 8; full Tables 6–7, pp. 16–17. Accuracy (%).

Different RULER groups account for the average gains

Family	Model	S-NIAH ↑	MK-NIAH ↑
Recurrent	GDN2	68.6	10.8
	GSA2	72.8	11.4
Hybrid	GDN2	**63.2**	32.2
	GSA2	62.1	39.1

- Each subgroup averages 3 variants $\times$ 4 lengths.
- Recurrent GSA2 still scores zero on all MK-NIAH-3 settings.
- At 8K S-NIAH-1: recurrent GSA2 86.0%, GDN2 92.8%.

Computed from paper Table 6, PDF p. 16; group means rounded to one decimal.

LongBench gains vary across tasks

Model	Recurrent avg. ↑	Hybrid avg. ↑
GDN2	16.8	16.4
OJA2	13.6	15.9
GSA	13.6	10.4
GSA2	17.3	17.3

- Average of 11 task scores across QA, summaries, few-shot and code.
- Recurrent TREC: 46.5 versus GDN2's 37.0; a large contributor.
- Recurrent GSA2 trails GDN2 on several QA and code tasks.
- Hybrid SAMSum improves; MultiNews and RepoBench-P decline.

Paper, Table 4, PDF p. 8. Benchmark task scores, not a uniform accuracy metric.

Complementary rules outperform repeating one rule overall

Pairing	Wiki. ↓	LMB. ↓	Acc. ↑	MK ↑	LongB. ↑
OJA2 + Delta-2	**15.69**	10.68	**54.08**	11.4	**17.3**
Both Delta-2	15.92	**10.27**	53.03	**13.1**	16.8
Both OJA2	16.97	12.46	52.35	8.4	13.8

- The first and second memories benefit from different correction roles.
- Both-Delta wins LAMBADA PPL and multi-key NIAH.
- The evidence supports an overall pairing advantage with task trade-offs.

Paper, Table 5, PDF p. 9. All variants: 1.3B parameters, 100B tokens; Acc. and MK in %.

Scaling, independent activation and 128 slots work best here

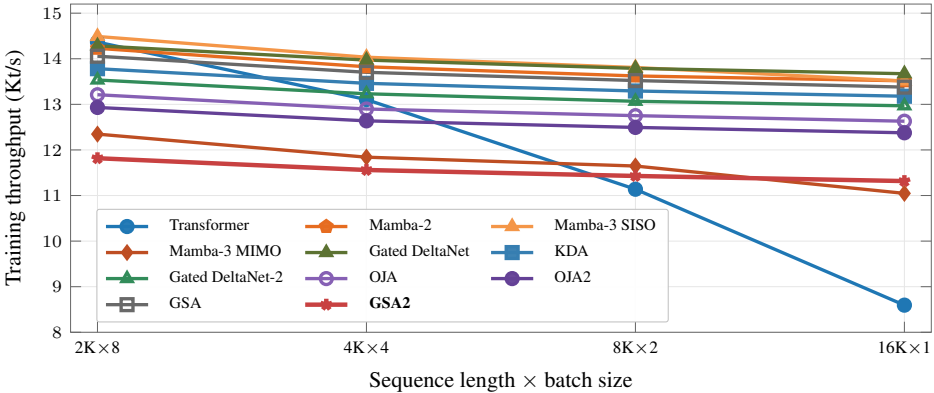
Variant	Wiki. ↓	Acc. ↑	S-NIAH ↑	LongB. ↑
Default, 128 slots	**15.69**	**54.08**	**72.8**	**17.3**
No $1/\sqrt{d}$ scaling	19.76	48.50	61.8	13.0
SiLU $\rightarrow$ Softmax	16.92	50.66	55.6	14.0
64 slots	16.55	53.48	62.6	16.9
256 slots	16.37	52.87	61.7	15.3

More slots do not automatically improve the measured balance.

The compact readout equation does not spell out the scaling placement.

Paper, Table 5 and ablation discussion, PDF pp. 8–9. Acc. and S-NIAH in %.

Training throughput stays flat with a two-stage constant cost



GSA2: 11.82 → 11.32 Kt/s (−4.2%).
Transformer: 14.37 → 8.60 Kt/s (−40.2%).

Original Fig. 3, PDF p. 9. Hybrid 1.3B models, one H20, 16,384 tokens per training step.

The practical gain is better memory editing at an added cost

1. **Mechanism:** OJA2 edits key–slot associations; Delta-2 edits slot–value associations.
2. **Evidence:** recurrent PPL and cross-task averages improve; individual tasks retain trade-offs.
3. **Cost:** at 16K, GSA2 training throughput is about 8.56% below OJA2 and 12.73% below GDN2.

Follow-up questions tied to the evidence

Separate state capacity from correction direction, explain hybrid WikiText PPL, and evaluate seed variation and inference-time costs.

Paper, Secs. 3–6, PDF pp. 3–10. Throughput ratios calculated from Fig. 3 coordinates.

References and implementation status

- Li et al. *Gated Slot Attention-2: Two-Sided Associative Memory Correction in Linear Attention*.
  arXiv:2610.02816v1, 2026.
- First author's homepage: lists arXiv 2026.
  ICLR 2027 publication is not independently verified as of 2026-11-13.
- FLA README lists GSA2; populated layer and model implementations are public.
- Community implementation; author-official status and complete training release unverified.

Paper v1 promises full source upon acceptance (reproducibility statement, PDF p. 10).

Backup: decoupling both stages gives the best overall balance

Stage 1	Stage 2	Wiki. ↓	LMB. ↓	Acc. ↑	LongB. ↑
Coupled	Coupled	16.22	**10.29**	53.25	17.1
Coupled	Decoupled	16.01	10.65	52.46	14.8
Decoupled	Coupled	16.28	11.25	53.37	16.2
Decoupled	Decoupled	**15.69**	10.68	**54.08**	**17.3**

- Independent erase/write control benefits both memory roles overall.
- Decoupling only one side gives inconsistent gains.
- Coupling both retains the best LAMBADA PPL of these four variants.

Paper, Table 5, PDF p. 9. All variants: 1.3B parameters, 100B tokens. Acc. in %.

Backup: OJA2 erase gates enter the triangular dependency

For a generic OJA2 state $S \in \mathbb{R}^{D_v \times D_k}$:

$$P_{r,:} = p_r^\top, \qquad p_r = \alpha_1 \odot \cdots \odot \alpha_r$$

$$L = \text{lower}\left([(B \odot V) \odot P](V \oslash P)^\top \right), \quad T = (I + L)^{-1}$$

$$W = T[P \odot (B \odot V)], \qquad U = T(C_{\text{w}} \odot K)$$

- B: erase-gate rows; C_{w}: write-gate rows.
- Erase affects L, T, W; write gates act directly on U.
- Strictly positive decay supports the divisions; triangular solve uses FP32.

Paper, Appendix A.2, Eq. (26), PDF p. 15. T renames the paper's auxiliary M matrix.

Backup: chunk outputs use effective key-side residuals

$$R = U - WS_{\text{in}}, \qquad R \in \mathbb{R}^{C \times D_k}$$

$$S_{\text{out}} = \text{Diag}(p_C) S_{\text{in}} + (A_{\text{suffix}} \odot V)^\top R$$

$$O = P \odot \left[Q S_{\text{in}}^\top + \text{Tril}(QR^\top)(V \oslash P) \right]$$

- Suffix decay applies after each write; its last row is all ones.
- Tril includes the diagonal, so queries see the current write.
- Within-chunk computation is dominated by matrix multiplications.

Paper, Appendix A.1–A.2, Eqs. (22), (26), PDF pp. 14–15. R introduced for readability.

Backup: preserve the metric and table identities

- Table 2 accuracy average contains nine columns: LAMBADA plus eight commonsense benchmarks.
- Table 5's caption says LAMBADA plus nine tasks; this does not match the displayed columns.
- Default real-world retrieval is 31.26 in Table 3 and 31.51 in Table 5; the paper does not reconcile the difference.
- LongBench averages task-specific scores; it is not one uniform accuracy measure.

Compare rows within their stated protocols and keep these values separate.

Paper, Tables 2–5, PDF pp. 6–9; benchmark descriptions, Appendix A.3.2, pp. 15–16.

Backup: throughput measures complete optimization steps

- One NVIDIA H20, BF16, approximately 1.30B parameters.
- Fixed 16,384 tokens: 2K×8, 4K×4, 8K×2, 16K×1.
- Forward, fused cross-entropy, backward, gradient clipping and fused AdamW; no gradient accumulation.
- Random token IDs generated on GPU; excludes input pipeline costs.
- 20 warm-up and 50 measured iterations per configuration.
- Three physical GPUs; report the median of three run-level means.

Bounded-state decoding is an architectural property; this experiment measures training.

Paper, Appendix A.3.3, PDF p. 16; Fig. 3, p. 9. Units: thousand tokens per second.